

**SVEUČILIŠTE U ZAGREBU
GEODETSKI FAKULTET**

Nenad Klobučar

**FuelGIS – aplikacija za pregled benzinskih postaja na području
Grada Zagreba, razvijena na QGIS platformi**

DIPLOMSKI RAD

Zagreb, rujan 2012.

I. Autor

Ime i prezime: Nenad Klobučar

Datum i mjesto rođenja: 26. 03. 1986., Zagreb, Hrvatska

II. Diplomski rad

Predmet: Programiranje u geoinformacijskim sustavima

Naslov: FuelGIS – aplikacija za pregled benzinskih postaja na području Grada Zagreba, razvijena na QGIS platformi

Mentor: prof. dr. sc. Damir Medak

Voditelj: dipl. ing. Dražen Odobašić

III. Ocjena i obrana

Datum zadavanja zadatka: 25. 01. 2012.

Datum obrane: 14. 09. 2012.

Sastav povjerenstva pred kojim je branjen rad:

prof. dr. sc. Damir Medak

prof. dr. sc. Drago Špoljarić

dr. sc. Ivan Medved

Zahvala:

Zahvaljujem se svome mentoru prof. dr. sc. Damiru Medaku i voditelju mog diplomskog rada znanstvenom novaku Draženu Odobašiću dipl. ing. koji su svojim znanstvenim i stručnim savjetima pomogli u izradi ovog diplomskog rada.

Posebno se želim zahvaliti svojim roditeljima i rodbini koji su me tijekom čitavog školovanja podupirali te omogućili ostvarenje sve viših i viših ciljeva.

Na kraju se želim zahvaliti svim kolegama koji su mi vrijeme provedeno na fakultetu uljepšali svojom prisutnošću.

FuelGIS – aplikacija za pregled benzinskih postaja na području Grada Zagreba, razvijena na QGIS platformi

Sažetak: Ovaj diplomska rad opisuje razvoj GIS aplikacije za pregled benzinskih postaja na području Grada Zagreba, razvijene na QGIS platformi – FuelGIS. Prva poglavlja opisuju prikupljanje i obradu rasterskih te vektorskih prostornih i atributnih podataka koji su bili potrebni za rad aplikacije. Daljnja poglavlja objašnjavaju izradu same aplikacije pomoću programskog jezika Python te njegovih modula PyQt4-a i PyQGIS-a. Rezultat svega toga je samostalna aplikacija koja ovisno o korisničkim upitima postavljenim putem grafičkog sučelja prikazuje na rasterkoj podlozi odgovarajuće prostorne i atributne informacije o benzinskim postajama na području grada Zagreba.

Ključne riječi: GIS aplikacija, benzinske postaje, QGIS, prostorni i atributni podaci, Python, PyQt4, PyQGIS, rasterska podloga, grafičko sučelje

FuelGIS – application for browsing petrol stations in city of Zagreb, developed on QGIS platform

Abstract: This master thesis describes the development of a GIS application for browsing petrol stations in city of Zagreb, developed on QGIS platform – FuelGIS. First chapters describe collection and processing of raster and vector spatial and attribute data which is required for the application. Next chapters explain how to create the application using the Python programming language and its modules PyQt4 and a PyQGIS. The result is a standalone application which displays on the raster background appropriate spatial and attribute information about petrol stations in city of Zagreb, depending on the user created queries through the graphical interface.

Keywords: GIS application, petrol stations, QGIS, spatial and attribute data, Python, PyQt4, PyQGIS, raster background, graphical interface

FuelGIS – aplikacija za pregled benzinskih postaja na području Grada Zagreba, razvijena na QGIS platformi

S A D R Ž A J

1. UVOD.....	8
2. KORIŠTENE TEHNOLOGIJE, APLIKACIJE I ALATI	10
2.1. GOOGLE EARTH	10
2.2. MICROSOFT EXCEL I CSV	10
2.3. QUANTUM GIS (QGIS)	11
2.4. ESRI SHAPE	12
2.5. RASTERSKI FORMATI JPG I TIFF	14
2.6. GIMP	15
2.7. PYTHON	15
2.8. PYQT4	17
2.8.1. <i>Općenito o PyQt4</i>	17
2.8.2. <i>Mehanizam signala i slotova</i>	18
2.9. PYQGIS	20
2.10. SUBLIME TEXT 2.....	21
3. PRIKUPLJANJE PODATAKA	23
3.1. RASTERSKA PODLOGA	23
3.2. POLOŽAJNI I ATRIBUTNI PODACI	23
4. OBRADA PODATAKA	28
4.1. IZRADA KOORDINATNOG SUSTAVA.....	28
4.2. STVARANJE SLOJA IZ CSV DATOTEKE	29
4.3. IZRADA ESRI SHAPE DATOTEKE	32
4.4. OBRADA RASTERSKE PODLOGE	34
4.5. IZRADA PIRAMIDE	36

5. IZRADA FUELGIS APLIKACIJE	39
5.1. INSTALACIJA OSGEO4W PROGRAMSKOG PAKETA.....	39
5.2. PODEŠAVANJE OSGEO4W PROGRAMSKOG PAKETA I OSTALE PRIPREMNE RADNJE .	47
5.2.1. <i>Podešavanje OsGeo4W programskog paketa.....</i>	48
5.2.2. <i>Provjera podešavanja OsGeo4W programskog paketa.....</i>	49
5.2.3. <i>Izrada bat datoteke</i>	51
5.3. IZRADA SUČELJA FUELGIS APLIKACIJE	52
5.3.1. <i>Početni parametri PyQt4 aplikacije</i>	53
5.3.2. <i>Osnovne funkcionalnosti PyQt4 aplikacije.....</i>	54
5.3.3. <i>Glavni okvir aplikacije.....</i>	55
5.3.4. <i>Podloga za iscrtavanje karte</i>	55
5.3.5. <i>Statusna traka</i>	56
5.3.6. <i>Prikaz koordinata u statusnoj traci</i>	56
5.3.7. <i>Gumbi za približavanje, udaljavanje i pomicanje</i>	58
5.3.8. <i>Gumb za odabir</i>	59
5.3.9. <i>Aktivacijski okviri</i>	60
5.3.10. <i>Glavna alatna traka</i>	62
5.3.11. <i>Info traka.....</i>	63
5.3.12. <i>Zatvaranje FuelGIS aplikacije</i>	65
5.4. IMPLEMENTACIJA QGIS FUNKCIONALNOSTI U FUELGIS APLIKACIJU	66
5.4.1. <i>Početni parametri PyQGIS aplikacije</i>	67
5.4.2. <i>Pozivanje QGIS okoline.....</i>	67
5.4.3. <i>Koordinatni sustav.....</i>	68
5.4.4. <i>Rad s rasterskim i vektorskim slojem.....</i>	68
5.4.5. <i>Odabir želenog vektorskog elementa i prikaz njegovih atributa</i>	69
5.4.6. <i>Prikaz želenog sadržaja</i>	72
5.4.7. <i>Simbologija – izgled vektorskog sloja</i>	74
6. UPUTE ZA RAD S FUELGIS APLIKACIJOM	77
7. ZAKLJUČAK	81
8. PRILOZI.....	82

8.1.	PRILOG 1 – IZVORNI KOD APLIKACIJE	82
8.2.	SADRŽAJ PRILOŽENOG OPTIČKOG MEDIJA (DVD-A).....	93

Literatura

Popis slika

Popis tablica

Životopis

1. Uvod

Geografski informacijski sustav (*GIS*) je sustav za upravljanje prostornim podacima i osobinama pridruženih njima. U najstrožem smislu to je računalni sustav sposoban za integriranje, spremanje, uređivanje, analiziranje i prikazivanje geografskih informacija. U općenitijem smislu *GIS* je oruđe "pametne karte" koje dopušta korisnicima stvaranje interaktivnih upitnika (istraživanja koja stvara korisnik), analiziranje prostornih informacija i uređivanje podataka (URL 1).

U današnje vrijeme *GIS-ovi* se koriste u razne svrhe poput upravljanja prirodnim resursima, prometu i logistici, nacionalnoj obrani, kartiranju kriminala itd. Pritom se najčešće koriste razne komercijalne *GIS* aplikacije od kojih su najpoznatije *ArcGIS* i *GeoMedia*. Licence za korištenje takvih aplikacija su skupe te ih korisnici kupuju u slučaju kada pomoću njih žele ostvariti financijsku korist. Kako bi *GIS* aplikacije postale dostupne što većem krugu korisnika potrebno je sniziti njihovu cijenu. Na cijenu *GIS* aplikacije utječu financijska sredstva, vrijeme te ostali resursi korišteni u njenom razvoju. Stoga je potrebno smanjiti utjecaj navedenih elemenata na konačnu cijenu aplikacije. To je moguće ostvariti korištenjem tehnologija otvorenog koda (eng. *Open source technologies*) koje omogućuju brži i jeftiniji razvoj *GIS* aplikacija.

Ovaj rad prati razvoj jednostavne *GIS* aplikacije pomoću tehnologija otvorenog koda. Cilj je pokazati kako vlastite aplikacije temeljene na otvorenoj platformi mogu biti kvalitetna alternativa skupim, komercijalnim aplikacijama. Zbog svakodnevne potrebe modernog društva za naftnim derivatima odabrana je izrada *GIS* aplikacije za pregled benzinskih postaja na području Grada Zagreba s nazivom *FuelGIS*. Ideja ovog diplomskega rada je izrada samostalne aplikacije namijenjene izvršavanju na stolnim i prijenosnim računalima. *FuelGIS* se temelji na *Quantum GIS (QGIS)* platformi te je pisan u programskom jeziku *Python*. Budući da *QGIS* i *Python* oboje pripadaju tehnologijama otvorenog koda, njihovo korištenje ne zahtijeva nikakva financijska ulaganja za kupovinu licenci. Takoder, oni raspolažu s gotovo beskonačnim rasponom mogućnosti i funkcionalnosti. Zbog tih razloga su idealni za izradu *FuelGIS* aplikacije. Aplikacija se sastoji od sučelja (napravljenog korištenjem *PyQt4-a*) s potrebnim alatima te rasterskih i vektorskih podataka. Zamisao je

da se pomoću sučelja na rasterskoj podlozi omogući pregled položajnih i atributnih podataka benzinskih postaja na području Grada Zagreba. U tu svrhu iskorištene su mogućnosti *QGIS* platforme poput prikazivanja, rukovanja i analize rasterskih i vektorskih prostornih te atributnih podataka. Prije pisanja koda u *Python* programskom jeziku bilo je potrebno prikupiti te obraditi položajne i atributne podatke o benzinskim postajama na području Grada Zagreba. Pritom je posebna pažnja posvećena provjeri kvalitete i točnosti prikupljenih podataka jer su oni najvažniji dio svakog *GIS-a*. Izrada praktičnog dijela diplomskog rada sastojala se od tri koraka: prikupljanja podataka, obrade podataka i izrade *FuelGIS* aplikacije. U nastavku rada detaljno su opisani svi navedeni koraci.

2. Korištene tehnologije, aplikacije i alati

Prilikom prikupljanja i obrade podataka te izrade *FuelGIS* aplikacije korištene su različite tehnologije, aplikacije i alati. Većina tehnologija su otvorenog koda, poput programskog jezika *Python* koji je korišten za izradu aplikacije. Sve tehnologije, aplikacije i alati korišteni u ovom radu opisani su u sljedećim potpoglavljkima.

2.1. Google Earth

Google Earth omogućuje virtualni 3D prikaz Zemljine površine. Taj prikaz je stvoren od velikog broja različitih satelitskih snimaka koji su snimljeni u određenom vremenskom periodu te naknadno spojeni. *Google Earth* je slobodno dostupan putem Interneta svim korisnicima (URL 2). Unosom geografskih koordinata željenog mesta virtualna Zemlja se počinje okretati i namještati za prikaz tražene lokacije. Na taj način moguće je dobiti detaljne informacije, npr. postoji li na unesenim koordinatama benzinska postaja.

2.2. Microsoft Excel i CSV

Microsoft Excel je program za tablično računanje, proizvod kompanije *Microsoft*, te je sastavni dio programskog paketa *Microsoft Office*. *Microsoft Excel* uglavnom služi za rješavanje problema matematičkog tipa pomoći tablica i polja koje je moguće povezivati različitim formulama, no može poslužiti i za izradu jednostavnijih baza podataka (URL 3).

CSV (eng. *comma-separated values*) datoteka pohranjuje tablične podatke (brojeve i tekst) kao običan tekst. Običan tekst znači da je datoteku moguće pročitati s drugim programima jer umjesto binarnog zapisa podataka, koji je čitljiv jedino *Microsoft Excelu*, sadrži niz znakova u tekstualnom obliku. *CSV* datoteka može pohraniti različit broj zapisa međusobno odvojenih nekom vrstom graničnika. Svaki zapis se sastoji od polja odvojenih pomoći nekog drugog znaka ili niza, najčešće običnog zareza ili točke zareza. *CSV* je zajednički, relativno jednostavan format datoteke koji je široko podržan od strane korisničkih, poslovnih i znanstvenih aplikacija. Najčešće se koristi za razmjenu tabličnih podataka između programa koji koriste međusobno nekompatibilne (često zaštićene i/ili nedokumentirane) formate. To je moguće jer veliki broj programa podržava neku od

varijacija *CSV* formata barem kao alternativu za uvoz/izvoz podataka (URL 4). Zbog svih navedenih svojstava *CSV* je idealan kao format za pohranu i razmjenu podataka između raznih programa te se u tu svrhu koristi u ovom radu.

2.3. *Quantum GIS (QGIS)*

Gary Sherman započinje razvoj *Quantum GIS* aplikacije početkom 2002. godine, a verzija 1.0 je izdana u siječnju 2009. godine. *Quantum GIS* je razvijen u C++ programskom jeziku te opsežno koristi *Qt* biblioteke (eng. *library*). Danas *QGIS* održavaju aktivne grupe volontera koje redovito izdaju novije verzije softvera (eng. *updates*) i zatrpe (eng. *bug fixes*). Trenutačno je *Quantum GIS* preveden na 31 jezik uključujući i hrvatski, te se aplikacija koristi na međunarodnoj razini u akademskim i profesionalnim krugovima (URL 5). *Quantum GIS (QGIS)* je korisnički orijentiran geografski informacijski sustav otvorenog koda koji radi na *Linuxu*, *Unixu*, *Mac OS X-u*, *Microsoft Windowsima* i *Android OS-u*. *QGIS* podržava vektorske i rasterske formate prostornih podataka kao i baze podataka te je licenciran pod *GNU-ovom* Općom javnom licencom (URL 6). *GNU* Opća javna licenca jamči slobodu dijeljenja i mijenjanja slobodnih programa, na taj način je program slobodan za sve svoje korisnike. *GNU* Opća javna licenca je jedna od najpopularnijih i najpoznatijih licenca koje se koriste za licenciranje slobodnog softvera (URL 7).

QGIS omogućuje pregledavanje, uređivanje i stvaranje raznih vektorskih i rasterskih formata, uključujući *ESRI Shape* datoteke, prostorne podatke u *PostgreSQL/PostGIS* bazama podataka, *GRASS-ove* vektorske i rasterske podatke ili *GeoTIFF*. Koristeći *Python* ili C++ moguće je stvoriti prilagođene dodatke (eng. *customised plugins*) i samostalne *GIS* aplikacije. *QGIS* dodaci proširuju postojeću funkcionalnost, npr. uvoz podataka u obliku razgraničenog teksta, preuzimanje staza, putova i točaka iz *GPS* uređaja ili prikazivanje slojeva korištenjem *OGC WMS* i *WFS* standarda (URL 6). *QGIS* je moguće preuzeti s matične web stranice (URL 8) na dva načina, kao samostalnu aplikaciju ili kao dio *OSGeo4W* programskog paketa. *OSGeo4W* je programski paket namijenjen *Windows* operativnim sustavima koji održava i razvija *OSGeo* (eng. *Open Source Geospatial Foundation*). *OSGeo* je neprofitna organizacija čiji je cilj podržavanje i promicanje zajedničkog razvoja tehnologija otvorenog koda koje se bave prostornim podacima (URL

9). Cilj *OSGeo4W* projekta je stvoriti programski paket koji će biti jednostavan za instalaciju i korištenje u *Windows* okruženju. Njegov razvoj započet je 2008 godine i još uvijek traje. Važnost tog razvoja je velika jer se pomoću *OSGeo4W* projekta nastoji isporučiti tehnologije otvorenog koda, koje se bave prostornim podacima, većini korisnika diljem svijeta. *OSGeo4W* uz *QGIS*, i sve potrebno za njegovo korištenje, sadrži *Python*, *GRASS*, *GDAL* itd. *Quantum GIS* je preuzet kao samostalna aplikacija te je napravljena njegova instalacija na računalo. Instalacija aplikacije je krajnje jednostavna i poznata svakom prosječnom korisniku računala. U nekoliko koraka potrebitno je pratiti upute te odabrati željenu lokaciju i direktorij na računalu gdje će se aplikacija nalaziti. Prilikom instalacije moguće je odabir i dodatnog sadržaja u obliku raznih primjera, skupova podataka. Navedeni služe za upoznavanje s mogućnostima *Quantum GIS-a*.

2.4. ESRI Shape

ESRI Shape je popularni vektorski zapis podataka koji služi za spremanje prostornih podataka. Razvijen je od strane tvrtke *ESRI* kao (većinom) otvoreni podatkovni oblik, kako bi omogućio međusobnu komunikaciju između *ESRI* aplikacija te ostalog *GIS* softvera. Taj podatkovni oblik je uveden početkom 1990-tih godina prilikom izdavanja *ESRI* aplikacije *ArcView 2* (URL 10). Danas ga je moguće pregledavati i uređivati pomoću raznih komercijalnih i slobodnih programa, poput *Quantum GIS-a*. *ESRI Shape* omogućava opisivanje geometrije nekog objekta pomoću točaka, linija i poligona. Uz geometriju svaki objekt može imati i atributne podatke koji ga opisuju. Stoga *ESRI Shape* uz geometrijske sadržava i pripadajuće atributne podatke o nekom objektu. *ESRI Shape* se zapravo sastoji od skupa datoteka s različitim ekstenzijama. Od toga su tri datoteke obavezne za funkcioniranje tog podatkovnog oblika, dok su ostale opcionalne te pomažu u korištenju pohranjenih podataka. Tablica 1. sadrži popis i opis obaveznih datoteka *ESRI Shape* formata zapisa. Zapisi tih datoteka su međusobno ovisni i to prema položaju zapisa. Dakle, prvi zapis u *shx* ili *dbf* datoteci odnosi se na prvi zapis u *shp* datoteci.

Naziv ekstenzije	Opis ekstenzije
shp	Shape format, služi za spremanje geometrije objekta
shx	Shape indeks format, indeks koji označava geometriju i omogućava brzo pretraživanje u oba smjera, sprijeda ili straga
dbf	Atributni format, atributi poredani u stupce za svaki objekt

Tablica 1. Popis obaveznih ESRI Shape datoteka

Postoje i opcionale datoteke ESRI Shape formata zapisa, one su navedene u tablici 2.

Naziv ekstenzije	Opis ekstenzije
prj	Projekcijski format, koordinatni sustav i projekcijski parametri
sbn i sbx	Prostorni indeks objekata
fbn i fbx	Prostorni indeks objekata za shp datoteke koje se mogu samo pregledavati
ain i aih	Atributni indeks aktivnih polja u tablici ili tematskoj atributnoj tablici
ixs	Indeks geokodiranja za shp datoteku koji se može pregledavati i uređivati
mxs	Indeks geokodiranja za shp datoteku koji se može pregledavati i uređivati u ODB formatu
atx	Atributni indeks za dbf datoteku
shp.xml	Metapodaci u XML formatu
cpg	Definira kodnu stranicu za znakove koji se koriste u dbf datoteci

Tablica 2. Popis opcionih ESRI Shape datoteka

Bitno je napomenuti kako *ESRI Shape* ne pohranjuje topološke odnose geometrijskih objekata, nego se topološki odnosi između objekata dobivaju pomoću topoloških analiza, a njihovi rezultati se zapisuju u nove datoteke. Također, *shp* datoteka koristi jedino polilinije tj. ne podržava krivulje te se zato na krupnjem mjerilu može uočiti izlomljenošć prikaza određenih obilježja. Problem možemo izbjegći povećanjem broja točaka u poliliniji, ali se tada znatno povećava i veličina datoteke. *Shp* datoteke mogu imati maksimalnu veličinu od 2 GB, što znači da se u pojedinačnu datoteku može pohraniti najviše 70 milijuna točaka. Također, u jednu *shp* datoteku moguće je pohraniti samo jednu vrstu geometrijskog objekta, odnosno moguće je pohraniti ili točke ili linije ili poligone. Postoje i druga ograničenja, *dbf* datoteka koja služi za pohranu atributnih podataka ne omogućava spremanje praznih polja. Umjesto toga prazno polje označava s vrijednošću nula što može štetno utjecati na razne statističke analize. Uz to *dbf* datoteka može sadržavati do maksimalno 255 polja, a maksimalna veličina naziva polja je 10 znakova itd (URL 10). Unatoč svim navedenim ograničenjima *ESRI Shape* je široko rasprostranjen podatkovni oblik.

2.5. *Rasterski formati jpg i tiff*

Jpg je uobičajena metoda sažimanja digitalnih fotografija uz određeni gubitak kvalitete. Stupanj kompresije se može podešiti čime se dobiva optimum između veličine datoteke i kvalitete slike. Algoritam kompresije najbolje rezultate postiže na fotografijama i slikama realnih scena s nježnim varijacijama tona i boje. Također, *jpg* je jako popularan i na Internetu gdje je jako važna veličina datoteka koje se koriste za prikaz slika. S druge strane, *jpg* nije pogodan za prikaz linijskih crteža te raznih tekstualnih i grafičkih oblika. U tim slučajevima oštri kontrasti između susjednih piksela mogu uzrokovati zamjetne nepravilnosti. Takve slike je preporučljivo spremati u formatima koji ne uzrokuju gubitak kvalitete, kao što su *tiff*, *gif*, *png* itd (URL 11). Također, *jpg* ne bi trebao biti korišten u situacijama gdje je potrebno točno reproduciranje podataka (npr. u raznim znanstvenim i medicinskim aplikacijama). Osim toga *jpg* nije prikladan za datoteke koje prolaze više izmjena jer prilikom svake dekompresije i ponovne kompresije dolazi do gubitka kvalitete slike. To osobito dolazi do izražaja u situacijama kada se slika pomiče ili izrezuje. Prilikom navigacije računalo mora napraviti dekompresiju cijele *jpg* datoteke što je jako zahtjevna operacija te je za nju potrebno značajno vrijeme (nekoliko sekundi).

Tiff je format za pohranu slika popularan među grafičarima, izdavačkoj industriji te amaterskim i profesionalnim fotografima. *Tiff* je fleksibilan, prilagodljiv format za rukovanje slikama i podacima unutar jedne datoteke. To uključuje i oznake zaglavlja (veličina, raspored podataka o slici, korištena kompresija slike itd) koje određuju geometriju slike. Sposobnost pohrane slikovnih podataka bez gubitka kvalitete čini *tiff* prikladnim formatom za pohranu rasterskih podataka. Za razliku od standardnih *jpg* datoteka, *tiff* datoteke koje ne koriste kompresiju mogu se uređivati i ponovno spremati bez gubitaka u kvaliteti slike (URL 12). Kad ne koriste nikakvu kompresiju *tiff* datoteke su u usporedbi s *jpg* datotekama višestruko veće tj. zauzimaju znatno više prostora na računalu. Prema potrebi *tiff* datoteke mogu koristiti kompresiju s gubitkom kvalitete kako bi se uštedio prostor na računalu.

2.6. *GIMP*

GIMP (eng. *GNU Image Manipulation Program*) je najpopularnija slobodna aplikacija otvorenog koda za obradu i stvaranje digitalnih fotografija. Njegov razvoj započeli su 1995. godine *Spencer Kimball* i *Peter Mattis* u obliku studentskog projekta. Od tada je *GIMP* prošao veliki put, danas ga razvija i održava organizirana grupa volontera te je dostupan u verzijama za operativne sustave *Linux*, *Mac OS X* i *Microsoft Windows*. *GIMP* sadrži alate koji se mogu koristiti za retuširanje i uređivanje slike, slobodno crtanje, mijenjanje veličine, rezanje, foto montažu (kombiniranje više slika), konverziju između različitih slikovnih formata te ostale specijalizirane zadaće (URL 13). *GIMP* je moguće preuzeti s Interneta (URL 14) u obliku instalacijske datoteke. Instalacija aplikacije je krajnje jednostavna i poznata svakom prosječnom korisniku računala. U nekoliko koraka potrebno je pratiti upute te odabratи željenu lokaciju i direktorij na računalu gdje će se aplikacija nalaziti. Prilikom instalacije moguć je odabir i dodatnog sadržaja u obliku raznih dodataka napravljenih u *Pythonu* koji mogu proširiti njegove mogućnosti.

2.7. *Python*

Python je dinamički programski jezik opće namjene, te se može koristiti u različite svrhe razvoja softvera. Dizajn *Pythona* naglašava čitljivost koda kako bi povećao produktivnost programera. *Python* prvotno razvija *Guido van Rossum* koji izdaje prvu verziju 1991. godine. Naziv je dobio po *Monty Python* serijalu. Danas *Python* održavaju

grupe volontera širom svijeta te pripada u programske jezike otvorenog koda i specifikacije. *Python* je inspiriran programskim jezicima: *ABC*, *Haskell*, *Java*, *Lisp*, *Icon* i *Perl*. *Python* podržava nekoliko programskih paradigmi uključujući objektno orijentirano i strukturirano programiranje. Također postoji i ograničena podrška za funkcionalno programiranje. *Python* je multiplatformni, interpreterski programski jezik visoke razine (URL 15). Interpreterski programski jezik znači da se izvorni kod prevodi u izvršni u trenutku izvršavanja programa. Interpreterski jezici su multiplatformni, jer kako bi se kod izvršio potrebno je napisati interpreter za neku platformu i operativni sustav (Miler i Odobašić 2010). On je također i minimalistički programski jezik, izgrađen na malom broju osnovnih naredbi koje je potrebno znati kako bi se efektivno pisao kod. Jedno od njegovih vidljivih obilježja je da ne koristi točku-zarez ili zgrade za definiranje blokova koda. Umjesto toga koristi se uvlačenje tj. prazan prostor. Dinamičnost jezika je jedna od njegovih bitnih karakteristika, te za razliku od drugih programskih jezika (*C/C++*, *Java*, itd) nije potrebno deklarirati tip variable prije izvršavanja. Na taj način tip variable je dinamičan, odnosno može se lako promijeniti. Također svaki objekt je dinamičan, što omogućava definiranje novih objekata u trenutku izvršavanja. Osim toga, *Python* za terminator naredbe programa koristi oznaku novi red dok je kod npr. *Java* programskog jezika terminator točka-zarez. Takav pristup ima prednosti i mane. Jer iako je *Python* kod lakše čitati, zbog jasno prepoznatljivih blokova koda, jedna pogreška u formatiranju rezultira pogreškom u programu. Stoga se za formatiranje koda koristi ili razmak ili tabulator, nikako se ne smije koristiti oboje. *Python* standardna biblioteka sadrži velik broj ugrađenih biblioteka i modula, no unatoč tome na Internetu je dostupan još veći broj biblioteka i modula kojima je moguće proširiti njegovu funkcionalnost. Proširivost i jednostavnost omogućile su *Pythonu* široki raspon iskoristivosti, pa ga tako koriste *Youtube*, *Google* itd. Moguće ga je koristiti za skriptiranje *ArcGIS* programskog paketa ili *AutoCAD* aplikacije kroz *IronPython*. Može se reći da ne donosi neke nove revolucionarne značajke u programiranju, već na optimalan način ujedinjuje sve najbolje ideje i načela rada drugih programskih jezika (Miler i Odobašić 2010). Zbog toga se *Python* koristi za izgradnju web aplikacija, obradu podataka, interakciju s bazama podataka, razvoj korisničkih sučelja, skriptiranje procesa, vizualizacije itd. Trenutno postoji dvije glavne verzije *Pythona*, to su *Python 2.x* i *Python 3.x*. *Python 3.x* prekida kompatibilnost s prethodnim *Python* izdanjima jer nastoji ispraviti neke dizajnerske nedostatke jezika te ga

učiniti još čišćim. Kako je većina koda i dalje pisana za *Python 2.x*, bit će potrebno određeno vrijeme dok softverska baza i programeri ne prijeđu na *Python 3.x*. *Python* je idealan programski jezik za one koji žele naučiti programirati što je uz njegov otvoreni kod jedan od razloga zašto je odabran za korištenje u ovom radu. Treba istaknuti da je u sklopu ovog rada umjesto samostalne verzije *Pythona* korištena njegova verzija u sastavu *OSGeo4W* programskog paketa.

2.8. *PyQt4*

PyQt4 je skup *Python* modula koji služe za stvaranje grafičkih korisničkih sučelja (eng. *GUI*) raznih aplikacija. On omogućava spoj *Python* programskog jezika s *Qt* bibliotekom.

2.8.1. Općenito o *PyQt4*

Qt je jedna od najmoćnijih biblioteka za izradu grafičkih korisničkih sučelja, čiji razvoj potpomaže *Nokia*. *Qt* je napravljen s ciljem da ubrza razvojni ciklus aplikacija (URL 16). Neka grafička korisnička sučelja napravljena pomoću *Qt-a* su *Google Earth*, *Skype*, *Autodesk*, *Quantum GIS* itd. *PyQt4* je implementiran kao skup *Python* modula, ima više od 300 klasa i skoro 6000 funkcija i metoda. On je multiplatformni alat, radi na svim glavnim operativnim sustavima uključujući *Unix*, *Windows* i *Mac*. *PyQt4* je dvostruko licenciran, programeri mogu birati između korištenja pod Općom javnom licencom ili komercijalnom licencem. Prije je Opća javna licenca bila dostupna jedino na *Unix* operativnom sustavu, no počevši od *PyQt* verzije 4 ona je dostupna na svim podržanim platformama (URL 17). Budući da je unutar *PyQt4-a* dostupan veliki broj klasa, one su raspoređene u nekoliko glavnih modula: *QtCore*, *QtGui*, *QtNetwork*, *QtXml*, *QtSvg*, *QtOpenGL*, *QtSql*, *QtMultimedia*, *QtDesigner* i *Qt*. Modul *QtCore* sadrži osnovne negrafičke funkcionalnosti poput rada s vremenom, datotekama i direktorijima, raznim tipovima podataka, *URL-ovima* i procesima. *QtGui* modul sadrži grafičke komponente i srodne klase. Tu pripadaju grafički elementi poput tipki, okvira, statusnih traka, alatnih traka, kliznika, boja, fontova itd. *QtNetwork* modul sadrži klase za mrežno programiranje. Te klase omogućuju pisanje *TCP/IP* i *UDP* klijenata i poslužitelja te čine mrežno programiranje lakšim i prenosivim. *QtXml* sadrži klase potrebne za rad s *XML* datotekama. *QtSvg* modul sadrži klase za prikaz sadržaja *SVG* datoteka. *SVG* (eng. *Scalable Vector*

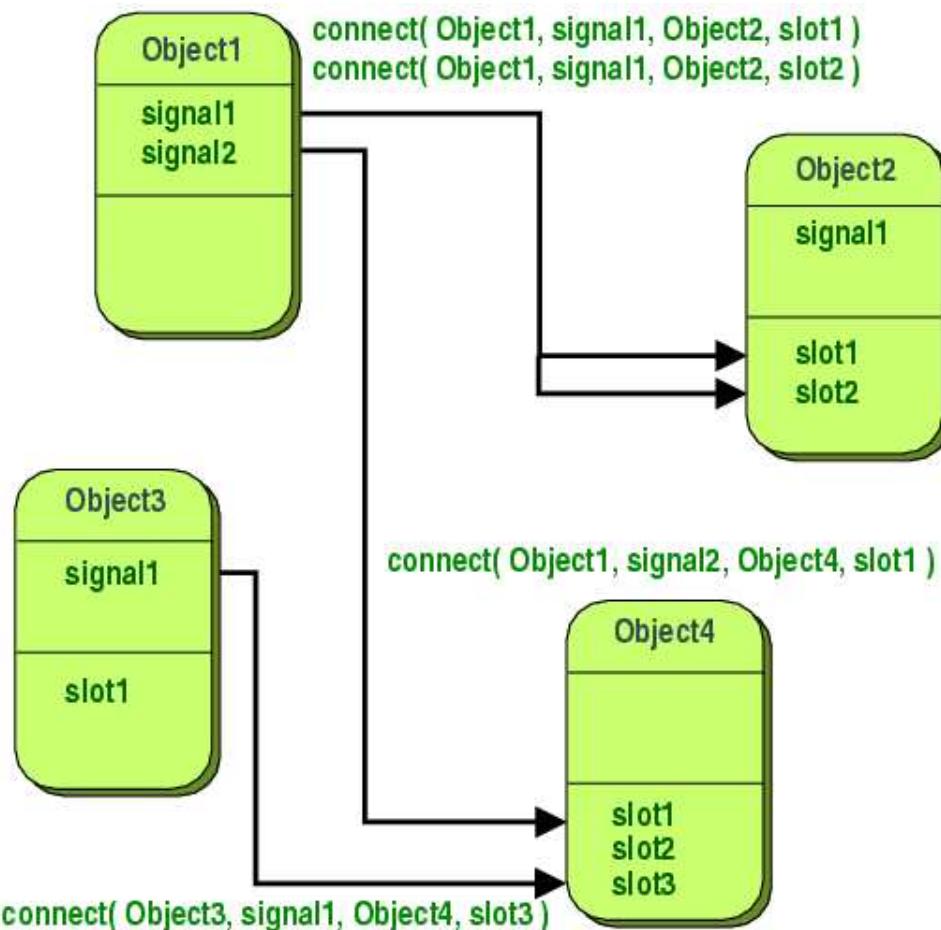
Graphics) je jezik koji služi za opisivanje dvodimenzionalne grafike i grafičkih aplikacija u *XML-u*. *QtOpenGL* modul sadržava *OpenGL* biblioteku te služi za prikaz 2D i 3D grafike, on omogućuje jednostavnu integraciju *QtGui* i *OpenGL* biblioteke. *QtSql* modul sadrži klase za rad s bazama podataka. Modul *QtMultimedia* sadrži osnovne multimedijске funkcionalnosti. *QtDesigner* modul sadrži klase koje omogućuju proširenje *Qt Designer* aplikacije pomoću *PyQt-a*. *Qt* modul objedinjuje sve klase iz prethodno opisanih modula u jedan modul. Prednost toga je što se ne mora brinuti o tome koji temeljni modul sadrži koju klasu, no nedostatak je što se učitava cijelo *Qt* okruženje čime se povećava količina radne memorije potrebne za rad aplikacije (URL 18). Od svih navedenih *PyQt4* modula u ovom radu su korištena dva: *QtCore* i *QtGui* sa svojim pripadajućim klasama, metodama i funkcijama.

Treba napomenuti kako za *PyQt4* postoji dobra podrška u obliku dokumentacije i primjera aplikacija koji se mogu pronaći na Internetu. Pomoću tih izvora je postupak upoznavanja s *PyQt4* bio znatno lakši i brži što je rezultiralo brzom izradom grafičkog korisničkog sučelja za *FuelGIS* aplikaciju. Treba istaknuti da je u ovom radu umjesto samostalne verzije *PyQt4* korištena verzija koja je dio *OSGeo4W* programskog paketa.

2.8.2. Mehanizam signala i slotova

Uz grafičke elemente sučelja jedno od glavnih obilježja *Qt-a* je mehanizam signala i slotova koji se koristi za komunikaciju između objekata. Primjerice, ako korisnik klikne na gumb *Close* vjerojatno želi da se pozove pripadajuća funkcija tog okvira, odnosno da se taj okvir zatvori. Kada se dogodi određeni događaj emitira se signal. *Qt* grafički elementi imaju mnoge unaprijed definirane signale, koje je moguće proširiti te im dodati vlastite signale. Slot (prijemnik, utor) je funkcija koja se poziva kao odgovor na određeni signal (URL 19). *Qt* grafički elementi imaju mnoge unaprijed definirane slotove, ali je uobičajena praksa da se klase proširuju i da im se dodaju vlastiti slotovi koji su potrebni. Kada se signal emitira, svi slotovi koji su spojeni na njega obično se izvršavaju odmah, baš kao obični pozivi funkcija. Kada se taj proces pokrene mehanizam signala i slotova je potpuno neovisan o prikazivanju grafičkog korisničkog sučelja i procesima vezanim uz njega. Ukoliko je nekoliko slotova povezano na jedan signal, slotovi će se izvršavati jedan za drugim u proizvoljnem redoslijedu nakon što se signal emitira. Mehanizam signala i

slotova je tako zamišljen da argumenti signala moraju odgovarati argumentima slot-a. Zapravo, slot može imati manji broj argumenata od signala kojeg prima jer slot može ignorirati dodatne argumete. Budući da su argumenti kompatibilni, prevodilac (eng. *compiler*) može pomoći otkriti nepoklapanja. Kako su signali i slotovi potpuno neovisni: klasa, koja emitira signal ne zna, a u principu joj to nije ni važno, tko prima signal. Taj *Qt* mehanizam osigurava da se slot pozove u pravo vrijeme s parametrima signala. Sve klase koje nasleđuju klasu *QObject* ili neku od njenih podklasa (npr. *QWidget*) mogu sadržavati signale i slotove. Kao što objekt ne zna prima li netko njegov signal, slot ne zna je li neki signal spojen na njega. To osigurava potpunu neovisnost komponenti stvorenih pomoću *Qt-a*. Na jedan slot može se spojiti signala koliko god se želi, a signal može biti spojen s onoliko slotova koliko je potrebno (Slika 1.). Čak je moguće spojiti signal izravno na drugi signal, na taj način moguće je emitirati drugi signal odmah nakon prvoga. Sve navedeno čini *Qt* moćnim alatom za izradu sučelja.



Slika 1. Princip rada signala i slotova (URL 30)

2.9. *PyQGIS*

PyQGIS je aplikacijsko programsko sučelje koje služi za stvaranje aplikacija temeljenih na *Quantum GIS (QGIS)* platformi, ono je spoj *Python* programskog jezika i *QGIS* biblioteka. Aplikacijsko programsko sučelje (eng. *application programming interface, API*) ili sučelje za programiranje aplikacija je skup određenih pravila i specifikacija koje programeri slijede kako bi se mogli služiti uslugama ili resursima operacijskog sustava, nekog drugog složenog programa, strukture podataka, objekta ili protokola (URL 20). Korištenje aplikacijskog programskog sučelja omogućava programerima korištenje rada drugih programera štедеći vrijeme i trud koji je potreban da se napiše neki složeni program, pri čemu svi programeri koriste iste standarde. Napretkom u operacijskim sustavima, osobito napretkom u grafičkom korisničkom sučelju, aplikacijsko programsko sučelje je postalo nezaobilazno u stvaranju novih aplikacija. Umjesto da se novi programi pišu iz temelja, programeri nastavljaju na radu drugih. Nažalost, trenutačno ne postoji dokumentacija napisana za *PyQGIS*, što predstavlja ozbiljnu prepreku za izradu aplikacija temeljenih na *QGIS* platformi. No postoji dokumentacija za *QGIS* aplikacijsko programsko sučelje napisana u C++ programskom jeziku. Treba napomenuti da je aplikacijsko programsko sučelje za *Python* gotovo identično aplikacijskom programskom sučelju za C++. Stoga je za izradu *FuelGIS* aplikacije korištena dokumentacija napisana za C++ dostupna na Internetu (URL 21).

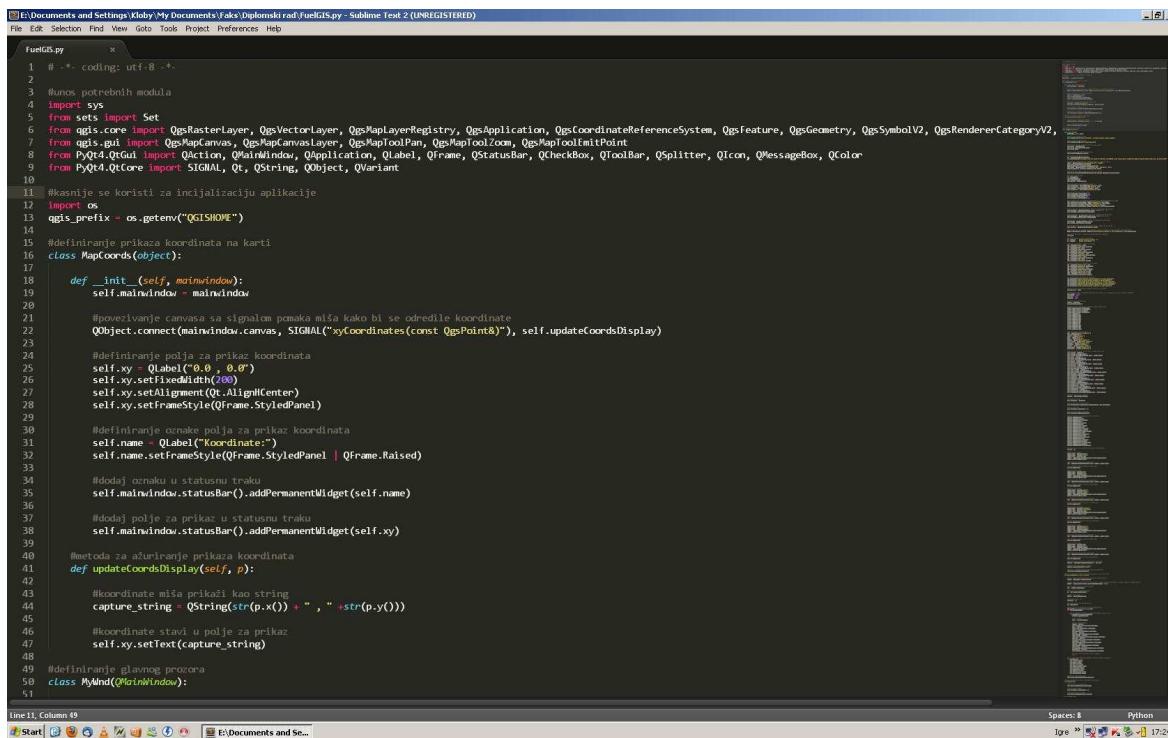
PyQGIS se sastoji od velikog broja klasa, funkcija i metoda koje su raspoređene u nekoliko modula tj. biblioteka: *QGIS core*, *QGIS gui*, *QGIS analysis*, *QGIS network analysis* i *MapComposer*. *QGIS core* modul sadrži sve klase potrebne za osnovne *GIS* funkcionalnosti poput koordinatnog sustava, rasterskih i vektorskih slojeva, objekata, geometrije, simbologije itd. Ukratko, taj modul je osnova svake aplikacije temeljene na *Quantum GIS* platformi te bez njega nije moguće napraviti aplikaciju. *QGIS gui* modul temeljen je na biblioteci *QGIS core* te sadrži klase potrebne za stvaranje elemenata (eng. *widgets*) koji se mogu staviti u grafičko korisničko sučelje. U te dodatke za višestruku upotrebu pripadaju alati za pomicanje (eng. *pan*), približavanje (eng. *zoom in*), udaljavanje (eng. *zoom out*) i emitiranje položaja na karti. Takoder tu pripada i podloga za iscrtavanje karte (eng. *map canvas*) koja je neophodna ukoliko se želi u aplikaciji temeljenoj na *Quantum GIS* platformi prikazati bilo kakav sloj podataka. Biblioteka *QGIS analysis* je

napravljena na temelju biblioteke *QGIS core*, ona pruža vrhunske alate za obavljanje prostornih analiza na vektorskim i rasterskim podacima. *QGIS network analysis* biblioteka sadrži alate potrebne za izradu i analizu topologije. *MapComposer* sadrži klase potrebne za izradu skupa karata pomoću elemenata kao što su karta, legenda, mjerilo, prikaz itd. Od svih navedenih *PyQGIS* modula i biblioteka u ovom radu su korištena dva: *QGIS core* i *QGIS gui* sa svojim pripadajućim klasama, metodama i funkcijama. Ti moduli, za razliku od modula kao što su *Shapely*, *Matplotlib* itd, nisu dostupni zasebno nego dolaze zajedno s *Quantum GIS* aplikacijom.

Treba napomenuti kako za *PyQGIS* postoji jako slaba podrška, uz dokumentaciju za *QGIS* aplikacijsko programsko sučelje u C++ pronađeno je i korišteno svega 2 do 3 izvora na Internetu. To je rezultiralo sporijim i težim upoznavanjem s mogućnostima *PyQGIS-a* te samim time i sporijim razvojem *FuelGIS* aplikacije tj. sporijom implementacijom funkcionalnosti *QGIS-a* u samu aplikaciju. Stoga ovaj rad može poslužiti nekim budućim korisnicima kao dokumentacija za korištenje mogućnosti *PyQGIS-a* tj. kao pomoć programerima prilikom razvoja njihovih aplikacija.

2.10. Sublime Text 2

Postoji puno uređivača teksta i razvojnih okruženja koji omogućavaju uređivanje *Python* koda. U ovom radu za pisanje koda *FuelGIS* aplikacije korišten je uređivač teksta *Sublime Text 2*. *Sublime Text 2* je moguće slobodno preuzeti s Interneta (URL 22) i testirati ga, međutim, korištenje se mora platiti. No treba napomenuti da trenutačno nema ograničenja vremenskog roka za testiranje. Također, uređivač je još uvijek u razvoju i testiranju te je trenutačno dostupan u beta verziji. Osim atraktivnog sučelja pruža velike mogućnosti za uređivanje teksta. Ima mogućnost rada s više datoteka istodobno pomoći tabulatora, pomoći višestrukog odabira moguće je napraviti višestruke promjene odjednom umjesto svake pojedinačno. Podrška za isticanje sintakse podržava velik broj jezika: *Java*, *Python*, *SQL*, *HTML*, *C*, *C++*, *Perl*, *XML* itd. Posjeduje mogućnost umanjenog prikaza u kojem je moguće vidjeti sav napisani kod (Slika 2.) te bogati izbor naredbi za uređivanje teksta uključujući uvlačenje/izvlačenje teksta, komentiranje blokova teksta itd. Uz sve to je jednostavan i brz te je dostupan u verzijama za *Linux*, *Windows* i *Mac OS X*. Instalacija aplikacije je krajnje jednostavna a rad s njom jako ugodan.



```

File Edit Selection Find View Go Tools Project Preferences Help
FuelGIS.py
1  # -*- coding: utf-8 -*-
2
3 #unos potrebnih modula
4 import sys
5 from sets import Set
6 from qgis.core import QgsRasterLayer, QgsVectorLayer, QgsMapLayerRegistry, QgsApplication, QgsCoordinateReferenceSystem, QgsFeature, QgsGeometry, QgsSymbolV2, QgsRendererCategoryV2,
7 from qgis.gui import QgsMapCanvas, QgsMapCanvasLayer, QgsMapToolPan, QgsMapToolZoom, QgsMapToolEditPoint
8 from PyQt4.QtGui import QAction, QMainWindow, QLabel, QFrame, QStatusBar, QToolBar, QSplitter, QIcon, QMessageBox, QColor
9 from PyQt4.QtCore import SIGNAL, QString, QVariant
10
11 #kasnije se koristi za inicijalizaciju aplikacije
12 import os
13 qgis_prefix = os.getenv("QGISHOME")
14
15 #definiranje prikaza koordinata na karti
16 class MapCoordinates(object):
17
18     def __init__(self, mainwindow):
19         self.mainwindow = mainwindow
20
21     #povezivanje canvasa sa signalom pomaka miša kako bi se odredile koordinate
22     Qobject.connect(mainwindow.canvas, SIGNAL("xyCoordinates(const QgsPoint&)"), self.updateCoordsDisplay)
23
24     #definiranje polja za prikaz koordinata
25     self.xy = QLabel("0.0 , 0.0")
26     self.xy.setFixedWidth(100)
27     self.xy.setAlignment(Qt.AlignCenter)
28     self.xy setFrameStyle(QFrame.StyledPanel)
29
30     #definiranje oznake za prikaz koordinata
31     self.name = QLabel("Koordinate:")
32     self.name setFrameStyle(QFrame.StyledPanel | QFrame.Raised)
33
34     #dodataj oznaku u statusnu traku
35     self.mainwindow.statusBar().addPermanentWidget(self.name)
36
37     #dodataj polje za prikaz u statusnu traku
38     self.mainwindow.statusBar().addPermanentWidget(self.xy)
39
40     #metoda za ažuriranje prikaza koordinata
41     def updateCoordsDisplay(self, p):
42
43         #koordinate miša prikaži kao string
44         capture_string = QString(str(p.x()) + " , " + str(p.y()))
45
46         #koordinate stavi u polje za prikaz
47         self.xy.setText(capture_string)
48
49     #definiranje glavnog prozora
50 class MyWindow(MainWindow):
51
Line 11, Column 49
File Start Stop Maximize E:\Documents and Se...
Spaces: 8 Python
Type: 17:24

```

Slika 2. Pisanje koda u Sublime Text 2

3. Prikupljanje podataka

Prikupljanje podataka obuhvaćalo je pronalazak i preuzimanje odgovarajuće rasterske podloge te položajnih i atributnih podataka o pojedinim benzinskim postajama na području Grada Zagreba.

3.1. Rasterska podloga

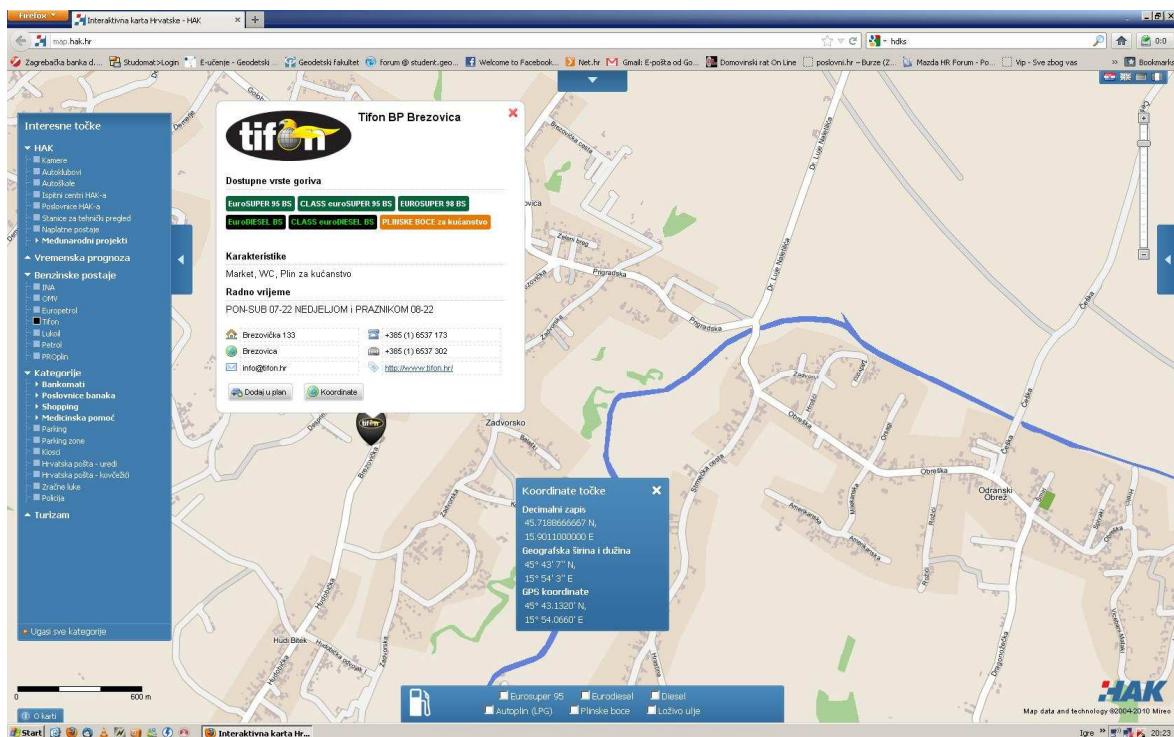
U suradnji s Gradskim uredom za strategijsko planiranje i razvoj Grada dobiven je na korištenje digitalni ortofoto (DOF) Grada Zagreba iz 2006. godine. Navedeni DOF je bio u *jpg* formatu zapisa datoteke s nazivom *Zagreb*. Dimenzije te *jpg* datoteke su bile 15300 x 16800 piksela, a uz nju je dobivena i pripadajuća *jgw* datoteka s podacima o početnoj koordinati DOF-a te veličini piksela. Treba naglasiti da je DOF 2006 napravljen u Hrvatskom državnom koordinatnom sustavu (HDKS) s Besselovim 1841 elipsoidom te Gauss-Krügerovom projekcijom. Navedeni koordinatni sustav zbog svoje zastarjelosti više nije u službenoj upotrebi.

3.2. Položajni i atributni podaci

Prikupljanje položajnih i atributnih podataka o benzinskim postajama obavljeno je putem Interneta. Kao primarni izvor tih podataka korištena je HAK-ova interaktivna karta Republike Hrvatske (URL 23). HAK-ova interaktivna karta sadrži informacije o benzinskim postajama šest najvećih naftnih kompanija u Republici Hrvatskoj: Tifon d.o.o. (URL 24), OMV Hrvatska d.o.o. (URL 25), Lukoil Hrvatska d.o.o. (URL 26), Euro – PETROL d.o.o. (URL 27), Petrol Hrvatska d.o.o. (URL 28) i INA – Industrija nafte d.d. (URL 29). Kao sekundarni izvori podataka korištene su web stranice tih naftnih kompanija. S obzirom da su navedene naftne kompanije vlasnici položajnih i atributnih podataka traženo je njihovo dopuštenje za korištenje istih. Treba napomenuti da neke kompanije nisu odgovorile na zahtjev za korištenjem podataka. Unatoč tome preuzeti su potrebni podaci svih kompanija jer su korišteni u akademske, nekomercijalne, svrhe te stoga nije bilo potrebno njihovo dopuštenje.

Preuzimanje podataka s prethodno navedenih web stranica napravljeno je ručno za svaku benzinsku postaju. Na HAK-ovoj interaktivnoj karti odabrana je željena naftna kompanija,

a zatim željena benzinska postaja. Odabirom benzinske postaje prikazale bi se detaljne informacije o njoj (Slika 3.). Za svaku benzinsku postaju prikupljeni su položajni podaci u obliku geografskih koordinata. Treba napomenuti kako su koordinate bile izražene u WGS84 koordinatnom sustavu. Provjera njihove točnosti napravljena je pomoću *Google Earth* aplikacije, pritom je utvrđeno da većina benzinskih postaja ima točne koordinate.



Slika 3. Preuzimanje podataka s HAK-ove interaktivne karte

Za manji broj su utvrđena položajna odstupanja različitih veličina, u pojedinim slučajevima odstupanja su iznosila i do 150 m. Slika 4. prikazuje jedan takav slučaj, radi se o koordinatama benzinske postaje OMV BS Zagreb – Branimirova. Na slici je vidljivo da unesene koordinate ukazuju na stambenu zgradu na lijevoj strani (označeno plavim križem i koordinatama) dok je stvarni položaj benzinske postaje označen crvenom točkom. Pritom je zamjetno veliko odstupanje između te dvije lokacije. Budući da se nisu mogla zanemariti, takva odstupanja su uklonjena očitavanjem ispravnih koordinata benzinskih postaja pomoću *Google Earth* aplikacije. Na taj način su dobiveni točni položajni podaci koji su neophodni za svaki geografski informacijski sustav (*GIS*).



Slika 4. Google Earth – primjer položajnog odstupanja benzinske postaje

Uz položajne, za svaku benzinsku postaju, preuzeti su s HAK-ove interaktivne karte i atributni podaci. Prilikom preuzimanja atributnih podataka vršena je provjera njihove točnosti pomoću podataka dostupnih na web stranicama matičnih naftnih kompanija. Pritom su otkrivene razlike između atributnih podataka prikazanih na HAK-ovoj interaktivnoj karti te atributnih podataka na web stranicama matičnih naftnih kompanija. Navedeno se uglavnom odnosilo na radno vrijeme pojedinih benzinskih postaja te dostupnost autopraonice. U pojedinim slučajevima su primijećene neispravne adrese, npr. benzinske postaje smještene na Zagrebačkoj aveniji imale su stare adrese (Ljubljanska avenija). Također, na HAK-ovoj interaktivnoj karti nalazile su se benzinske postaje koje su u međuvremenu zatvorene, poput benzinske postaje INA BP Zagreb – Harambašićeva. Uzrok svih tih razlika je slaba ažurnost HAK-ove interaktivne karte u odnosu na web stranice matičnih naftnih kompanija. Stoga su u takvim situacijama kao točni uzimani atributni podaci s web stranica matičnih naftnih kompanija. Treba naglasiti da su sva navedena zapažanja uzeta u obzir prilikom preuzimanja podataka te je pritom uložen dodatni trud kako bi prikupljeni atributni podaci bili što točniji tj. ažurniji. Njihova točnost

i ažurnost također utječe na kvalitetu i upotrebljivost svakog *GIS-a*. Tablica 3. sadrži primjer prikupljenih položajnih i atributnih podataka, njihove opise i vrijednosti.

Naziv	Opis	Vrijednost	Primjer
BenzID	Jedinstveni identifikator	Integer	2
X	Geografska širina (ϕ)	Float	45,75901667
Y	Geografska dužina (λ)	Float	15,90176667
Vlasnik	Vlasnik benzinske postaje	String	Tifon
Naziv	Naziv benzinske postaje	String	Tifon BP Lučko
Adresa	Adresa benzinske postaje	String	Blato bb
Broj telefona	Telefonski broj benzinske postaje	String	38516522201
Radno vrijeme	Radno vrijeme benzinske postaje	String	PON-NED 00-24
Eurosuper 95	Dostupnost eurosuper 95 goriva	String	Da
Eurodiesel	Dostupnost eurodiesel goriva	String	Da
Autoplín	Dostupnost autoplína	String	Da
Plinske boce	Dostupnost plinskih boca	String	Ne
Autopraonica	Dostupnost autopraonice	String	Ne

Tablica 3. Popis prikupljenih atributa za svaku benzinsku postaju

Nakon provjere točnosti svi položajni i atributni podaci pohranjeni su u *CSV* formatu zapisa datoteke pomoću *Microsoft Excela*. Slika 5. prikazuje unos položajnih i atributnih podataka u *CSV* datoteku s nazivom *benz*. *Benz* datoteka sadrži 13 stupaca koji predstavljaju 13 različitih atributa za svaku benzinsku postaju (Tablica 3.). *Benz* datoteka sadrži položajne i atributne podatke o 80 benzinskih postaja na području Grada Zagreba.

Treba napomenuti da se podaci o benzinskim postajama odnose na šest najvećih naftnih kompanija u Republici Hrvatskoj zbog njihove dostupnosti na HAK-ovoj interaktivnoj karti.

Slika 5. Unos položajnih i atributnih podataka u CSV datoteku

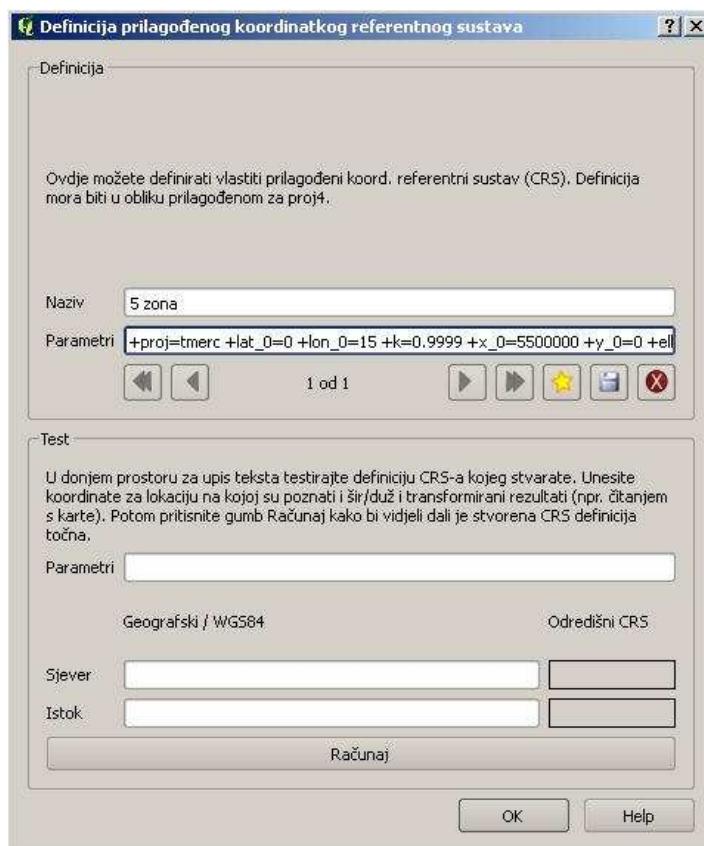
Podaci o benzinskim postajama ostalih, manjih naftnih kompanija nisu obuhvaćeni prilikom prikupljanja podataka zbog svoje nedostupnosti. Izradom *benz* datoteke u CSV formatu završena je prva faza diplomskog rada, prikupljanje podataka.

4. Obrada podataka

Obrada podataka sastojala se od daljnog rada na prikupljenim položajnim i atributnim podacima te rasterskoj podlozi. Cilj ove, druge, faze diplomskog rada bio je priprema prikupljenih podataka za korištenje u *FuelGIS* aplikaciji. Budući da se *FuelGIS* temelji na *Quantum GIS (QGIS)* platformi, veći dio obrade podataka izvršen je upravo pomoću *Quantum GIS* aplikacije.

4.1. Izrada koordinatnog sustava

Zbog različitih koordinatnih sustava rasterske podloge (HDKS) te prikupljenih položajnih podataka (*WGS84*) nije ih bilo moguće zajedno smjestiti u prostor. Stoga je bilo potrebno napraviti transformaciju tih podataka u jedinstveni koordinatni sustav. Odlučeno je da se pomoću *QGIS-a* napravi transformacija položajnih podataka iz *WGS84* u Hrvatski državni koordinatni sustav (HDKS).



Slika 6. Dijaloški okvir za definiranje koordinatnog sustava

Budući da *QGIS* nije sadržavao definiciju HDKS trebalo ga je stvoriti, što je i napravljeno pomoću parametara dostupnih na Internetu (URL 30). Na glavnoj alatnoj traci *QGIS-a* odabere se izbornik *Postavke* zatim opcija *Prilagođeni CRS*. Nakon toga se otvara dijaloški okvir za definiranje koordinatnog sustava (Slika 6.). Za stvaranje novog koordinatnog sustava klikne se gumb *Novo*, zatim se unese naziv koordinatnog sustava te njegovi parametri. Kako bi se ispravno definirao koordinatni sustav parametri moraju biti izraženi u *PROJ.4* formatu. Nakon toga potrebno je spremiti definirani koordinatni sustav pomoću gumba *Spremi*. Novi koordinatni sustav spremljen je pod nazivom *5 zona*. Navedeni postupak je opisan na engleskom jeziku u uputama za korištenje *Quantum GIS-a* (URL 31).

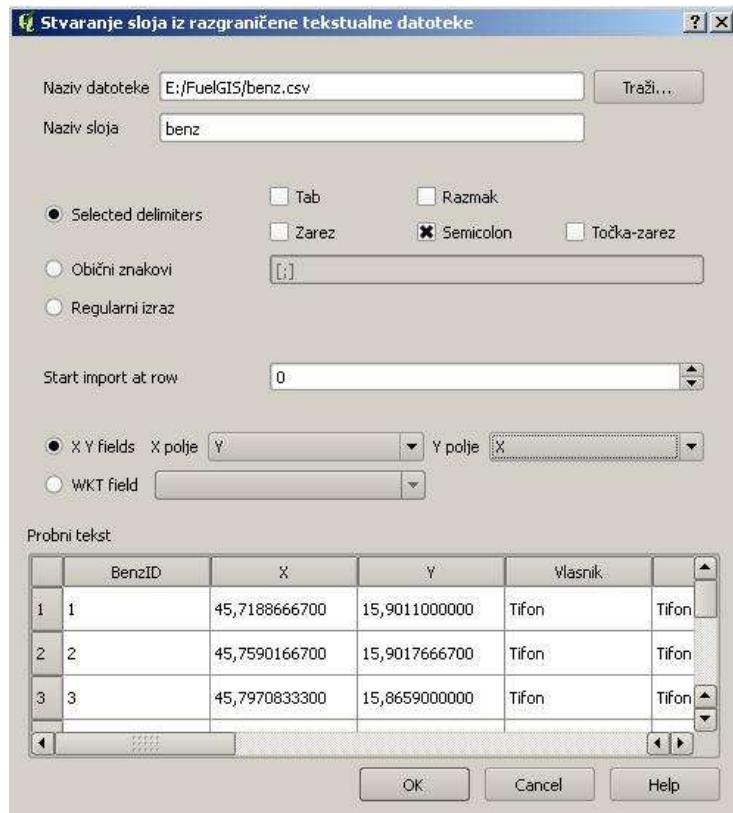
4.2. *Stvaranje sloja iz CSV datoteke*

Sljedeći korak bio je dodavanje prethodno stvorene *CSV* datoteke u *QGIS*, to je također napravljeno prema uputama za korištenje *Quantum GIS-a* (URL 31). Pritom trebaju biti ispunjeni određeni uvjeti:

- prva linija tekstualne datoteke mora biti zaglavljena s razgraničenim nazivima polja,
- redak zaglavlja mora sadržavati X i Y polja koja mogu imati bilo koji naziv,
- x i y koordinate moraju biti u obliku brojeva, pritom koordinatni sustav nije važan.

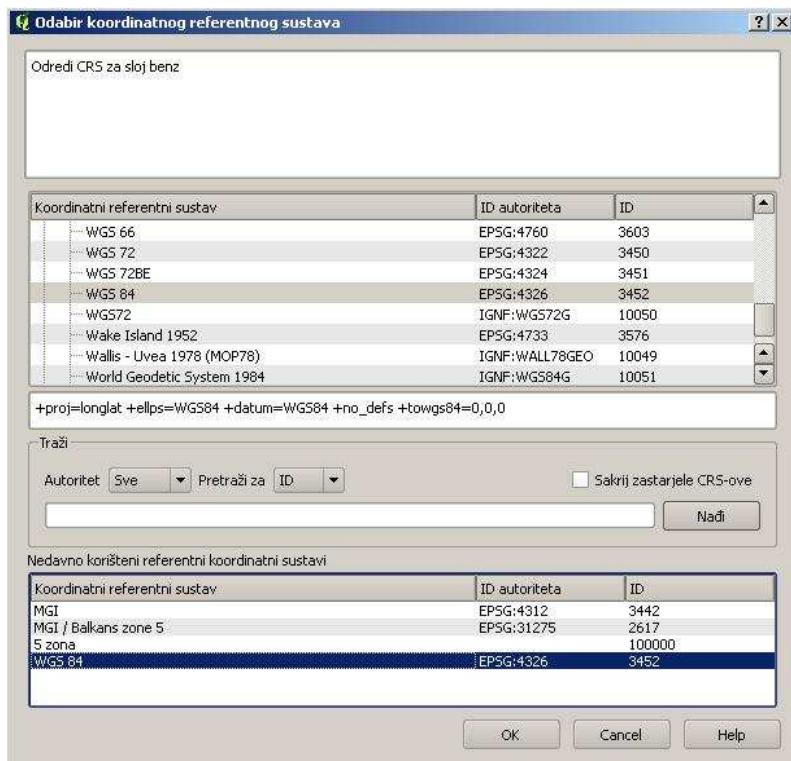
Za dodavanje *CSV* datoteke odabere se na glavnoj alatnoj traci *QGIS-a* izbornik *Sloj* zatim opcija *Dodaj sloj razgraničenog teksta*. Nakon toga otvara se dijaloški okvir *Stvaranje sloja iz razgraničene tekstualne datoteke* (Slika 7.). Prvo se odabere željena *CSV* datoteka klikom na gumb *Traži* te se upiše naziv sloja u predviđeno polje. Nakon odabira datoteke ona se obrađuje tj. analizira se njena sintaksa pomoću zadnje korištenog graničnika. Kako bi se datoteka pravilno obradila potrebno je odabrati ispravni graničnik, na raspolaganju stoji više graničnika poput zareza, točke-zareza, dvotočke, razmaka, itd. U ovom slučaju kao graničnik je korištена točka-zarez. Treba napomenuti kako za ispravan odabir točke zareza kao graničnika treba odabrati opciju *Semicolon* umjesto opcije *Točka-zarez*. Uzrok

tome je krivi prijevod napravljen od strane razvojne zajednice. Odabirom opcije *Točka-zarez* zapravo se za graničnik odabire dvotočka.



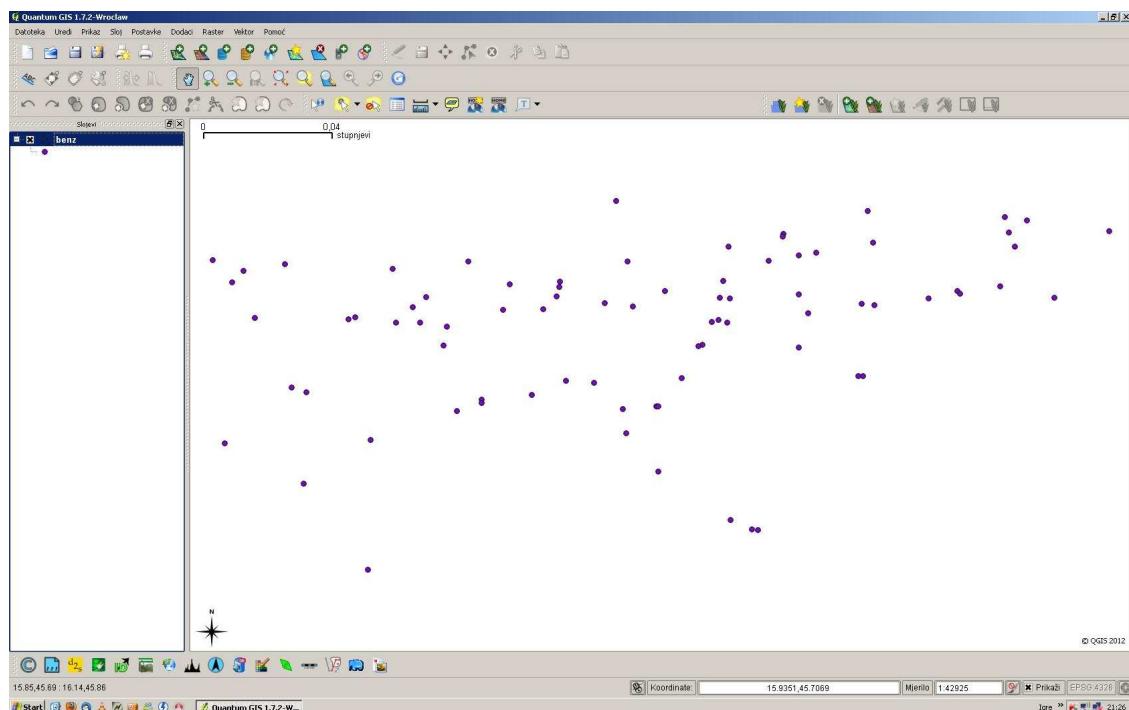
Slika 7. Dijaloški okvir za stvaranje sloja iz CSV datoteke

Nakon obrade odabrana su X i Y polja, pritom treba naglasiti da je za X polje odabранo polje iz CSV datoteke s nazivom Y, a za Y polje s nazivnom X. Uzrok takvog odabira je sadržaj koordinatnih polja iz CSV datoteke. X polje sadrži podatke o geografskoj širini (ϕ), a Y polje o geografskoj dužini (λ). U suprotnom bi položaji benzinskih postaja bili krivo smješteni u prostoru. Za završetak dodavanja CSV datoteke kao novog sloja s nazivom *benz* u *Quantum GIS* odabere se opcija *OK*. Na taj način se CSV datoteka ponaša i prikazuje kao bilo koji drugi sloj u *QGIS-u*. Prije prikaza tog sloja otvara se dijaloški okvir *Odabir koordinatnog referentnog sustava* (Slika 8.) u kojem je potrebno odabrati koordinatni sustav u kojem se sloj prikazuje. Budući da su prikupljeni položajni podaci u *WGS84* potrebno je za ispravan prikaz odabrati taj koordinatni sustav.



Slika 8. Odabir koordinatnog sustava za sloj benz

Slika 9. prikazuje sloj *benz*, učitan u *QGIS-u*, nastao iz istoimene *CSV* datoteke.

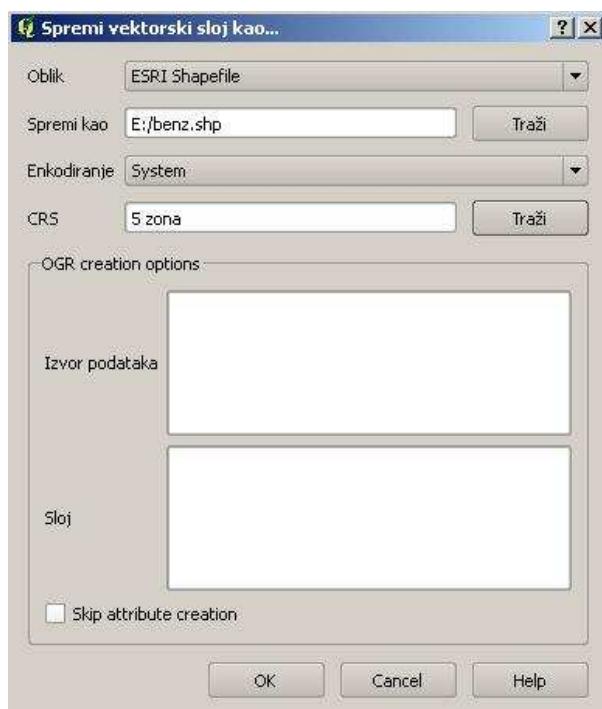


Slika 9. Sloj benz učitan u QGIS-u

Sloj *benz* prikazuje položaje svih 80 benzinskih postaja na području Grada Zagreba u pripadajućem koordinatom sustavu (*WGS84*). U idućem koraku je, pomoću *QGIS-a*, iz prethodno stvorenog sloja *benz* napravljena *ESRI Shape* datoteka.

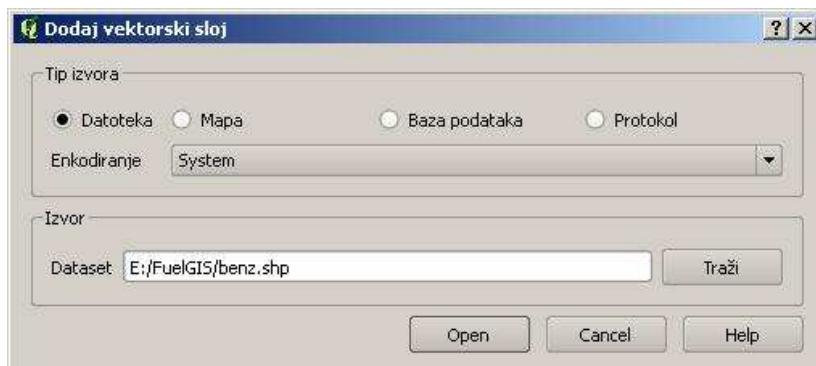
4.3. Izrada *ESRI Shape* datoteke

Postupak izrade *ESRI Shape* datoteke iz postojećeg sloja u *QGIS-u* je krajnje jednostavan. Desnim klikom miša iz popisa prikazanih slojeva odabran je prethodno stvoren sloj s nazivom *benz*. Pritom se otvorio izbornik s popisom opcija tog sloja. Odabirom opcije *Spremi kao*, pojavljuje se dijaloški okvir s nazivom *Spremi vektorski sloj kao* (Slika 10.). U njemu je pod opcijom *Oblik* odabran *ESRI Shapefile* te je unesen budući naziv *ESRI Shape* datoteke (*benz*). Pod opcijom *CRS* za koordinatni sustav *ESRI Shape* datoteke odabran je ručno napravljen HDKS (prethodno stvoren pod nazivom *5 zona*). Tim odabirom je uz stvaranje *ESRI Shape* datoteke omogućena i transformacija položajnih podataka iz *WGS84* u *HDKS*. Ostale postavke u dijaloškom okviru nisu mijenjane te je odabirom opcije *OK* stvorena *ESRI Shape* datoteka. Tom prilikom uz 3 obavezne (*shp*, *shx* i *dbf*) dobivene su i dvije optionalne datoteke (*prj* i *qpj*) s podacima o koordinatnom sustavu i projekcijskim parametrima.



Slika 10. Stvaranje *ESRI Shape* datoteke

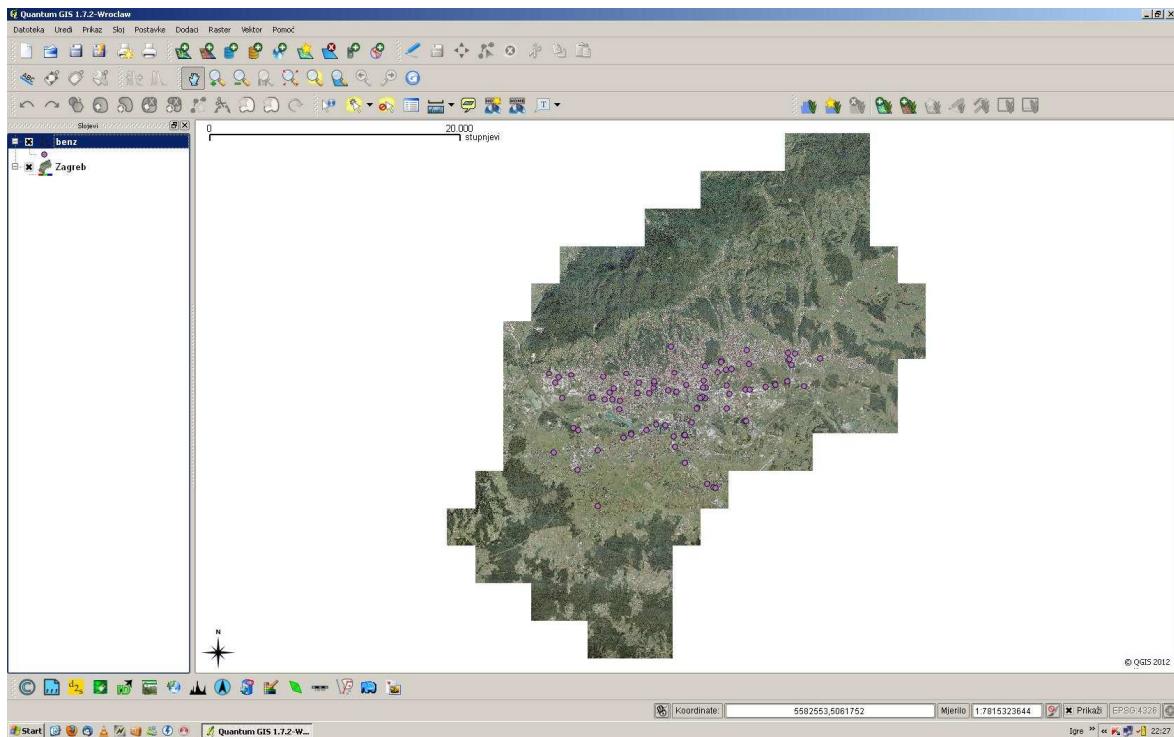
Na taj način su svi prikupljeni podaci (rasterska podloga i položaji benzinskih postaja) izraženi u jedinstvenom koordinatnom sustavu (HDKS) te ih je bilo moguće zajedno smjestiti u prostor. Preklapanje podataka je također napravljeno u *QGIS-u* učitavanjem rasterske podloge (DOF 2006) te vektorskog sloja (*ESRI Shape* datoteka). Za dodavanje rasterske podloge odabere se na glavnoj alatnoj traci *QGIS-a* izbornik *Sloj* zatim opcija *Dodaj rasterski sloj*. Nakon toga odabere se željena datoteka te koordinatni sustav u kojem je izražena (HDKS stvoren pod nazivom *5 zona*). Na sličan način se dodaje i vektorski sloj, na glavnoj alatnoj traci *QGIS* aplikacije odabere se izbornik *Sloj* zatim opcija *Dodaj vektorski sloj*. Otvara se dijaloški okvir *Dodaj vektorski sloj* (Slika 11.).



Slika 11. Dodavanje vektorskog sloja u QGIS

Na njemu se kao tip izvora odabere *Datoteka* te kao izvor prethodno stvorena *ESRI Shape* datoteka pod nazivom *benz*. Ostale postavke se ne mijenjaju nego se za dodavanje odabere opcija *Open*. Kako je uz *shp* dobivena i *qpj* datoteka *QGIS* ne traži unos koordinatnog sustava nego odmah učitava datoteku. Slika 12. prikazuje učitane rasterske i vektorske slojeve u *QGIS-u*. Na njoj je vidljivo i položajno poklapanje rasterskih i vektorskih podataka tj. koordinate benzinskih postaja se poklapaju s položajem benzinskih postaja na rasterskoj podlozi. Vizualnom provjerom utvrđeno je da neke benzinske postaje nisu vidljive na rasterskoj podlozi makar postoje vektorski podaci za njih (npr. benzinska postaja Tifon BP Jankomir). Provjerom pomoću *Google Eartha* utvrđeno je da ih nema na rasteru jer u vrijeme izrade digitalnog ortofota, 2006. godine, još nisu bile izgrađene nego su napravljene nakon tog perioda. Obzirom da za ovaj rad nisu bile dostupne novije verzije digitalnog ortofota, nisu se mogle ni ukloniti zapažene nepravilnosti. Također, interakcija s rasterskim slojem pomoću *QGIS* alata za pomicanje, približavanje i udaljavanje bila je

spora. Nakon napravljene akcije (npr. pomicanja) bilo je potrebno nekoliko sekundi kako bi se rasterska podloga ponovno iscrtala sukladno njoj.



Slika 12. Prikaz rasterskog i vektorskog sloja u QGIS-u

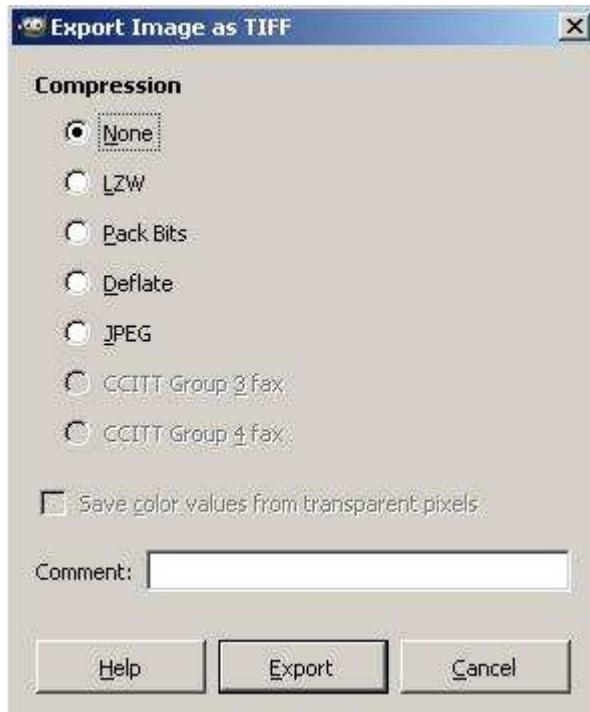
Sljedeća dva potpoglavlja opisuju postupak obrade rastera kako bi se riješio taj problem.

4.4. Obrada rasterske podloge

Kako je DOF 2006 kompresiran u *jpg* datoteku velikih dimenzija (15300 x 16800 piksela) navigacija predstavlja za računalo veliko opterećenje. Zbog toga je odlučeno da se napravi konverzija DOF-a 2006 iz *jpg* u *tiff* format zapisa datoteke.

Konverzija je pokušana s programima poput *Paint.NET* i *IrfanView*. No navedeni nisu podržavali rad s datotekama tako velikih dimenzija, stoga je za konverziju korištena aplikacija *GIMP*. Rad u toj aplikaciji je jednostavan, na glavnoj alatnoj traci *GIMP-a* odabere se izbornik *File* zatim opcija *Open*. Na taj način je u *GIMP* učitana *jpg* datoteka koju je trebalo pretvoriti u *tiff* format. Nakon toga ponovno se odabere izbornik *File* te zatim opcija *Export*, pritom se otvara dijaloški okvir *Export Image*. U njemu se odabere željena lokacija i direktorij za pohranu datoteke te *tiff* format kao format u kojem će

datoteka biti pohranjena. Ostale postavke ostaju nepromijenjene. Zatim se odabere opcija *Export*, pritom se otvara dodatni dijaloški okvir *Export Image as TIFF* (Slika 13.).



Slika 13. GIMP - postavke za konverziju iz jpg u tiff datoteku

U njemu se pod opcijom *Compression* odabere vrijednost *None* te se odabirom opcije *Export* izvrši konverzija. Na taj način dobiven je DOF 2006 u obliku nove *tiff* datoteke. Obzirom da prilikom konverzije *GIMP* nije uz *tiff* datoteku napravio i pripadajuću *tfw* datoteku ona je stvorena ručno iz *jgw* datoteke na način da je toj *jgw* datoteci zamijenjena ekstenzija *jgw* u *tfw*. Ovim postupkom je *tiff* datoteka je dobila pripadajuću *tfw* datoteku s podacima o početnoj koordinati DOF-a te veličini piksela. Navedena *tfw* datoteka omogućuje ispravno smještanje DOF-a 2006 u prostor prilikom učitavanja *tiff* datoteke u prikladnim *GIS* aplikacijama, npr. u *Quantum GIS-u*. Treba napomenuti da veličina novostvorene *tiff* datoteke iznosi 735 MB, za razliku od početne *jpg* datoteke čija je veličina 45 MB. Uzrok tolike razlike u veličini *tiff* datoteke je što ona ne koristi nikakvu kompresiju. Nepostojanje kompresije omogućuje bržu navigaciju po DOF-u nego što je bila prije konverzije.

4.5. Izrada piramide

Kako je brzina navigacije po DOF-u i nakon konverzije u *tiff* bila nezadovoljavajuća pristupilo se daljnjoj optimizaciji te datoteke. Rasteri velike rezolucije mogu usporiti navigaciju unutar *QGIS-a*. Stvaranjem kopija podataka niže rezolucije (piramide) mogu se znatno poboljšati performanse jer *QGIS* prikazuje prikladnu rezoluciju ovisno o blizini prikazivanja. Izrada piramide za *tiff* datoteku napravljena je pomoću *QGIS* aplikacije. Prvo je u *QGIS* dodana *tiff* datoteka kao rasterski sloj, to je obavljeno prema prije opisanom postupku za dodavanje rasterskog sloja. Desnim klikom miša iz popisa prikazanih slojeva odabran je prethodno učitan rasterski sloj s nazivom *Zagreb*. Nakon otvaranja izbornika s popisom opcija za taj sloj odabrana je opcija *Osobine* te se pojavio dijaloški okvir s nazivom *Osobine sloja – Zagreb*. U njemu je odabran tabulator *Piramide* (Slika 14.) koji služi za izradu piramida.



Slika 14. Izrada piramide u *QGIS-u*

U izborniku *Rezolucije piramide* odabrane su sve ponuđene rezolucije, a za metodu resampliranja (interpolacije) odabrana je metoda *Najближи susјед* (eng. *nearest neighbor*), ostale opcije nisu mijenjane. Zatim je odabrana opcija *Izgradi piramide* nakon čega je *QGIS* počeo izradu piramide za *tiff* datoteku. Postupak izrade piramide je zahtjevan zbog veličine *tiff* datoteke te traje određeno vrijeme, ovisno o jačini računala. Na korištenom računalu (dvojezgreni procesor *Core 2 Duo* na 2.4 GHz, 4 GB DDR2 RAM-a te grafička kartica *NVIDIA GeForce 8600M GT* s 512 MB RAM-a) taj postupak traje oko 3 minute. U tom periodu moguće je pratiti putem statusnog polja napredak (izražen u postotcima) u izradi pojedine rezolucije piramide. Nakon završetka izrade piramide odabirom opcije *OK* zatvoren je dijaloški okvir, time je bio gotov postupak obrade *tiff* datoteke. Pritom treba napomenuti da je piramida stvorena u obliku posebne, nove datoteke s *ovr* ekstenzionom. *OVR* datoteka veličine 248 MB sadrži 8 rezolucija piramide te omogućuje zadovoljavajuću brzinu navigacije po *tiff* datoteci. Više detalja o rezultatima obrade rasterske datoteke *Zagreb* dostupno je u tablici 4.

Format datoteke	Veličina datoteke [MB]	Brzina navigacije [s]
jpg	45	5-10
tiff	735	3-7
ovr	248	-
tiff + ovr	735+248	1-2

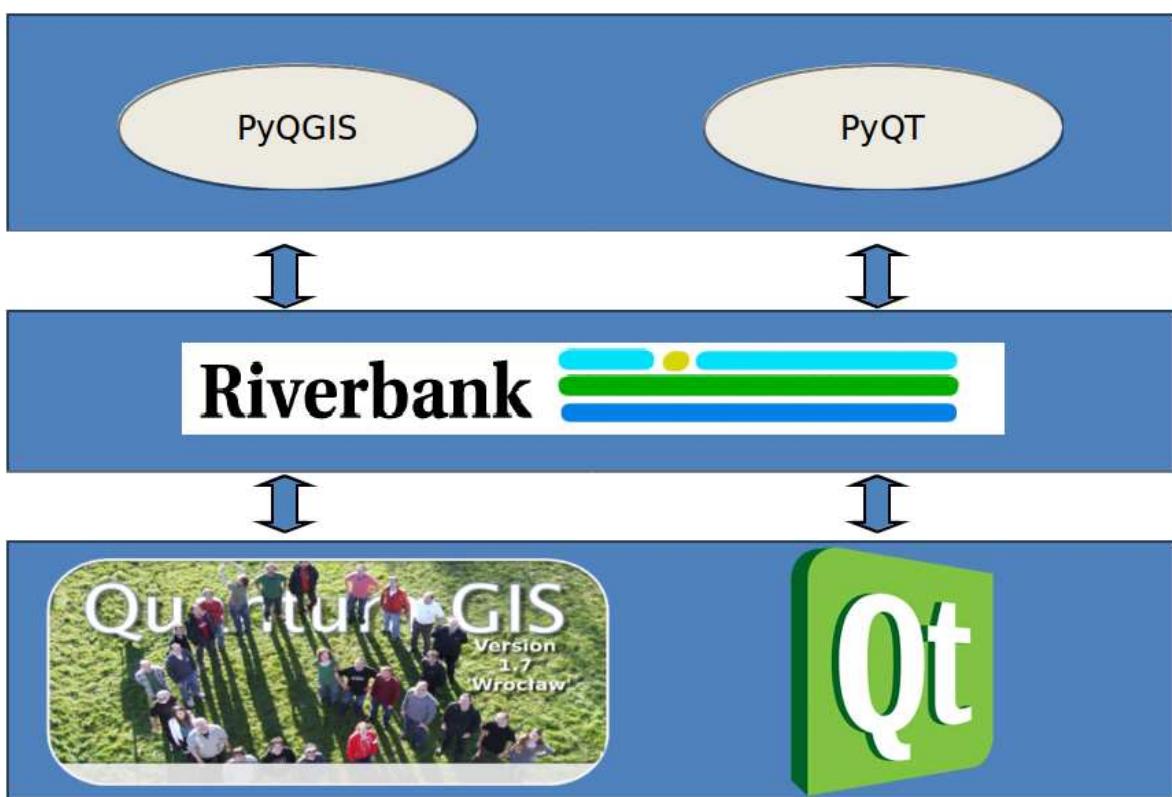
Tablica 4. Rezultati obrade rasterske datoteke Zagreb

U tablici su vidljivi formati datoteke *Zagreb*, njene veličine te brzine navigacije po njoj. Testiranja brzine napravljena su u istim uvjetima kako bi rezultati bili što objektivniji. Pritom je kao testno računalo korišteno ono s prethodno navedenim specifikacijama. Za navigaciju po datotekama korišten je *Quantum GIS*, konkretno korišteni su njegovi alati za približavanje, udaljavanje i pomicanje. Mjerenje brzine navigacije obavljen je pomoću ručne digitalne štoperice obzirom da nije bila potrebna visoka točnost rezultata. Sami rezultati pokazuju kako *jpg* datoteka ima najmanju veličinu ali i najmanju brzinu navigacije koja ovisno o alatu iznosi između 5 i 10 sekundi. *Tiff* datoteka je znatno veća,

ali ima veću brzinu navigacije. Za *ovr* datoteku nije navedena brzina navigacije jer se ta datoteka ne može koristiti samostalno nego jedino u kombinaciji s nekom drugom datotekom. Najbolji rezultati su postignuti zajedničkim korištenjem *tiff* i *ovr* datoteke prilikom čega je brzina navigacije iznosila između 1 i 2 sekunde. No, vidljivo je da kombinacija tih dviju datoteka zauzima najviše prostora na računalu. Obzirom da je za ovaj rad brzina navigacije po datoteci bitnija od zauzeća prostora, u dalnjem radu će se koristiti kombinacija *tiff* i *ovr* datoteka. Ovime je završena druga faza diplomskog rada, obrada podataka. Svi prikupljeni podaci (položajni i atributni podaci te rasterska podloga) bili su spremi za korištenje u *FuelGIS* aplikaciji te se moglo pristupiti sljedećoj fazi diplomskog rada, izradi *FuelGIS* aplikacije.

5. Izrada FuelGIS aplikacije

Izrada *FuelGIS* aplikacije odnosila se na pisanje koda aplikacije u *Python* programskom jeziku. Bilo je potrebno napraviti sučelje aplikacije koristeći *PyQt4* te implementirati funkcionalnosti iz *QGIS-a* koristeći *PyQGIS*. Slika 15. prikazuje povezanost *Pythona*, *Qt-a* i *Quantum GIS-a*. Na njoj je također naveden i pojam *Riverbank* koji se odnosi na tvrtku *Riverbank Computing*. Ona se bavi razvojem *PyQt-a* te je time bitno doprinijela njegovom korištenju u velikom broju aplikacija, među koje spada i *Quantum GIS*.



Slika 15. Prikaz povezanosti Pythona, Qt-a i QGIS-a (URL 32)

5.1. Instalacija OsGeo4W programskega paketa

Budući da su za programiranje aplikacije, koristeći *Python* programski jezik, bile potrebne *Qt* i *QGIS* biblioteke instaliran je *OsGeo4W* programski paket. Taj paket sadrži sve potrebne aplikacije i alate za izradu *FuelGIS* aplikacije: *Python* i pripadajuće module i biblioteke, *Quantum GIS* s pripadajućim bibliotekama i *PyQt4* alat s pripadajućim

modulima i bibliotekama. Drugi razlog instalacije tog paketa bila je nemogućnost povezivanja samostalnih instalacija *Pythona* i *Quantum GIS-a*. Konkretno, nije bilo moguće koristiti potrebne module, *QGIS core* i *QGIS gui*, koji su dio *Quantum GIS-a*. Pri pokušaju korištenja tih modula *Python* interpreter bi javljaо pogreške prilikom izvršavanja naredbi:

```
from qgis.core import *
from qgis.gui import *
```

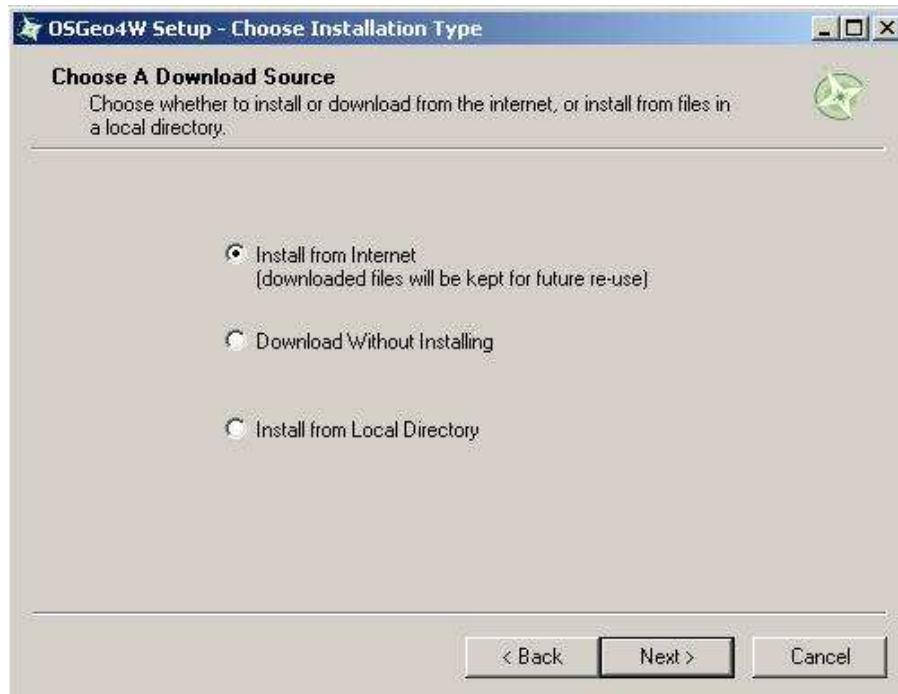
Te dvije naredbe omogućavaju korištenje svih elemenata iz modula *QGIS core* i *QGIS gui*. No, u ovom slučaju *Python* interpreter je javio pogreške: *ImportError: No module named qgis.core* i *ImportError: No module named qgis.gui*. Navedeno je značilo da *Python* ne može pronaći tražene module unatoč tome što su oni prisutni na računalu nakon instalacije *Quantum GIS-a*. Uzrok tome je nekompatibilnost *Windows* operativnih sustava s aplikacijama, poput *Quantum GIS-a*, koje nisu napravljene od strane *Microsofta*. Problem se pokušao riješiti definicijom dodatnih sistemskih i korisničkih varijabli unutar *Windows* operativnog sustava. No, bez uspjeha. Stoga se pristupilo instalaciji *OsGeo4W* programskog paketa.

Za početak je potrebno s Interneta preuzeti instalacijsku datoteku *OsGeo4W* programskog paketa (URL 8). Nakon preuzimanja iste potrebno ju je pokrenuti, nakon čega se pojavljuje početni izbornik (Slika 16.). Na njemu je moguće ovisno o željama odabrati između više različitih opsega instalacije. Manje iskusni korisnici imaju na raspolaganju opcije *Express Desktop Install* i *Express Web-GIS Install* s već definiranim parametrima, aplikacijama i opcijama za instalaciju. Na taj način se može napraviti instalacija programskog paketa *OSGeo4W* bez previše zamaranja s detaljima. U ovom slučaju je ipak odabrana opcija *Advanced Install* kako bi se samostalno mogli podesiti svi detalji vezani za instalaciju tog programskog paketa. Na taj način bilo je moguće odabrati potrebne aplikacije i alate te ih smjestiti na željenu lokaciju na računalu.



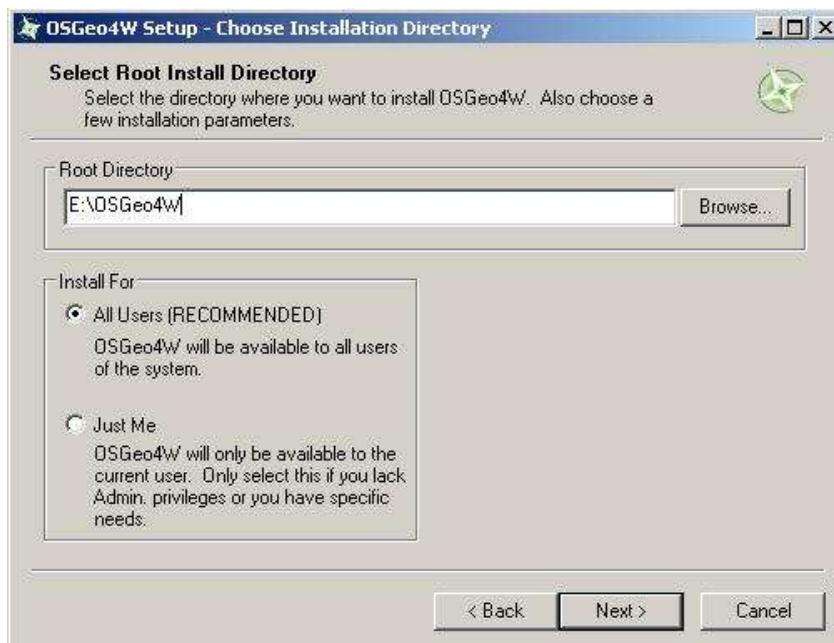
Slika 16. Početni izbornik OSGeo4W instalacije

Nakon izvedenog odabira isti se potvrdi klikom na tipku *Next* te se otvara sljedeći izbornik (Slika 17.).



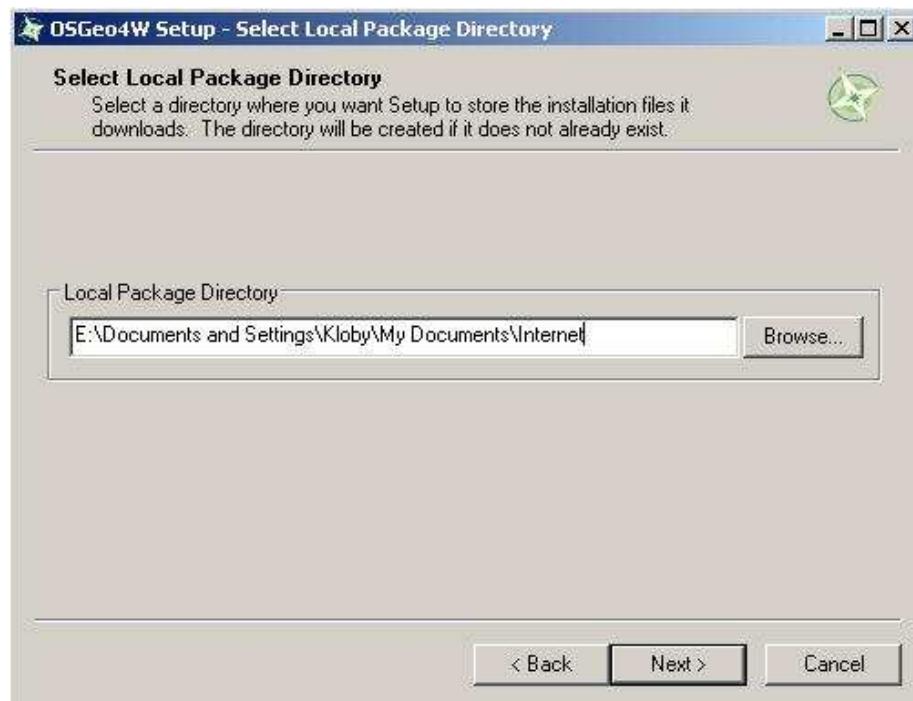
Slika 17. Izbornik s mogućnošću odabira načina instalacije

Na njemu je moguće odabrati način instalacije, hoće li se ona odvijati putem Interneta ili lokalno, s računala. Moguće je odabrati da se samo preuzmu potrebni podaci bez pokretanja instalacije. U ovom koraku odabранo je da se instalacija izvrši putem Interneta. Nakon izvedenog odabira isti se potvrdi klikom na tipku *Next* te se otvara idući izbornik (Slika 18.).



Slika 18. Izbornik s mogućnošću odabira mjesta i direktorija instalacije

Ovaj izbornik omogućuje odabir željenog mesta i direktorija gdje će biti instaliran programski paket *OSGeo4W*. Konkretno upisana je putanja E:\OSGeo4W koju je važno zapamtiti jer će biti potrebna kod dalnjih podešavanja. Također, na tom izborniku je moguće odabrati za koje će korisnike biti napravljena instalacija tog programskog paketa. Odluka ovisi o vlastitom odabiru korisnika, ali preporučuje se instalaciju provesti tako da *OSGeo4W* bude dostupan svim korisnicima računala. To je u ovom slučaju i odabran te se klikom na tipku *Next* otvara idući izbornik (Slika 19.). Ovaj izbornik omogućuje odabir željenog mesta i direktorija gdje će se pohraniti datoteke preuzete s Interneta koje su potrebne za instalaciju programskog paketa *OSGeo4W*. Navedena putanja nije važna tako da može poslužiti bilo koja lokacija na računalu.



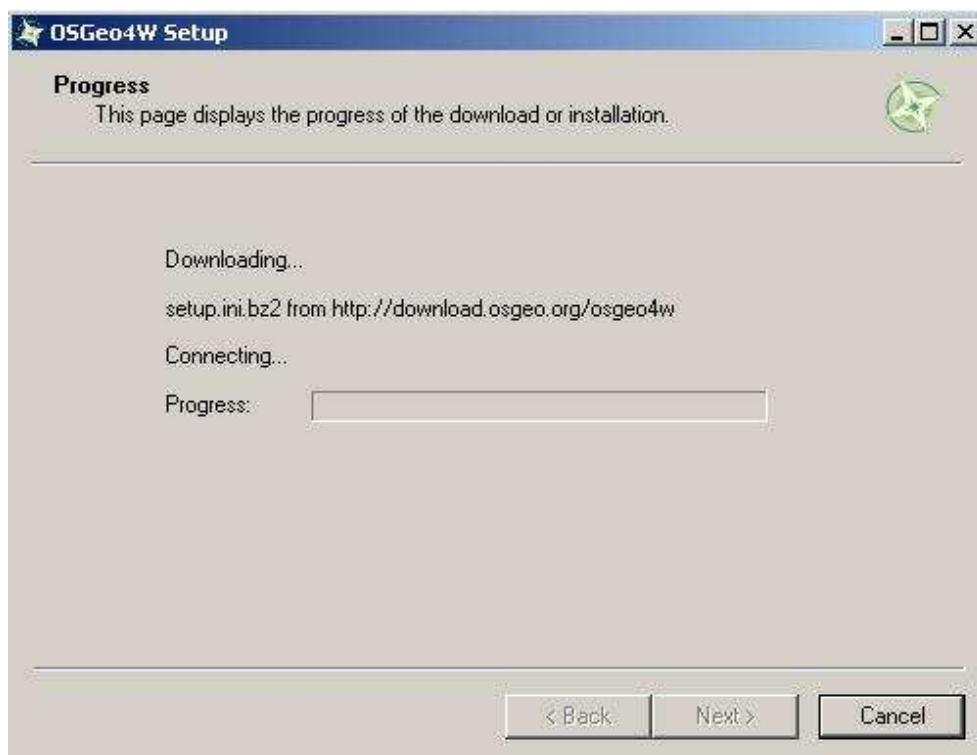
Slika 19. Izbornik s mogućnošću odabira mesta i direktorija za preuzimanje datoteka potrebnih pri instalaciji

Nakon unosa željene lokacije, klikom na tipku *Next* otvara se sljedeći izbornik (Slika 20.).



Slika 20. Izbornik za odabir vrste Internet konekcije

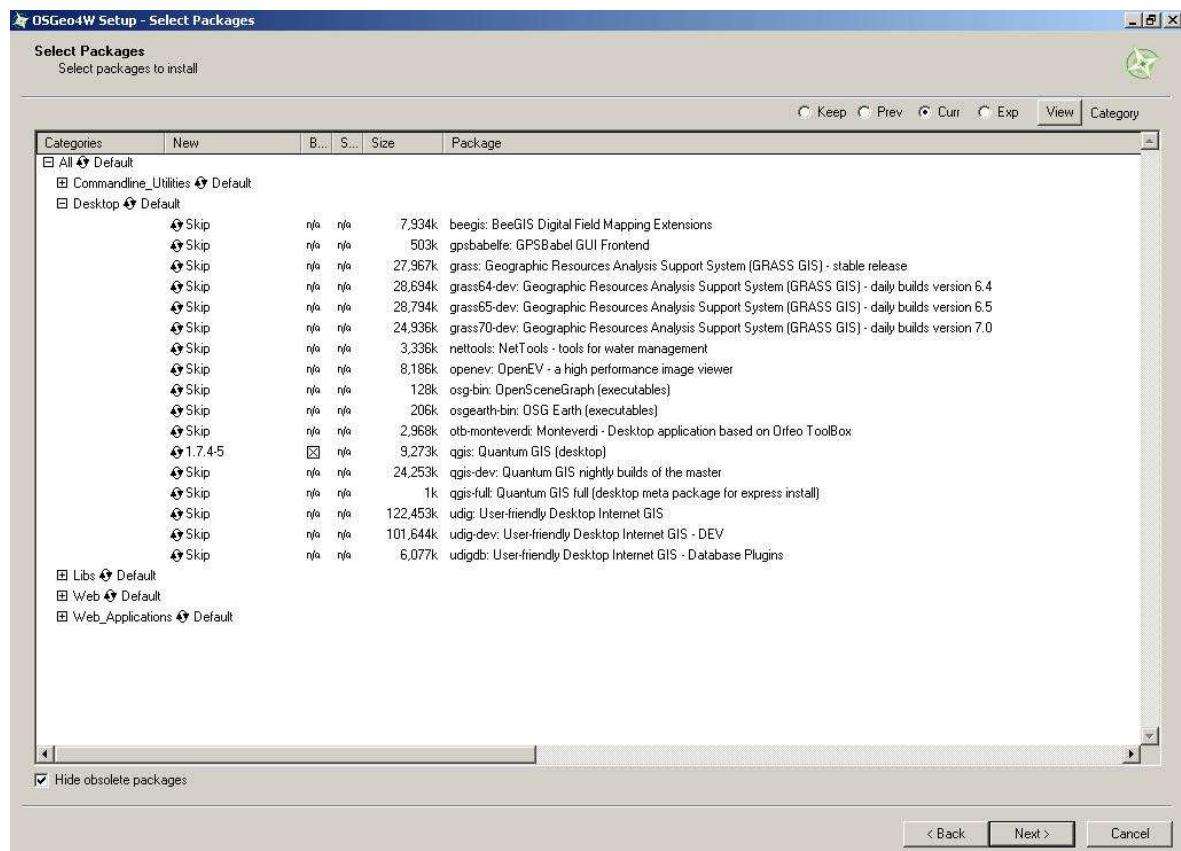
U njemu je potrebno odabrati vrstu konekcije koja se koristi za pristup Internetu. Velika većina korisnika ima računalo direktno spojeno na Internet, stoga se tim slučajevima odabire opcija *Direct Connection*. U suprotnom je potrebno provjeriti na koji način je računalo spojeno na Internet. Kako je korišteno računalo izravno spojeno na Internet, odabere se opcija *Direct Connection* te se klikom na tipku *Next* otvara informativni prozor (Slika 21.).



Slika 21. Informativni prozor – preuzimanje instalacijskih informacija

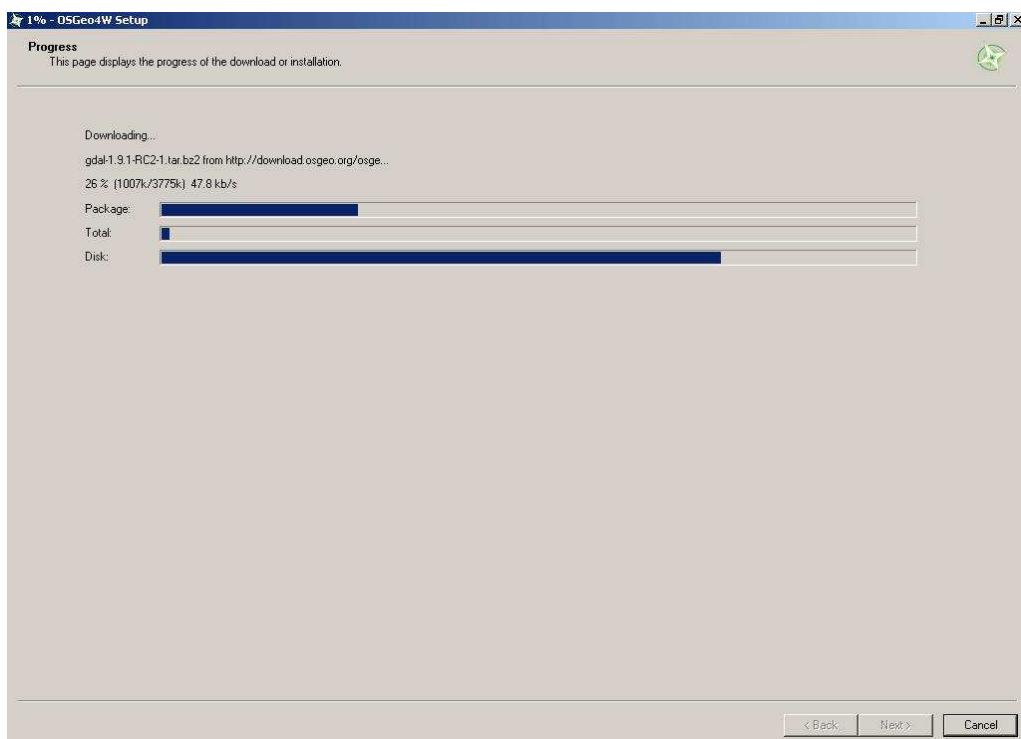
On prikazuje napredak, izražen u postotcima, vezan za preuzimanje potrebnih programskih paketa. Oni se preuzimaju putem Interneta s poslužitelja koji sadrže ažurne informacije o svim dostupnim aplikacijama, alatima i bibliotekama unutar *OSGeo4W* programske pakete. Nakon završetka preuzimanja tih programskih paketa otvara se sljedeći izbornik (Slika 22.). Taj izbornik sadrži popis svih aplikacija, alata i biblioteka dostupnih za instalaciju kroz programski paket *OSGeo4W*. Za instalaciju *Quantum GIS* aplikacije potrebno je odabrati kategoriju *Desktop* te unutar nje opciju *Quantum GIS (desktop)*. Prilikom tog odabira automatski se odabiru i svi nužni alati i biblioteke koji su potrebni za rad s *QGIS* aplikacijom, poput *Pythona*, *PyQT4-a*, *PROJ.4* biblioteka itd. Njih je moguće

ručno ukloniti s instalacijskog popisa, no to nije preporučljivo. Također, moguće je kasnije ponovnim pokretanjem instalacije naknadno instalirati željene komponente *OSGeo4W* programskog paketa.



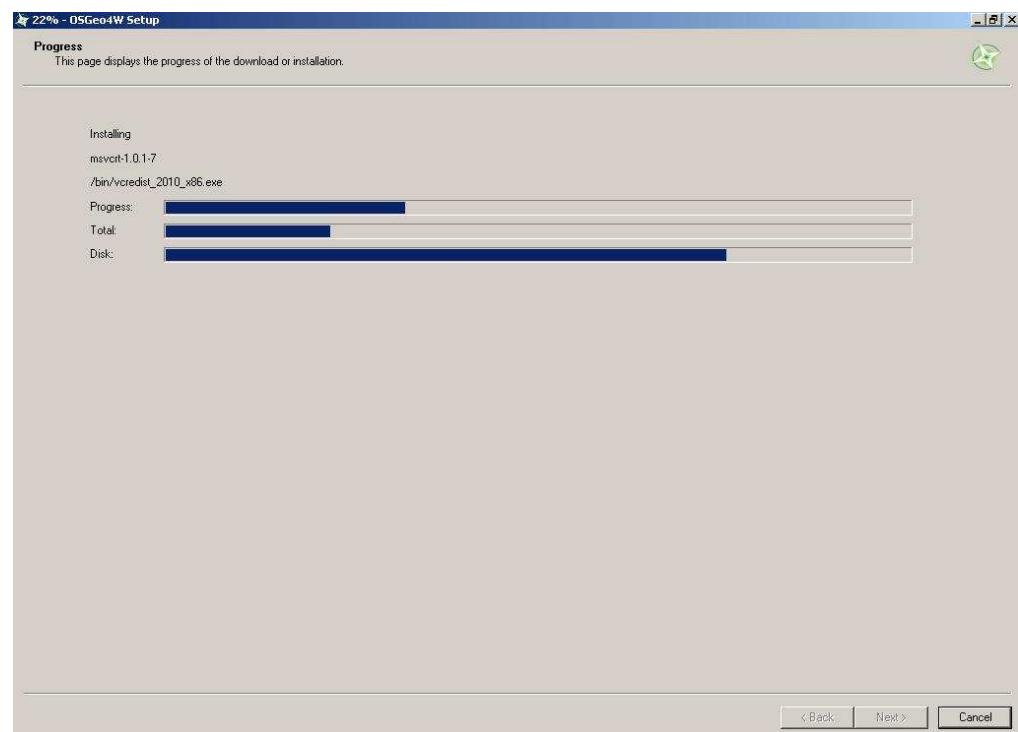
Slika 22. Izbornik s instalacijskim popisom

Nakon završenog odabira željenih komponenti klikom na tipku *Next* otvara se informativni prozor (Slika 23.). Na njemu je moguće pratiti napredak, izražen u postotcima, preuzimanja potrebnih datoteka za instalaciju odabranih komponenti. Vrijeme preuzimanja ovisi o brzini korištene Internet veze te odabranim komponentama tj. o njihovoj veličini, u ovom slučaju korištenjem ADSL pristupa Internetu potrebno je nekoliko minuta da se preuzmu potrebne datoteke.



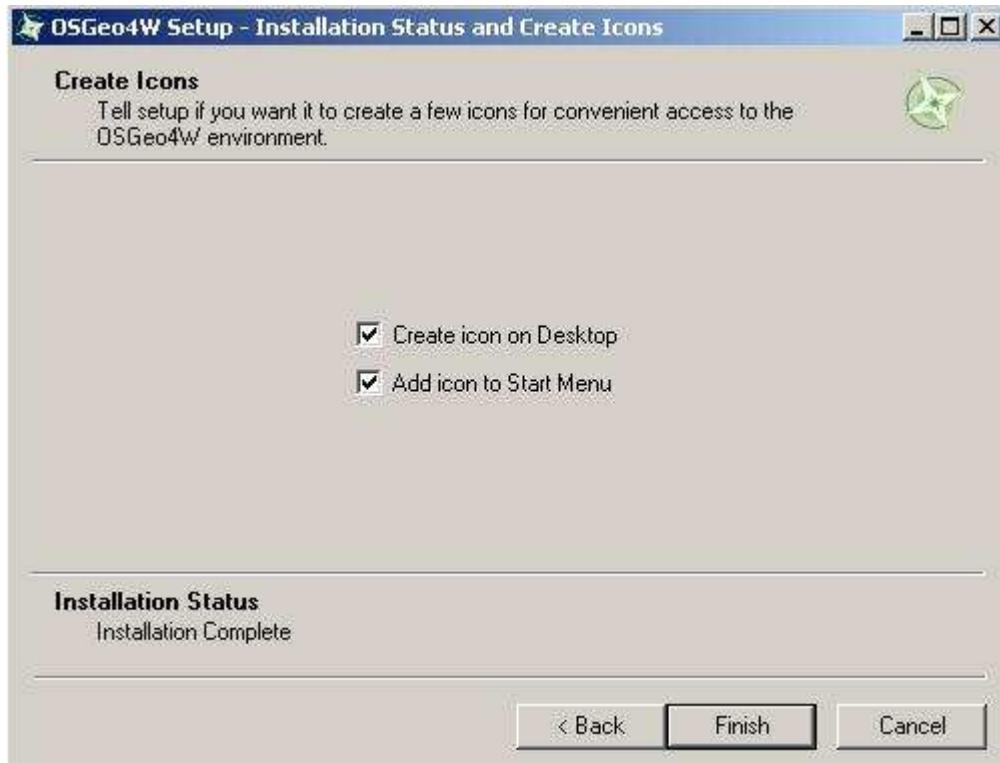
Slika 23. Informativni prozor – preuzimanje potrebnih datoteka

Završetkom preuzimanja potrebnih datoteka otvara se novi informativni prozor (Slika 24.).



Slika 24. Informativni prozor – instalacija odabranih komponenti

On prikazuje napredak, izražen u postotcima, instalacije odabranih komponenti. Nakon završetka instalacije prikazuje se završni izbornik (Slika 25.).



Slika 25. Završni izbornik OSGeo4W instalacije

Na njemu je moguće odabrati stvaranje prečaca (eng. *shortcut*) na radnoj površini (eng. *desktop*) te u *Windows Start* izborniku. Preporučuje se odabir ponuđenih opcija kako bi se olakšao pristup instaliranim komponentama *OSGeo4W* programskog paketa. Za potvrdu završetka instalacijskog postupka potrebno je odabrati tipku *Finish*. Time je završen postupak instalacije *OSGeo4W* programskog paketa.

5.2. Podešavanje OsGeo4W programskog paketa i ostale pripremne radnje

Kako bi komponente *OSGeo4W* programskog paketa mogle ispravno raditi u *Windows* okruženju potrebno je nakon njegove instalacije napraviti određena podešavanja u *Windows* operativnom sustavu. Potrebno je podesiti dodatne sistemske i korisničke varijable unutar *Windowsa* kako bi *Quantum GIS* i *Python* ispravno radili. Također, trebalo je riješiti i druge tehničke poteškoće koje su se pojavile prilikom pisanja, pokretanja i izvršavanja *Python* koda.

5.2.1. Podešavanje *OsGeo4W* programskog paketa

Budući da je u radu korišten *Windows XP* operativni sustav, ovdje će biti opisan postupak za podešavanje u tom okruženju. Za ostale *Windows* operativne sustave postupak se neće opisivati, no bitno je naglasiti da je on drugačiji ovisno o operativnom sustavu. Za početak podešavanja odabere se *Windows Start* izbornik te u njemu izbornik *Settings*. Unutar njega se odabere opcija *Control Panel*, te se od ponuđenog odabere opcija *System*, čime se otvara novi prozor s nazivom *System Properties*. Na njemu se odabere tabulator *Advanced* te opcija *Environment Variables* čime se otvara istoimeni prozor. Unutar njega potrebno je pronaći pod *System variables* varijablu pod imenom *Path* te odabrati opciju *Edit*. Time se omogućava uređivanje navedene varijable, na kraju reda pod *Variable value* dodaje se točka-zarez (;), a iza nje dodaje se lokacija *bin* direktorija od *Quantum GIS-a* (Slika 26.). Ovisno o instalacijskoj putanji dodaje se odgovarajuća putanja, u ovom slučaju dodana je E:\OSGeo4W\apps\qgis\bin. Odabirom tipke *OK* završava se uređivanje varijable *Path*.



Slika 26. Forma za uređivanje sistemskih varijabli

Drugi korak je stvaranje nove korisničke varijable kako bi *Python* ispravno radio. To se također obavlja u prozoru *Environment Variables*, pod *User variables* odabere se tipka *New* te se otvara forma za unos nove korisničke varijable (Slika 27.).



Slika 27. Forma za dodavanje korisničkih varijabli

Za *Variable name* se unosi naziv varijable, u ovom slučaju PYTHONPATH. Pod *Variable value* unosi se vrijednost *Python* varijable tj. njegova putanja, koja je ovisna o instalacijskoj putanji kao i u prethodnom slučaju. Unesena je putanja E:\OSGeo4W\apps\qgis\python te je odabirom tipke *OK* završeno stvaranje nove korisničke varijable. Zatim je potrebno odabirom tipke *OK* zatvoriti prozore *Environment Variables* i *System Properties* te napraviti restart računala kako bi napravljene promjene imale učinka. Ovime se završava postupak podešavanja dodatnih sistemskih i korisničkih varijabli unutar *Windows XP* operativnog sustava. Rezultat instalacije i podešavanja komponenti *OSGeo4W* programskog paketa trebalo bi biti računalo podešeno za programiranje s *Pythonom* koristeći potrebne module iz *PyQt4-a* i *QGIS-a*.

5.2.2. Provjera podešavanja *OsGeo4W* programskog paketa

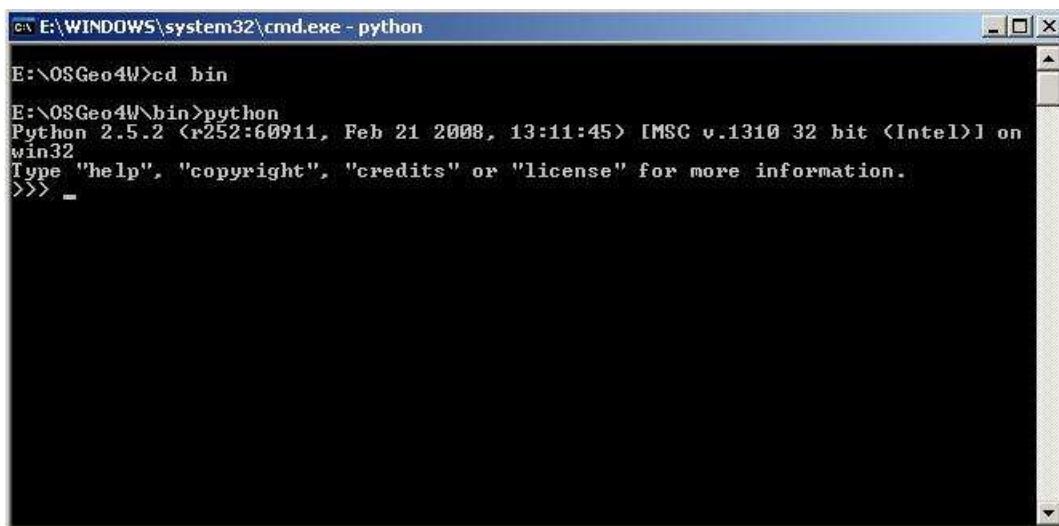
Provjera ispravnosti podešavanja obavlja se pokretanjem *bat* datoteke s nazivom *OSGeo4W* koja se nalazi u instalacijskom direktoriju istoimenog programskog paketa (E:\OSGeo4W). Ta *bat* datoteka otvara *cmd* (eng. *Command Prompt*) prozor (Slika 28.) koji služi za ručno unošenje i izvršavanje pojedinačnih naredbi.



Slika 28. *Command Prompt* prozor

On služi za pokretanje *Pythona* te ostalih komponenti *OSGeo4W* programskog paketa, pritom je potrebno upisati odgovarajuće naredbe. Za pokretanje *Pythona* se prvo upiše naredba *cd bin* te se za njeno izvršavanje pritisne tipka *Enter*. Na taj način pristupa se

direktoriju *bin* u kojem se nalaze datoteke potrebne za pokretanje *Pythona*. Zatim se upiše naredba *python* te se ponovno pritisne tipka *Enter*. Na taj način je izvršeno pokretanje *Pythona* te se u *cmd* prozoru prikazuje *Python* interpreter (Slika 29.) koji se može prepoznati po tri znaka *>>>*.



Slika 29. Python interpreter u cmd prozoru

U njemu je moguće provjeriti ispravnost prethodno podešenih sistemskih i korisničkih varijabli pomoću naredbi:

```
from qgis.core import *
from qgis.gui import *
from PyQt4.QtGui import *
from PyQt4.QtCore import *
```

Te naredbe služe za pozivanje svih elemenata iz modula *QGIS core*, *QGIS gui*, *PyQt4 Gui* i *PyQt4 Core*. Navedene naredbe se unose svaka pojedinačno, nakon unosa pritiskom tipke *Enter* *Python* interpreter će je izvršiti. Ukoliko je sve ispravno podešeno neće se javiti nikakva poruka o pogrešci. U suprotnom je potrebno provjeriti ispravnost stvorenih sistemskih i korisničkih varijabli te jesu li instalirane sve potrebne komponente *OSGeo4W* programskog paketa. Nakon završetka provjere *cmd* prozor je moguće zatvoriti pritiskom na tipku *Close (X)*.

5.2.3. Izrada *bat* datoteke

Pokretanje *Python* interpretera putem *cmd* prozora je složena radnja stoga je odlučeno da se interpreter pokreće na drugi način. Također, postojao je i problem s izvršavanjem napisanog koda *FuelGIS* aplikacije, u obliku *py* datoteke, obzirom da *cmd* zbog svojih ograničenja nije udovoljavao potrebama. Kod aplikacije ne izvršava se sekvencijalno nego uvjetno. To predstavlja problem jer se naredbe pomoću *cmd* prozora unose ručno te izvršavaju u *Python* interpretalu sekvencijalno jedna po jedna. Uz to ne postoji mogućnost uređivanja *Python* skripti putem *cmd* prozora. Prethodno je korišten *IDLE (Python GUI)* koji omogućava interakciju s *Python* interpretatom. *IDLE* (eng. *integrated development environment*) je zapravo integrirano razvojno okruženje za *Python* koji dolazi zajedno s tim alatom za programiranje još od verzije 0.1 (URL 33). *IDLE* omogućuje uređivanje datoteka, tzv. *Python* skripti, koje se mogu jednostavno pokretati direktno iz sučelja. Za razliku od samostalne verzije *Python* iz programskog paketa *OSGeo4W* nema te mogućnosti. Kako bi se riješili problemi s pokretanjem *Python* interpretera i *Python* skripti (*py* datoteka) korištena je *bat* datoteka. *Bat* datoteke su obične tekstualne datoteke koje sadrže naredbe koje je potrebno izvršiti jednu poslije druge (URL 34). Naredbe upisane u *bat* datoteku se izvršavaju kroz *Windows Command Prompt (cmd)*. Kako bi napravili *bat* datoteku potrebno je otvoriti program za uređivanje teksta (npr. *Notepad*) i upisati naredbe koje se trebaju izvršiti. Nakon upisa naredbi datoteka se sprema s ekstenzijom *bat*. Spremljena datoteka se pokreće dvostrukim klikom te se time automatski pokreće izvršavanje naredbi napisanih u njoj. Pokretanje *Python* interpretera i izvršavanje koda riješeno je preuzimanjem i prilagođavanjem sadržaja tj. naredbi *OSGeo4W.bat* datoteke. Na taj način napravljena je nova datoteka, s nazivom *FuelGIS.bat*, koja je postavljala potrebnu okolinu za izvršavanje željenih radnji. Konkretno, *OSGeo4W.bat* datoteka je sadržavala sljedeće naredbe:

```
@echo off
set OSGEO4W_ROOT=E:\OSGeo4W
PATH=%OSGEO4W_ROOT%\bin;%PATH%
for %%f in (%OSGEO4W_ROOT%\etc\ini\*.bat) do call %%f
@echo on
@cmd.exe
```

Za potrebe *FuelGIS* aplikacije dodane su dvije naredbe koje definiraju varijable okoline potrebne za pokretanje aplikacije. Te varijable su potrebne kako bi se postavila okolina

potrebna za ispravan rad *QGIS* funkcionalnosti. Također, na kraju *bat* datoteke naređeno je izvršavanje napisanog koda iz *FuelGIS.py* datoteke, pritom *Python* interpreter ostaje upaljen. To je riješeno unosom dodatnog parametra (-i) koji ostavlja interpreter aktivnim nakon izvršavanja programa, neovisno o tome je li se program uspješno ili neuspješno izvršio. Na taj je način moguće dobiti od *Python* interpretera povratnu informaciju o pogrešci. Konačan sadržaj novostvorene *bat* datoteke s nazivom *FuelGIS* je:

```
@echo off
set OSGEO4W_ROOT=E:\OSGeo4W
PATH=%OSGEO4W_ROOT%\bin;%PATH%
for %%f in (%OSGEO4W_ROOT%\etc\ini\*.bat) do call %%f

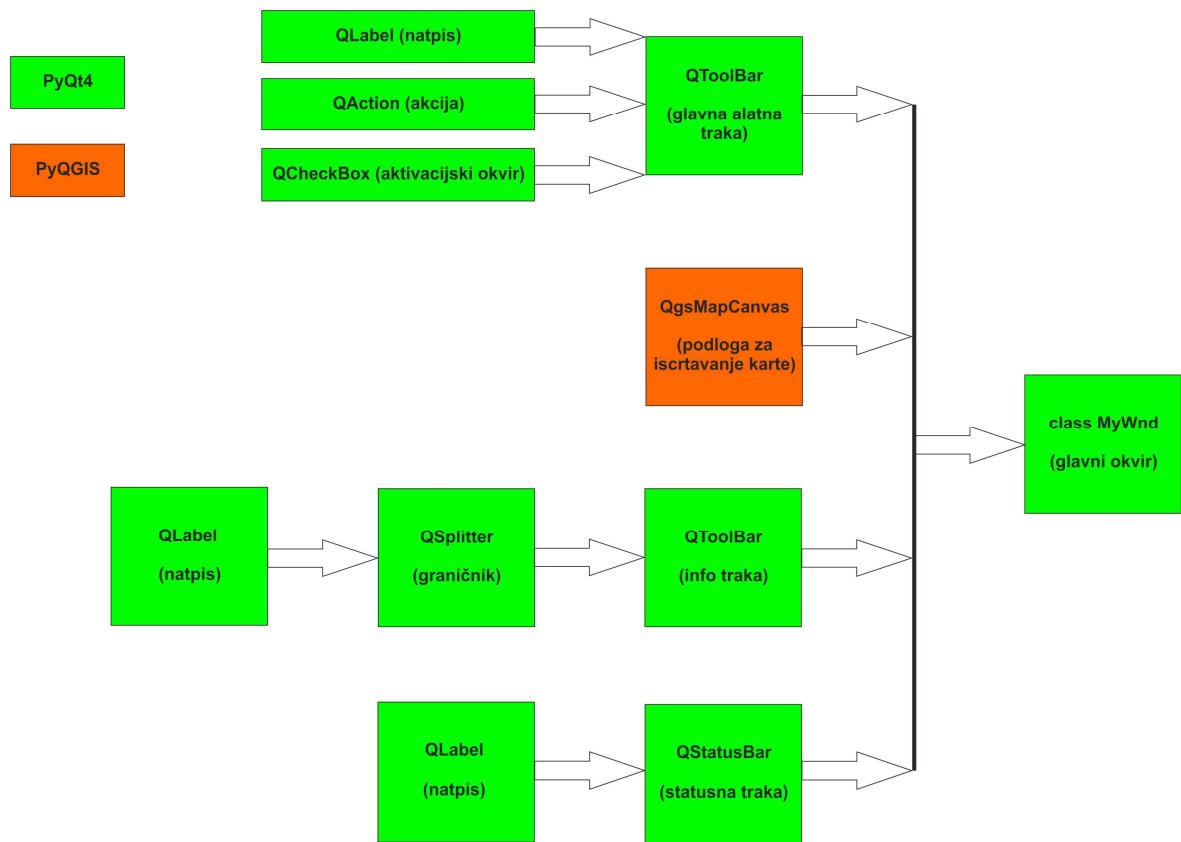
rem dodane varijable okoline potrebne za pokretanje aplikacije
set QGISHOME=%OSGEO4W_ROOT%\apps\qgis
PATH=%OSGEO4W_ROOT%\apps\qgis\bin;%PATH%
@echo on
@python -i FuelGIS.py
```

Treba naglasiti da naziv *bat* datoteke može biti proizvoljan, ne mora biti jednak nazivu *py* datoteke, ali obje datoteke se moraju nalaziti u istom direktoriju kako bi se pokretanjem *bat* datoteke izvršio kod iz *py* datoteke. Time su riješeni svi tehnički problemi vezani uz pokretanje i izvođenje napisanog koda *FuelGIS* aplikacije te dobivanje povratnih informacija iz *Python* interpretera. Na taj način ostvareni su preduvjeti za daljnji rad na *FuelGIS* aplikaciji, tj. bilo je moguće započeti pisanje koda.

5.3. Izrada sučelja *FuelGIS* aplikacije

Izrada sučelja *FuelGIS* aplikacije temeljena je na korištenju alata *PyQt4*, tj. njegovih modula *QtCore*, *QtGui* koji su sadržavali potrebne klase, metode i funkcije za stvaranje sučelja. Također, uz njih su korišteni i drugi moduli sa svojim pripadajućim klasama i metodama. Na dijagramu (Slika 30.) prikazani su svi njegovi elementi. Na njemu je vidljivo da su za izradu uz *PyQt4* korišteni i *PyQGIS* elementi. Elementi *QLabel*, *QAction*, *QCheckBox*, *QSplitter*, *QToolBar* i *QStatusBar* pripadaju *PyQt4-u*, dok *QgsMapCanvas* pripada *PyQGIS-u*. Klasa *MyWnd* predstavlja glavni okvir te zajedno sa svim nabrojanim elementima čini sučelje *FuelGIS* aplikacije. Sučelje je napravljeno ručno, pisanjem koda pomoću kojeg su definirani svi elementi sučelja: glavni okvir aplikacije, podloga za iscrtavanje karte (eng. *map canvas*), statusna traka, glavna alatna traka, aktivacijski okviri (eng. *checkbox*), gumbi, polja za prikaz informacija, info traka itd.

Sučelje je bilo moguće izraditi pomoću programa *Qt Designer*, koji omogućava brzu i jednostavnu izradu na principu povlačenja i ispuštanja (eng. *drag-and-drop*) elemenata. Unatoč tome odlučeno je da se sučelje napravi ručno kako bi se dobio bolji uvid u funkcioniranje pojedinih elemenata. U nastavku će biti detaljno opisan način izrade svih prethodno navedenih elemenata sučelja.



Slika 30. Elementi sučelja FuelGIS aplikacije

5.3.1. Početni parametri *PyQt4* aplikacije

Kako bi započeli izradu aplikacije potrebno je na samom početku učitati potrebne module te njihove komponente (klase, funkcije i metode) koje će kasnije biti korištene. Sljedeći isječak koda prikazuje naredbe pomoću kojih se određuju početni parametri *PyQt4* aplikacije.

```
# -*- coding: utf-8 -*-
#unos potrebnih modula
import sys
```

```
from PyQt4.QtGui import QAction, QMainWindow, QApplication, QLabel, QFrame, QStatusBar,
QCheckBox, QToolBar, QSplitter, QIcon, QMessageBox, QColor
from PyQt4.QtCore import SIGNAL, Qt, QString, QObject, QVariant
```

Na samom početku koda (Prilog 1, linija 1) postavljen je *UTF8* format za kodiranje teksta, navedeni se koristi za prikaz latiničnog teksta tj. kako bi se u aplikaciji mogli ispravno prikazati hrvatski dijakritici, npr: ž,ć,đ itd. Zatim su učitani potrebni moduli s pripadajućim komponentama. Učitan je modul *sys* (Prilog 1, linija 4) koji sadržava argumente potrebne za rad *PyQt4* aplikacije. Zatim su učitani moduli *QtGui* i *QtCore* (Prilog 1, linija 8-9), njihova uloga je prethodno opisana u potpotpoglavlju 2.8.1. Treba napomenuti da su se tokom pisanja koda te naredbe nadopunjavale potrebnim komponentama. Na taj način se koriste potrebne komponente umjesto nepotrebnog učitavanja cijelog sadržaja navedenih modula. Tako se racionalno koristi memorijski prostor prilikom pokretanja aplikacije.

5.3.2. Osnovne funkcionalnosti *PyQt4* aplikacije

U osnovne funkcionalnosti *PyQt4* aplikacije spada: povezivanje s pripadajućom okolinom, pokretanje aplikacije te zatvaranje tj. gašenje aplikacije. Za izvršavanje navedenih funkcionalnosti koristi se *sys* modul tj. njegovi argumenti koji omogućavaju pokretanje okoline i aplikacije te na kraju prema korisnikovoj želji i zatvaranje aplikacije. Sljedeći isječak koda prikazuje naredbe koje određuju te osnovne funkcionalnosti svake *PyQt4* aplikacije.

```
#povezivanje aplikacije s PyQt4-om
app = QApplication(sys.argv)

#pokreni aplikaciju
retval = app.exec_()

#zatvorи aplikaciju
sys.exit(retval)
```

Svaka *PyQt4* aplikacija je objekt, odnosno instanca klase *QApplication* koja pomoću funkcija i metoda iz modula *sys* povezuje stvorenu aplikaciju s *PyQt4* okolinom. U ovom slučaju to je napravljeno pomoću vrijednosti varijable *app* (Prilog 1, linija 758). Pokretanje aplikacije u takvoj okolini obavlja se pomoću varijable *retval* (Prilog 1, linija 769) a zatvaranje, kada korisnik to zatraži, predajom te iste varijable kao argumenta funkciji *sys.exit* (Prilog 1, linija 775). Vidljivo je da se u svim tim procesima kao osnova koriste

argumenti i pojedine funkcije iz modula sys. Pritom ovisno o stanju navedenih varijabli uzrokovanim korisnikovom interakcijom dolazi do pokretanja ili zatvaranja aplikacije.

5.3.3. Glavni okvir aplikacije

Temelj svakog sučelja je glavni okvir aplikacije, u njemu se smještaju i prikazuju svi ostali elementi sučelja poput alatnih traka, okvira za izbor itd. Sljedeći isječak koda prikazuje potrebne naredbe za izradu glavnog okvira te njegovo prikazivanje.

```
#definiranje glavnog prozora
class MyWnd(QMainWindow):

    def __init__(self):
        QMainWindow.__init__(self)

        #postavi naziv glavnog prozora
        self.setWindowTitle("FuelGIS - aplikacija za pregled benzinskih postaja na
        području Grada Zagreba, razvijena na QGIS platformi")

        #postavi ikonu glavnog prozora
        self.setWindowIcon(QIcon("E:/FuelGIS/bp.png"))

    #prikaz glavnog prozora
    w = MyWnd()
    w.show()
```

Prvo se stvori nova klasa s nazivom *MyWnd* koja sadrži sve funkcionalnosti klase *QMainWindow*. Te funkcionalnosti imaju unaprijed zadane postavke koje omogućuju osnovne funkcionalnosti poput smanjenja glavnog okvira itd. Zatim se pomoću metode *__init__* definira konstruktor klase *MyWnd* koji služi za inicijalizaciju tog objekta kada on bude stvoren kasnije u kodu. Unutar te metode postavljaju se svi atributi vezani uz glavni okvir, poput njegovog naziva te ikone kojom će biti prikazan što je vidljivo u Prilogu 1, linija 49-59. Treba napomenuti kako se skoro sav kod *FuelGIS* aplikacije nalazi unutar klase *MyWnd* jer su ostali elementi sučelja i funkcionalnosti direktno povezani s glavnim okvirom. Nakon izrade potrebno je i prikazati glavni okvir. Prvo varijabla *w* spremi rezultat instanciranja klase *MyWnd* odnosno njenog konstruktora. Zatim se pozivom *show* metode *QMainWindow* klase taj okvir i prikazuje (Prilog 1, linija 764-766).

5.3.4. Podloga za iscrtavanje karte

Podloga za iscrtavanje karte služi za prikaz rasterskih i vektorskih slojeva podataka te interakciju s njima pomoću *QGIS* funkcionalnosti. Podloga za iscrtavanje karte ne pripada *PyQt4-u*, nego *PyQGIS-u*. Obzirom na važnost, njeno stvaranje i povezivanje sa

sučeljem bit će opisano u ovom dijelu rada. U idućem isječku koda (Prilog 1, linija 61-67) prikazane su sve potrebne naredbe za to.

```
#definiranje prostora za prikaz slojeva tj. map canvasa
self.canvas = QgsMapCanvas()
self.canvas.setCanvasColor(Qt.white)
self.canvas.show()

#postavi canvas kao centralni dio prozora
self.setCentralWidget(self.canvas)
```

Prvo se definira podloga za iscrtavanje karte instanciranjem klase *QgsMapCanvas*. Nakon toga odabrana je bijela kao boja podloge te je napravljen njen prikaz pomoću naredbe *show*. Osim toga potrebno je smjestiti podlogu unutar glavnog okvira na odgovarajuće mjesto. Obzirom na njenu važnost postavljena je kao centralni, glavni grafički element (eng. *widget*) unutar glavnog okvira.

5.3.5. Statusna traka

Statusna traka je horizontalna traka na dnu glavnog okvira koja prikazuje informacije o trenutnom stanju programa poput statusa zadatka koji se trenutno izvršava, informacije o odabranoj opciji itd. U ovom slučaju statusna traka služi za prikaz koordinata ovisno o položaju miša na podlozi za iscrtavanje karte. Idući isječak koda (Prilog 1, linija 83-87) prikazuje naredbe potrebne za stvaranje statusne trake, njeno smještanje u glavni okvir te povezivanje s prikazom koordinata.

```
#stvaranje statusne trake
sb = QStatusBar()
sb.setFixedHeight(22)
self.setStatusBar(sb)
self.mapcoords = MapCoords(self)
```

Za početak potrebno je definirati statusnu traku, to se napravi instanciranjem klase *QStatusBar*. Zatim se odredi njena visina te se poveže s glavnim okvirom tj. smješta se na njegovo dno naredbom *setStatusBar*. Nakon toga vrši se instanciranje klase *MapCoords* koja služi za prikazivanje koordinata.

5.3.6. Prikaz koordinata u statusnoj traci

Kako bi statusna traka mogla prikazivati koordinate stvorena je posebna klasa tj. objekt s nazivom *MapCoords* sa svim potrebnim parametrima. Sljedeći isječak koda

(Prilog 1, linija 15-47) prikazuje potrebne naredbe za izradu te klase. Treba napomenuti da ta klasa treba biti stvorena na samom početku prije klase glavnog okvira (*MyWnd*) kako bi se mogla kasnije instancirati unutar nje kako je navedeno u prethodnom potpotpoglavlju.

```
#definiranje prikaza koordinata na karti
class MapCoords(object):

    def __init__(self, mainwindow):
        selfmainwindow = mainwindow

        #povezivanje canvasa sa signalom pomaka miša kako bi se odredile koordinate
        QObject.connect(mainwindow.canvas, SIGNAL("xyCoordinates(const QgsPoint&)"),
        self.updateCoordsDisplay)

        #definiranje polja za prikaz koordinata
        self.xy = QLabel("0.0 , 0.0")
        self.xy.setFixedWidth(200)
        self.xy.setAlignment(Qt.AlignHCenter)
        self.xy.setFrameStyle(QFrame.StyledPanel)

        #definiranje oznake polja za prikaz koordinata
        self.name = QLabel("Koordinate:")
        self.name.setFrameStyle(QFrame.StyledPanel | QFrame.Raised)

        #dodaj oznaku u statusnu traku
        selfmainwindow.statusBar().addPermanentWidget(self.name)

        #dodaj polje za prikaz u statusnu traku
        selfmainwindow.statusBar().addPermanentWidget(self.xy)

    #metoda za ažuriranje prikaza koordinata
    def updateCoordsDisplay(self, p):

        #koordinate miša prikaži kao string
        capture_string = QString(str(p.x()) + " , " +str(p.y()))

        #koordinate stavi u polje za prikaz
        self.xy.setText(capture_string)
```

Definira se nova klasa s nazivom *MapCoords*, nakon toga konstruktor u obliku metode *__init__* definira kako će se ta klasa instancirati u dalnjem kodu aplikacije. Pritom je potreban argument s nazivom *mainwindow* koji predstavlja klasu *MyWnd* tj. glavni okvir. Argument *mainwindow* koristi se kao argument konstruktora unutar klase *MapCoords*. Sljedeći korak je povezivanje podloge za isertavanje karte pomoću signala, koji očitava x i y koordinate miša, s metodom za prikaz tih koordinata naziva *updateCoordsDisplay*. Nakon toga se instanciranjem klase *QLabel* definira polje za prikaz koordinata s početnim vrijednostima 0. Odredi se dužina tog polja te način poravnjanja znakova u njemu, horizontalno u sredini polja pomoću parametra *setAlignment(Qt.AlignHCenter)*. Uz to zadan je i stil prikaza okvira tog polja s nazivom *StyledPanel*. Zatim je napravljena oznaka *name* s pripadajućim tekstrom koji opisuje sadržaj polja za koordinate. Njoj je također dodijeljen stil prikaza okvira kako bi bila adekvatno prikazana. Ta oznaka i prethodno stvoreno polje smješteni su u statusnu traku glavnog prozora kao stalni elementi grafičkog

sučelja pomoću naredbe *addWidget*. Na kraju je definirana metoda *updateCoordsDisplay* koja služi za prikaz koordinata. Njoj se kao argument predaje točka *p* smještena unutar podloge za iscrtavanje karte na kojoj se nalazi pokazivač miša. Vrijednosti varijable *capture_string* pretvaraju se iz brojeva u niz znakova za prikaz međusobno odvojenih zarezom. Nakon konverzije poziva se metoda koja postavlja trenutni prikaz koordinata na vrijednost *capture_string* varijable u polje za prikaz koordinata smješteno na statusnoj traci.

5.3.7. Gumbi za približavanje, udaljavanje i pomicanje

Ne postoji sučelje koje ne sadrži barem jedan gumb jer se on može koristiti u različite svrhe. Tako i *FuelGIS* aplikacija sadrži njih nekoliko čije se funkcije međusobno razlikuju, pritom je moguće koristiti samo jednu od njih. Tako postoje kontrole/funkcionalnosti za približavanje, udaljavanje i pomicanje povezane s elementom *QPushButton* čijim se odabirom aktivira željeni alat za interakciju s prikazanim slojevima. U sljedećem isječku koda bit će prikazana izrada jednog takvog gumba, konkretno gumba za približavanje. Treba napomenuti da su prilikom njegove izrade uz *PyQt4* korišteni i elementi *PyQGIS-a*. Gumbi za udaljavanje i pomicanje su napravljeni na identičan način stoga njihova izrada neće biti opisana.

```
#stvaranje akcije s pripadajućim nazivom
self.actionZoomIn = QAction(QString(u"Približi"), self)

#omogućavanje odabira željene akcije
self.actionZoomIn.setCheckable(True)

#povezivanje akcije s potrebnim signalom kako bi se aktivirala pripadajuća metoda
self.connect(self.actionZoomIn, SIGNAL("triggered()"), self.zoomIn)

#stvor QGIS alat za približavanje i poveži s pripadajućom PyQT akcijom
self.toolZoomIn = QgsMapToolZoom(self.canvas, False) # false = in
self.toolZoomIn.setAction(self.actionZoomIn)

def zoomIn(self):

    #uključi QGIS alat približavanje
    self.canvas.setMapTool(self.toolZoomIn)

    #isključi gumb "Odaber"
    self.actionSelect.setChecked(False)

    #isključi QGIS alat za odabir
    self.canvas.unsetMapTool(self.clickTool)
```

Za početak stvor se akcija s nazivom *Približi* (Prilog 1, linija 90) te se omogući odabir te akcije (Prilog 1, linija 96). Nakon toga tu akciju potrebno je povezati pomoću signala za

aktivaciju s metodom *zoomIn* (Prilog 1, linija 102). Zatim se pomoću klase *QgsMapToolZoom* stvori *QGIS* alat za približavanje te se povezuje s pripadajućom *PyQt4* akcijom (Prilog 1, linija 108-109). Na kraju se definira metoda *zoomIn* kojom se izvršava približavanje (Prilog 1, linija 446-455). Njenim izvršavanjem uključuje se *QGIS* alat za približavanje te se isključuje gumb *Odaberi* zajedno s pripadajućim alatom za odabir. Ove dvije zadnje naredbe su posljedica unaprijed definiranih *QGIS* alata za približavanje, udaljavanje i pomicanje pomoću odgovarajućih *PyQGIS* klasa dok se alat za odabir morao ručno napraviti. Navedeno znači da se prilikom aktivacije alata za približavanje deaktiviraju alati za udaljavanje i pomicanje dok to nije slučaj s alatom za odabir. Stoga je potrebno ručno izvršiti njegovo isključivanje.

5.3.8. Gumb za odabir

Zbog svoje važnosti ali i posebnosti izrade gumb za odabir zaslužuje posebno mjesto u ovom radu. On omogućava aktivaciju posebnog alata za odabir kojim je moguće odabrati željenu benzinsku postaju te dobiti detaljne informacije o njoj. Treba napomenuti da se prilikom aktivacije tog alata također aktivira i info alatna traka te se odabirom željene benzinske postaje u njoj prikazuju atributne informacije. Zbog kompleksnosti i veličine koda ovdje će biti opisan početni dio vezan uz izradu gumba za odabir te njegovo povezivanje s odgovarajućim alatom. Kao i u prethodnom slučaju prilikom njegove izrade uz *PyQt4* korišteni su i elementi *PyQGIS-a*. Kod potreban za njegovu izradu prikazan je u sljedećem isječku.

```

self.actionSelect = QAction(QString("Odaberi"), self)
self.actionSelect.setCheckable(True)
self.connect(self.actionSelect, SIGNAL("triggered()"), self.selectActivated)

#stvori QGIS alat koji šalje koordinate prilikom svakog klika na canvasu
self.clickTool = QgsMapToolEmitPoint(self.canvas)

#povezivanje alata za klikanje pomoću potrebnog signala s metodom za odabir
#pojedine benzinske postaje
QObject.connect(self.clickTool, SIGNAL("canvasClicked(const QgsPoint &,
Qt::MouseButton)"), self.selectFeature)

#metoda za aktivaciju QGIS alata "Odaberi"
def selectActivated(self):

    #otkrivanje pošiljatelja signala
    button_odaberi = self.sender()

    #ako je uključen gumb "Odaberi"
    if button_odaberi.isChecked():

        #isključi ostale QGIS alate
        self.canvas.unsetMapTool(self.toolZoomIn)
        self.canvas.unsetMapTool(self.toolZoomOut)

```

```

    self.canvas.unsetMapTool(self.toolPan)
    #uključi QGIS alat za odabir
    self.canvas.setMapTool(self.clickTool)

    #prikaži info traku
    self.infoToolbar.setVisible(True)

    #ponovno iscrtavanje canvasa kako bi se prikazala info traka
    self.canvas.refresh()

```

Prvo je stvorena akciju s nazivom *Odaberi*, (Prilog 1, linija 93). Nakon toga se omogući odabir pomoću parametra *setCheckable* (Prilog 1, linija 99) te poveže metoda akcije pomoću signala za aktivaciju s pripadajućom metodom (Prilog 1, linija 105) naziva *selectActivated*. Kako bi napravili alat koji će omogućiti da se klikom na kartu šalju koordinate koristi se objekt *QgsMapToolEmitPoint* (Prilog 1, linija 120). Sljedeći korak je povezivanje tog alata pomoću signala za klikanje s metodom za odabir pojedine benzinske postaje naziva *selectFeature* (Prilog 1, linija 123). Ta metoda zbog svoje kompleksnosti će biti opisana kasnije u radu. Na kraju je potrebno definirati metodu s nazivom *selectActivated* koja izvršava aktivaciju alata za odabir (Prilog 1, linija 479-500). Prvo se provjerava je li gumb *Odaberi* aktiviran, ako je onda se izvršava ostatak koda u toj metodi. Prvo se isključuju svi ostali *QGIS* alati te se uključuje alat za odabir. Zatim se poziva metoda info trake *setVisible* koja će postaviti vidljivost trake te se ponovno osvježava podloga za iscrtavanje karte kako bi info traka bila vidljiva.

5.3.9. Aktivacijski okviri

Aktivacijski okvir (eng. *checkbox*) je grafički element koji ima dva stanja, uključen i isključen. Definira se instanciranjem klase *QCheckBox* te definiranjem pripadajuće oznake, obično se koristi za elemente u aplikaciji koji se mogu uključiti ili isključiti. U ovom slučaju aktivacijski okviri korišteni su za prikaz benzinskih postaja prema kriteriju vlasništva i vrsti goriva, u nastavku će biti opisan način izrade ta dva tipa aktivacijskih okvira. Sljedeći isječak koda prikazuje izradu aktivacijskog okvira za prikaz benzinskih postaja prema kriteriju vlasništva, definirano je šest aktivacijskih okvira prema istom principu. Treba napomenuti da je pritom korišten i dodatni *Python* modul *sets* kako bi se postigla željena funkcionalnost tih aktivacijskih okvira.

```

from sets import Set

#stvoren set koji će sadržavati uvjete za prikaz bp po vlasniku
self.benz_list = Set()

```

```

#stvaranje checkboxova za prikaz po vlasniku
cb1 = QCheckBox("Tifon", self)
cb1.stateChanged.connect(self.showTifon)

#stvaranje dodatnih informacija za korisnika
cb1.setToolTip("<b>Tifon benzinske postaje prikazane su crvenom bojom</b>")

def showTifon(self, state):

    #ako je uključen checkbox
    if state == Qt.Checked:
        #dodaj uvjet u set
        self.benz_list.add("Tifon")

    #ako je isključen checkbox
    else:
        #ukloni uvjet iz seta
        self.benz_list.remove("Tifon")

    #pozovi metodu za prikaz
    self.showbenz()

```

Za početak je učitana klasa *Set* iz modula *sets* (Prilog 1, linija 5) koja služi za spremanje niza neuređenih elemenata. Najvažnija osobina te klase je da ne može sadržavati više istih elemenata nego samo jedan. Zbog toga se i koristi u ovoj aplikaciji, spremljena kao vrijednost varijable *benz_list*. Klasa *Set* služi za spremanje uvjeta pomoću kojih se vrši prikaz benzinskih postaja prema kriteriju vlasništva. Nakon tih pripremnih koraka pristupa se stvaranju aktivacijskog okvira pomoću klase *QCheckBox* (Prilog 1, linija 133), pritom se definira njegov naziv (*Tifon*). Početno stanje tog aktivacijskog okvira je isključen. Zatim se povezuje njegova promjena stanja s pripadajućom metodom *showTifon* (Prilog 1, linija 134). Uz aktivacijski okvir napravljen je i okvir s dodatnim informacijama koje se prikazuju korisniku (Prilog 1, linija 157). Na kraju se definira metoda *showTifon* (Prilog 1, linija 503-516), koja provjerava stanje aktivacijskog okvira. Ako je aktivacijski okvir uključen onda se u skup naziva *benz_list* dodaje uvjet za prikaz *Tifon* pomoću kojeg se prikazuju benzinske postaje čiji je vlasnik Tifon. Ako je stanje aktivacijskog okvira promijenjeno u isključen onda se iz skupa uklanja uvjet *Tifon*. Nakon provjere stanja aktivacijskog okvira poziva se metoda za prikaz benzinskih postaja *showbenz* koja ovisno o uvjetima iz seta prikazuje ili ne prikazuje određene benzinske postaje. Metoda *showbenz* zbog svoje kompleksnosti bit će detaljno opisana u nastavku rada.

Sljedeći isječak koda prikazuje izradu aktivacijskog okvira za prikaz benzinskih postaja prema vrsti goriva, u aplikaciji ih je napravljeno četiri prema istom principu.

```

#stvaranje checkboxova za prikaz po vrsti goriva
cb7 = QCheckBox("Eurosuper 95", self)
cb7.stateChanged.connect(self.showSuper95)

```

```

#definirane variabile s početnom vrijednosti False budući da su checkboxovi za
gorivo u početku isključeni
self.super95 = False

#ovisno o statusu pojedinog checkboxa šalju se vrijednosti pripadajućih varijabli metodi za
priček
def showSuper95(self, state):

    #ako je uključen checkbox
    if state == Qt.Checked:
        #varijabla ima vrijednost True
        self.super95 = True

    #ako je isključen checkbox
    else:
        #varijabla ima vrijednost False
        self.super95 = False

    #pozovi metodu za priček
    self.showbenz()

```

Stvara se aktivacijski prozor pomoću klase *QCheckBox* (Prilog 1, linija 147), pritom se definira njegov naziv (*Eurosuper 95*). Zatim se povezuje njegova promjena stanja s pripadajućom metodom *showSuper95* (Prilog 1, linija 148). Nakon toga stvorena je varijabla *super95* koja sadrži početno stanje tog aktivacijskog okvira (Prilog 1, linija 168), na početku je isključen što je označeno s vrijednošću *False*. Na kraju se definira metoda *showSuper95* (Prilog 1, linija 595-609), koja provjerava stanje aktivacijskog okvira. Ako je aktivacijski okvir uključen onda se varijabli *super95* dodjeljuje vrijednost *True*. Ako je aktivacijski okvir isključen onda se varijabli *super95* dodjeljuje vrijednost *False*. Nakon provjere stanja aktivacijskog okvira poziva se metoda za priček benzinskih postaja *showbenz* koja ovisno o vrijednosti varijable *super95* prikazuje ili ne prikazuje benzinske postaje s tom vrstom goriva.

5.3.10. Glavna alatna traka

Glavna alatna traka je horizontalna traka na vrhu glavnog okvira koja sadrži osnovne alate i elemente koji se koriste prilikom interakcije s *FuelGIS* aplikacijom. Glavna alatna traka sadrži alate za približavanje, udaljavanje, pomicanje i odabir te aktivacijske okvire za priček benzinskih postaja prema vlasniku i vrsti goriva. Uz to sadrži i odgovarajuće natpise i opise kako bi korisnicima rad bio lakši i jednostavniji. Izrada glavne alatne trake opisana je u sljedećem isječku koda.

```

#stvaranje natpisa za glavnu alatnu traku
n1 = QLabel(u"      Prikaži benzinske postaje:  ")
n2 = QLabel(u"      Prikaži vrstu goriva:  ")

#stvaranje glavne alatne trake

```

```

toolbar = QToolBar()
self.addToolBar(toolbar)

#dodavanje akcija, checkboxova i natpisa u glavnu alatnu traku
toolbar.addAction(self.actionZoomIn)
toolbar.addAction(self.actionZoomOut)
toolbar.addAction(self.actionPan)
toolbar.addAction(self.actionSelect)
toolbar.addWidget(n1)
toolbar.addWidget(cb1)
toolbar.addWidget(cb2)
toolbar.addWidget(cb3)
toolbar.addWidget(cb4)
toolbar.addWidget(cb5)
toolbar.addWidget(cb6)
toolbar.addWidget(n2)
toolbar.addWidget(cb7)
toolbar.addWidget(cb8)
toolbar.addWidget(cb9)
toolbar.addWidget(cb10)

```

Prvo su napravljeni natpisi koji će biti smješteni u glavnoj alatnoj traci (Prilog 1, linija 129-130). Zatim je pomoću klase *QToolBar* stvorena glavna alatna traka te je smještena u glavni prozor pomoću naredbe *addToolBar*. Nakon toga su u glavnu alatnu traku pomoću naredbi *addAction* i *addWidget* dodani svi prethodno stvoren elementi: alati za približavanje, udaljavanje, pomicanje i odabir, natpisi koji opisuju aktivacijske okvire te deset aktivacijskih okvira za prikaz benzinskih postaja prema vlasniku i vrsti goriva. Način izrade glavne alatne trake vidljiv je u Prilogu 1, linija 173-193.

5.3.11. Info traka

Info traka je vertikalna traka s lijeve strane glavnog okvira koja sadrži natpise i pripadajuća polja za prikaz detaljnih informacija o odabranoj benzinskoj postaji. Obzirom na to da se sastoji od većeg broja natpisa i pripadajućih polja, cijeli postupak izrade prikazan je, zbog lakšeg opisa, u sljedeća dva isječka koda koje je moguće pronaći u sklopu Priloga 1, linija 195-277.

```

#stvaranje natpisa za info traku
info = QLabel("Detaljne informacije:")
info.setFixedWidth(230)
vlasnik = QLabel("Vlasnik:")
naziv = QLabel("Naziv:")
adresa = QLabel("Adresa:")
broj_tel = QLabel("Broj telefona:")
radno_vrijeme = QLabel("Radno vrijeme:")
eurosuper = QLabel("Eurosuper 95:")
europodiesel = QLabel("Europodiesel:")
autoplín = QLabel("Autoplín (LPG):")
plinske_boce = QLabel("Plinske boce:")
autopraonica = QLabel("Autopraonica:")

#stvaranje praznih polja na info traci za prikaz info o bp
self.vlasnik2 = QLabel("")
self.vlasnik2.setFixedHeight(20)

```

```

self.vlasnik2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.naziv2 = QLabel("")
self.naziv2.setFixedHeight(20)
self.naziv2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.adresa2 = QLabel("")
self.adresa2.setFixedHeight(20)
self.adresa2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.broj_tel2 = QLabel("")
self.broj_tel2.setFixedHeight(20)
self.broj_tel2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.radno_vrijeme2 = QLabel("")
self.radno_vrijeme2.setFixedHeight(20)
self.radno_vrijeme2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.eurosuper2 = QLabel("")
self.eurosuper2.setFixedHeight(20)
self.eurosuper2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.eurodiesel2 = QLabel("")
self.eurodiesel2.setFixedHeight(20)
self.eurodiesel2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.autoplin2 = QLabel("")
self.autoplin2.setFixedHeight(20)
self.autoplin2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.plinske_boce2 = QLabel("")
self.plinske_boce2.setFixedHeight(20)
self.plinske_boce2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
self.autopraonica2 = QLabel("")
self.autopraonica2.setFixedHeight(20)
self.autopraonica2.setFrameStyle(QFrame.Panel | QFrame.Sunken)

```

Prvi korak bio je izrada natpisa za info traku koji će opisivati prikazane informacije, tako je napravljeno nekoliko varijabli koje sadrže opisne informacije. Pritom je određena i dužina tih natpisa kako bi bili adekvatno prikazani. Nakon toga su napravljena pripadajuća polja pomoću *QLabel* klase u kojima će se prikazivati detaljne informacije o odabranoj benzinskoj postaji. Za svako polje određena je njegova visina te stil prikaza njegovog okvira.

```

#stvaranje vertikalnog graničnika
splitter = QSplitter(Qt.Vertical)

#stvaranje info trake
self.infoToolbar = QToolBar()

#staviti info traku s lijeve strane
self.addToolBar(Qt.ToolBarArea(Qt.LeftToolBarArea), self.infoToolbar)

#sakriti info traku
self.infoToolbar.setVisible(False)

#dodavanje graničnika u info traku
self.infoToolbar.addWidget(splitter)

#dodavanje natpisa i pripadajućih polja u graničnik
splitter.addWidget(info)
splitter.addWidget(vlasnik)
splitter.addWidget(self.vlasnik2)
splitter.addWidget(naziv)
splitter.addWidget(self.naziv2)
splitter.addWidget(adresa)
splitter.addWidget(self.adresa2)
splitter.addWidget(broj_tel)
splitter.addWidget(self.broj_tel2)
splitter.addWidget(radno_vrijeme)
splitter.addWidget(self.radno_vrijeme2)
splitter.addWidget(eurosuper)

```

```

splitter.addWidget(self.eurosuper2)
splitter.addWidget(eurodiesel)
splitter.addWidget(self.eurodiesel2)
splitter.addWidget(autoplín)
splitter.addWidget(self.autoplín2)
splitter.addWidget(plinske boce)
splitter.addWidget(self.plinske boce2)
splitter.addWidget(autopraonica)
splitter.addWidget(self.autopraonica2)

```

Zatim je napravljen vertikalni graničnik pomoću *QSplitter* klase koji služi za međusobno razdvajanje svih prethodno stvorenih natpisa i pripadajućih polja. Konačno je stvorena info traka te je pomoću naredbe *addToolBar* smještena s lijeve strane glavnog okvira. Uz to info traku je sakrivena tako da se ne pokazuje dok se ne aktivira alat za odabir opisan u potpotpoglavlju 5.3.8. U info traku prvo se dodaje vertikalni graničnik te se nakon toga u njega dodaju svi natpisi i polja.

5.3.12. Zatvaranje *FuelGIS* aplikacije

Unatoč tome što zatvaranje aplikacije pripada osnovnim funkcionalnostima svake aplikacije ono se može usavršiti izradom dodatne metode. U ovom slučaju zatvaranje aplikacije se želi usavršiti kako ne bi slučajnim odabirom gumba *Close* došlo do neželjenog zatvaranja aplikacije. Nakon klika na gumb *Close* (X) smještenog u gornjem desnom uglu glavnog okvira odašilje se događaj (eng. *event*) koji uzrokuje zatvaranje glavnog prozora a time i aplikacije. No ako se želi promijeniti ponašanje glavnog okvira prilikom odabira gumba za zatvaranje potrebno je napraviti promjene u upravljaču tog događaja (eng. *event handler*). Definicijom metode *closeEvent* mijenja se ponašanje glavnog okvira prilikom odabira gumba za zatvaranje. Navedena metoda je opisana u sljedećem isječku koda koji je dostupan u Prilogu 1, linija 741-755.

```

#metoda za izlazak iz aplikacije
def closeEvent(self, event):

    #definiranje prozora s porukom te mogućim izborima
    reply = QMessageBox.question(self, "Poruka",
        u"Želite ugasiti FuelGIS aplikaciju?", QMessageBox.Yes |
        QMessageBox.No, QMessageBox.Yes)

    #ako se odabere "Yes" aplikacija se gasi
    if reply == QMessageBox.Yes:
        event.accept()

    #ako se odabere "No" aplikacija i dalje ostaje upaljena
    else:
        event.ignore()

```

Nakon klika na gumb za zatvaranje metodi *closeEvent* se šalje događaj nastao tim klikom. Nakon toga se prikazuje dijaloški okvir *Poruka* s upitom i ponuđenim mogućnostima. Kao odgovori na upit ponuđene su opcije *Yes* i *No* s time da je opcija *Yes* već označena kao aktivna. Ukoliko se odabere opcija *Yes* dolazi do zatvaranja glavnog okvira i aplikacije. U suprotnom, odabirom opcije *No* aplikacija ostaje i dalje upaljena. Treba napomenuti kako nije bilo moguće lokalizirati nazine ponuđenih mogućnosti s engleskog na hrvatski jer su oni kao takvi definirani u događaju za zatvaranje. Stoga su ostavljeni u izvornom obliku kako bi se metoda mogla ispravno izvršiti. Definicijom metode za zatvaranje završena je izrada sučelja te se pristupilo sljedećem koraku u izradi *FuelGIS* aplikacije.

5.4. Implementacija *QGIS* funkcionalnosti u *FuelGIS* aplikaciju

Implementacija *QGIS* funkcionalnosti u *FuelGIS* aplikaciju temeljila se na korištenju biblioteke *PyQGIS*, tj. njenih modula *core* i *gui* koji sadrže potrebne klase i metode. Također, uz njih je korištena i biblioteka *os* sa svojim pripadajućim klasama i metodama. Treba napomenuti da je dio *QGIS* funkcionalnosti poput podloge za iscrtavanje karte te *QGIS* alata za interakciju sa slojevima već opisan u prethodnom potpoglavlju. Razlog tome je isprepletenost *PyQt4* sučelja te *QGIS* funkcionalnosti, što otežava njihovo razdvajanje te zasebno opisivanje. Najbolji primjer toga je podloga za iscrtavanje karte (eng. *map canvas*) koja je sastavni dio sučelja unatoč tome što pripada *QGIS* funkcionalnostima. Osim prethodno opisanih *QGIS* funkcionalnosti korištene su još neke:

- koordinatni sustav,
- rad s rasterskim i vektorskim slojem,
- odabir željenog vektorskog elementa te prikaz njegovih atributa,
- prikaz željenog sadržaja,
- simbologija tj. način prikazivanja elemenata vektorskog sloja na podlozi za iscrtavanje karte.

U nastavku će biti detaljno opisan način izrade svih tih funkcionalnosti.

5.4.1. Početni parametri *PyQGIS* aplikacije

Kako bi *Python* kod ispravno radio potrebno je na samom početku učitati potrebne module i njihove komponente (klase, funkcije i metode) koje će biti korištene u dalnjem pisanju koda. Sljedeći isječak koda prikazuje naredbe kojima se određuju početni parametri *PyQGIS* aplikacije.

```
from qgis.core import QgsRasterLayer, QgsVectorLayer, QgsMapLayerRegistry, QgsApplication,
QgsCoordinateReferenceSystem, QgsFeature, QgsGeometry, QgsSymbolV2, QgsRendererCategoryV2,
QgsCategorizedSymbolRendererV2
from qgis.gui import QgsMapCanvas, QgsMapCanvasLayer, QgsMapToolPan, QgsMapToolZoom,
QgsMapToolEmitPoint
import os
```

Učitani su moduli *core* i *gui* (Prilog 1, linija 6-7), njihova uloga je prethodno opisana u potpoglavlju 2.9. Treba napomenuti da su se tokom pisanja koda te naredbe nadopunjavale potrebnim komponentama. Na taj način se koriste potrebne komponente umjesto nepotrebnog učitavanja cijelog sadržaja navedenih modula. Tako se racionalno koristi memorijski prostor prilikom pokretanja aplikacije. Zatim je učitan modul *os* (Prilog 1, linija 12) koji omogućuje korištenje funkcionalnosti ovisnih o operativnom sustavu. U ovom slučaju se koristi za čitanje vrijednosti varijabli okoline tj. putanja potrebnih za korištenje *QGIS* funkcionalnosti.

5.4.2. Pozivanje *QGIS* okoline

Kako bi *QGIS* funkcionalnosti mogle ispravno raditi unutar *FuelGIS* aplikacije potrebno je pozvati odgovarajuću okolinu. Također, prilikom zatvaranja *FuelGIS* aplikacije potrebno je i ugasiti *QGIS* okolinu. U sljedećem isječku koda prikazane su naredbe potrebne za pozivanje i gašenje *QGIS* okoline.

```
qgis_prefix = os.getenv("QGISHOME")
#potrebno je inicijalizirati aplikaciju u QGIS okolini
QgsApplication.setPrefixPath(qgis_prefix, True)
QgsApplication.initQgis()

#počisti nakon izvršavanja aplikacije
QgsApplication.exitQgis()
```

Prvo se pomoću metode *getenv* iz *os* modula dobiva vrijednost varijable *QGISHOME* tj. instalacijska putanja od *QGIS* aplikacije (Prilog 1, linija 13). Ta putanja se kasnije koristi za inicijalizaciju *FuelGIS* aplikacije u *QGIS* okolini pomoću metoda klase *QgsApplication*.

Prvo se postavlja prethodno dobivena putanja kao mjesto gdje se nalaze potrebni parametri za inicijalizaciju *QGIS* okoline. Zatim se pozivom metode *initQgis* izvršava inicijalizacija *FuelGIS* aplikacije u *QGIS* okolini (Prilog 1, linija 761-762). Prilikom zatvaranja *FuelGIS* aplikacije dolazi i do gašenja *QGIS* okoline pomoću naredbe *exitQgis* (Prilog 1, linija 772).

5.4.3. Koordinatni sustav

Kako bi se rasterski i vektorski slojevi mogli pravilno smjestiti u prostoru potrebno je definirati koordinatni sustav. Budući da je prethodno obavljena obrada rasterskih i vektorskih podataka (opisana u poglavlju 4.) njima je pridružen Hrvatski državni koordinatni sustav (HDKS). Stoga je bilo potrebno definirati taj koordinatni sustav i unutar *FuelGIS* aplikacije kako bi se rasterski i vektorski sloj ispravno smjestili i prikazali u prostoru. Sljedeći isječak koda (Prilog 1, linija 69-71) prikazuje izradu Hrvatskog državnog koordinatnog sustava pomoću *PyQGIS-a*.

```
#definiranje koordinatnog sustava
crs = QgsCoordinateReferenceSystem()
crs.createFromProj4("+proj=tmerc +lat 0=0 +lon 0=15 +k=0.9999 +x 0=5500000 +y 0=0
+ellps=bessel +towgs84=514.0188,155.448,507.0461,5.6136,3.676,-11.4667,0.9999979090043
+units=m +no_defs")
```

Pomoću klase *QgsCoordinateReferenceSystem* stvori se koordinatni sustav, zatim se definiraju njegovi parametri koji su preuzeti s Interneta (URL 17). Parametri moraju biti izraženi u *PROJ.4* formatu kako bi se ispravno definirao koordinatni sustav. Na taj način pozvani slojevi su ispravno smješteni u prostoru te se pomakom miša po podlozi za iscrtavanje karte u polju na statusnoj traci prikazuju odgovarajuće koordinate.

5.4.4. Rad s rasterskim i vektorskим slojem

Jedna od glavnih funkcionalnosti *QGIS-a* je njegova mogućnost rada s raznim rasterskim i vektorskim formatima odnosno rasterskim i vektorskim slojevima. U *FuelGIS* aplikaciji koristi se DOF 2006 kao rasterska podloga te *ESRI Shape* datoteka *benz*, s položajnim i atributnim podacima o benzinskim postajama, kao vektorski sloj. Kako bi se te datoteke mogle koristiti unutar *FuelGIS* aplikacije potrebno ih je dodati u obliku pripadajućih slojeva te ih prikazati na podlozi za iscrtavanje karte. Sljedeći isječak koda (Prilog 1, linija 73-81) prikazuje rad s rasterskim i vektorskim slojem.

```
#definiranje pojedinih slojeva i njihovo dodavanje u registar slojeva
```

```

layer1 = QgsRasterLayer("E:/FuelGIS/Zagreb.tif", "Zagreb")
self.canvas.setExtent(layer1.extent())
QgsMapLayerRegistry.instance().addMapLayer(layer1)
self.layer2 = QgsVectorLayer("E:/FuelGIS/benz.shp", "benzinske", "ogr")
QgsMapLayerRegistry.instance().addMapLayer(self.layer2)

#potrebno dodati slojeve u map canvas da bi se prikazali
self.canvas.setLayerSet([QgsMapCanvasLayer(self.layer2),
QgsMapCanvasLayer(layer1)])

```

Za početak potrebno je definirati svaki pojedini sloj, prvo je definiran rasterski sloj pomoću klase *QgsRasterLayer*, pritom treba definirati putanju gdje se nalazi rasterska datoteka te naziv pod kojim će sloj biti prikazan. Pomoću naredbe *setExtent* mjerilo prikaza podloge za iscrtavanje karte prilagođeno je veličini rasterske datoteke tako da se može vidjeti sav njen sadržaj. Nakon toga taj sloj se dodaje u registar slojeva karte (eng. *Map Layer Registry*) koji služi za upravljanje slojevima te za njihovo iscrtavanje. Zatim je definiran vektorski sloj pomoću klase *QgsVectorLayer*, pritom treba uz putanju te naziv sloja navesti i naziv pružatelja podataka. Kao pružatelj podataka za *ESRI Shape* datoteke koristi se *OGR* biblioteka stoga je potrebno navesti njen naziv *ogr*. *OGR* omogućava manipulaciju velikim brojem prostornih podataka, upotrebom standardnog programskog sučelja. U ovom slučaju *OGR* se koristi za učitavanje geometrije iz *ESRI Shape* datoteke. Vektorski sloj je također dodan u registar slojeva karte kako bi se mogao iscrtati i koristiti. Za kraj potrebno je dodati prethodno definirane slojeve kao slojeve karte na podlogu za iscrtavanje karte kako bi se prikazali.

5.4.5. Odabir željenog vektorskog elementa i prikaz njegovih atributa

Jedna od *QGIS* funkcionalnosti je mogućnost odabira pojedinog vektorskog elementa nastalog iz *ESRI Shape* datoteke. Pritom je moguće dobiti detaljne informacije o tom elementu iz tablice atributnih podataka koji se također nalaze u *ESRI Shape* datoteci. Navedene mogućnosti imaju važnu ulogu u *FuelGIS* aplikaciji jer je pomoću njih moguće odabrati željenu benzinsku postaju te dobiti detaljne informacije o njoj kroz info traku. Početni dio vezan uz izradu gumba za odabir te njegovo povezivanje s objektom *QgsMapTool Emit Point* je već objašnjen u potpotpoglavlju 5.3.8. Tada je također opisano i povezivanje tog alata pomoću signala za klikanje s metodom za odabir pojedine benzinske postaje naziva *selectFeature*. Ovdje će biti objašnjen daljnji tijek događaja, što se točno događa kad se korištenjem alata za odabir klikne na neko mjesto unutar podloge za

iscrtavanje karte tj. kako radi metoda za odabir *selectFeature*. Navedena metoda je opisana u sljedećem isječku koda dostupnom u Prilogu 1, linija 369-443.

```
#metoda za odabir bp i prikaz detaljnih informacija o njoj u info traci
def selectFeature(self, point):

    #napravi točku od koordinata koje su poslane od strane kontrole
    pntGeom = QgsGeometry.fromPoint(point)

    #napravi buffer oko te točke od 5 piksela ovisno o stupnju zooma (ovisno koliko je
    #stvarnih jedinica u jednom pikselu)
    pntBuff = pntGeom.buffer( (self.canvas.mapUnitsPerPixel() * 5),0)

    #stvori granični okvir za krug tj. buffer
    rect = pntBuff.boundingBox()

    #vektorski sloj je pružatelj podataka
    dp = self.layer2.dataProvider()

    #dohvati sve atribute
    allAttr = dp.attributeIndexes()

    #definirana lista za bp
    selectList = []

    #benzinska postaja je klasa QgsFeature
    bp = QgsFeature()

    #odaberite bp s ovim atributima koristeći granični okvir
    dp.select([3,4,5,6,7,8,9,10,11,12], rect)
    while dp.nextFeature(bp):

        #ako se geometrija bp siječe s geometrijom buffera stavi na listu
        if bp.geometry().intersects(pntBuff):
            selectList.append(bp.id())

        #izvuci podatke
        attrs = bp.attributeMap()

        #prikaži željene podatke
        vlasnik = attrs[3]
        self.vlasnik2.setText(vlasnik.toString())
        naziv = attrs[4]
        self.naziv2.setText(naziv.toString())
        adresa = attrs[5]
        self.adresa2.setText(adresa.toString())
        broj_tel = attrs[6]
        self.broj_tel2.setText(broj_tel.toString())
        radno_vrijeme = attrs[7]
        self.radno_vrijeme2.setText(radno_vrijeme.toString())
        eurosuper = attrs[8]
        self.eurosuper2.setText(eurosuper.toString())
        eurodiesel = attrs[9]
        self.eurodiesel2.setText(eurodiesel.toString())
        autoplin = attrs[10]
        self.autoplin2.setText(autoplin.toString())
        plinske_boce = attrs[11]
        self.plinske_boce2.setText(plinske_boce.toString())
        autopraonica = attrs[12]
        self.autopraonica2.setText(autopraonica.toString())

        #čim nađeš prvu točku (bp), završi s petljom
        break

    #ako ni jedna točka (bp) nije pronađena, počisti prethodne informacije
    if len(selectList) == 0:
        self.vlasnik2.clear()
        self.naziv2.clear()
        self.adresa2.clear()
        self.broj_tel2.clear()
```

```

    self.radno_vrijeme2.clear()
    self.eurosuper2.clear()
    self.eurodiesel2.clear()
    self.autoplinske2.clear()
    self.plinske_boce2.clear()
    self.autopraonica2.clear()

    #odaberi pronađenu točku (bp)
    self.layer2.setSelectedFeatures(selectList)

```

Klikom miša na podlogu za iscrtavanje karte objekt *QgsMapToolEmitPoint* šalje metodi *selectFeature* koordinate mjesta na kojem je klik napravljen. Metoda od tih koordinata napravi točku pomoću klase *QgsGeometry*. Nakon toga se napravi tampon područje (eng. *buffer*) oko nje u iznosu od pet piksela ovisno o stupnju povećanja (eng. *zoom*) tj. ovisno o tome koliko je stvarnih jedinica u jednom pikselu. Nakon toga stvori se granični okvir za to tampon područje koji služi kako bi se smanjilo područje pretrage, a time i ubrzao postupak pretrage. Nakon toga se kao izvor tj. pružatelj podataka postavlja vektorski sloj definiran u varijabli *layer2* te se iz njega preuzimaju atributni podaci koji nas zanimaju. Zatim je napravljena lista koja će sadržavati pronađenu benzinsku postaju koju je potrebno odabrati. Sljedećim korakom je definiran *QgsFeature* objekt, koji će privremeno sadržavati geografske objekte tj. benzinske postaje vektorskog sloja koji se pretražuje. Nakon toga započinje postupak selekcije tj. obavlja se presjek graničnog okvira sa svim objektima vektorskog sloja. Zatim se za svaki geografski objekt koji zadovoljava uvjet geografskog okvira provjerava presjeca li se njegova geometrija s geometrijom tampon zone. Ako se te dvije geometrije sijeku onda se ta benzinska postaja stavlja u listu za odabir te se odabire na vektorskome sloju. Pritom se pomoću metode *attributeMap* preuzimaju željeni atributni podaci od te benzinske postaje pozivanjem pripadajućih rednih brojeva. Važno je naglasiti da se vrijednosti atributa pretvaraju u niz znakova pomoću metode *toString* kako bi bili razumljivi korisniku te se zatim prikazuju u pripadajućim poljima u info traci pomoću metode *setText*. Treba napomenuti da čim prva točka zadovolji uvjet presjeka s tampon područjem pretraga se završava jer se želi odabrati samo jedan objekt. Ukoliko niti jedna benzinska postaja nije pronađena tj. nije zadovoljila postavljeni uvjet dolazi do brisanja prethodno prikazanih informacija iz polja u info traci. U tom slučaju potrebno je ponovno kliknuti mišem po podlozi za iscrtavanje karte kako bi se ponovno izvršila metoda *selectFeature* te odabrala željena benzinska postaja.

5.4.6. Prikaz željenog sadržaja

Ovisno o postavljenom upitu korisnika putem aktivacijskih okvira moguće je na podlozi za iscrtavanje karte dobiti prikaz željenog sadržaja. Navedena *QGIS* funkcionalnost je jako korisna jer omogućuje da se upotrebnom aktivacijskih okvira postavi upit prema kojem se u *FuelGIS* aplikaciji prikazuje željeni sadržaj tj. moguće je dobiti prikaz benzinskih postaja prema vlasniku i vrsti goriva. Početni dio vezan uz izradu aktivacijskih okvira je već objašnjen u potpotpoglavlju 5.3.9. Tada je također opisano i povezivanje tih aktivacijskih okvira s pripadajućom metodom. Ovdje će biti objašnjen daljnji tijek događaja, što se točno događa nakon provjere stanja aktivacijskog okvira kada se poziva metoda za prikaz benzinskih postaja *showbenz*. Ta metoda ovisno o uvjetima iz skupa i vrijednostima pojedinih varijabli goriva prikazuje ili ne prikazuje određene benzinske postaje. Metoda *showbenz* je opisana u sljedećem isječku koda dostupnom u Prilogu 1, linija 656-739.

```
#metoda za prikaz bp po vlasniku i vrsti goriva
def showbenz(self):

    #upit na početku ne sadrži nikakve uvjete
    upit = ''

    #ako je duljina seta veća od 0 tj. ako u setu postoji kakav uvjet
    if len(self.benz_list)>0:

        #definirana lista koja će sadržavati nazive kao vrijednosti atributa
        bp_lista=[]

        #za svaku bp u setu
        for benz in self.benz_list:

            #uzmi naziv bp i dodaj " " jer je naziv vrijednost atributa
            naziv_benz = "'"+benz+"'"

            #dodaj naziv u obliku vrijednosti atributa na listu
            bp_lista.append(naziv_benz)

        #stavi zarez između svakoga elementa u listi
        odabrane_bp = ','.join(bp_lista)

        #stvoren upit prema odabranim bp s liste
        upit = "Vlasnik in ("+odabrane_bp+")"

        #ako varijabla ima vrijednost True
        if self.super95 == True:

            #ako već postoji kakav uvjet u upitu
            if upit != '':
                #unesi u upit dodatni uvjet
                upit = upit + ' and "Eurosuper" = "Da"'

            #ako ne postoji nikakav uvjet u upitu
            else:
                #unesi u upit novi uvjet
                upit = upit + '"Eurosuper" = "Da"'

        #ako varijabla ima vrijednost True
```

```

if self.diesel == True:

    #ako već postoji kakav uvjet u upitu
    if upit != '':
        #unesi u upit dodatni uvjet
        upit = upit + 'and "Eurodiesel" = "Da"'


    #ako ne postoji nikakav uvjet u upitu
    else:
        #unesi u upit novi uvjet
        upit = upit + '"Eurodiesel" = "Da"'


#ako varijabla ima vrijednost True
if self.plin == True:

    #ako već postoji kakav uvjet u upitu
    if upit != '':
        #unesi u upit dodatni uvjet
        upit = upit + 'and "Autoplín" = "Da"'


    #ako ne postoji nikakav uvjet u upitu
    else:
        #unesi u upit novi uvjet
        upit = upit + '"Autoplín" = "Da"'


#ako varijabla ima vrijednost True
if self.boce == True:

    #ako već postoji kakav uvjet u upitu
    if upit != '':
        #unesi u upit dodatni uvjet
        upit = upit + 'and "Plinske bo" = "Da"'


    #ako ne postoji nikakav uvjet u upitu
    else:
        #unesi u upit novi uvjet
        upit = upit + '"Plinske bo" = "Da"'


#postavi zadanom sloju upit s odabranim uvjetima
self.layer2.setSubsetString(upit)

#ponovno iscrtavanje slojeva u QgsMapCanvas-u
self.canvas.refresh()

```

Ovisno o stanju aktivacijskih okvira metoda *showbenz* provjerava vrijednost varijabli *benz_list*, *super95*, *diesel*, *plin* i *boce*. Varijabla *benz_list* je lista koja sadrži popis naziva pomoću kojih se vrši prikaz benzinskih postaja prema vlasniku. Ostale nabrojane varijable služe kao uvjeti za prikaz benzinskih postaja prema vrsti goriva te sadrže samo jednu vrijednost (*True* ili *False*). U početku upit ne sadrži nikakve uvjete te se prikazuju sve benzinske postaje. Prvo se provjerava sadrži li varijabla *benz_list* kakve uvjete za prikaz benzinskih postaja po vlasniku. To se radi provjerom broja elemenata skupa koji predstavlja ta varijabla. Ako je broj elemenata skupa veći od nule onda sadrži uvjete te se izvršava pripadajući blok naredbi. Stvorit će se lista naziva *bp_lista* koja će sadržavati nazive iz varijable *benz_list* kao vrijednosti atributa. Zatim se izvršava petlja, niz naredbi, ovisno o broju elemenata u varijabli *benz_list*. Svaki element iz varijable *benz_list* se dodaje na popis atributa u varijablu *bp_lista*, pritom se atributi međusobno odvajaju zarezom. Na

kraju se prema atributima iz varijable *bp_lista* postavlja upit za prikaz benzinskih postaja prema vlasniku. Upit je analogan *WHERE* uvjetu *SQL* naredbe te se mogu koristiti isti operateri, ovdje se koristi operator *IN*. Zatim se provjeravaju vrijednosti varijabli *super95*, *diesel*, *plin* i *boce*. Za svaku se provjerava vrijednost. Ukoliko je *True*, provjerava se postoji li već uvjet u upitu. Ako postoji neki uvjet onda se u upit unosi dodatni uvjet za prikaz prema vrsti goriva, u suprotnom se u upit unosi novi uvjet za prikaz prema vrsti goriva. Dodatni uvjet se razlikuje po tome što se ispred uvjeta dodaje ključna riječ *+AND+* koja omogućuje logičko spajanje uvjeta. Ako nije definiran niti jedan uvjet onda nije potreban *AND*. Navedeni postupak se provodi za svaku varijablu te se na kraju ovisno o vrijednostima tih varijabli dobije upit za prikaz benzinskih postaja prema vrsti goriva. U slučaju da postoje uvjeti i za prikaz prema vlasniku prikazuje se sadržaj tj. benzinske postaje koje zadovoljavaju sve navedene uvjete iz upita. Na kraju se pomoću metode *setSubsetString* vektorskog sloju postavi *SQL* upit s odabranim uvjetima te dolazi do ponovnog iscrtavanja slojeva kako bi se prikazao željeni sadržaj. Svakom promjenom stanja od strane korisnika bilo kojeg od deset aktivacijskih okvira dolazi do promjene uvjeta a samim time do promjene upita te se metoda *showbenz* iznova izvršava kako bi se prikazao željeni sadržaj.

5.4.7. Simbologija – izgled vektorskog sloja

Kada se vektorski sloj iscrtava izgled njegovih elemenata ovisi o biblioteci za iscrtavanje (eng. *renderer*) i simbolima povezanim s tim slojem. Simboli su klase zadužene za vizualni prikaz elemenata sloja dok biblioteka za iscrtavanje određuje koji simbol će biti korišten za prikaz određenog elementa sloja.

Simbologija u *FuelGIS* aplikaciji bila je važna jer je trebalo prema vlasniku vizualno razlikovati pojedine benzinske postaje. Za njihov prikaz korištena je nova simbologija koja ima poboljšane biblioteke za iscrtavanje s širim rasponom mogućnosti prilikom izrade simbola. Pritom je korištena biblioteka za iscrtavanje koja omogućuje iscrtavanje elemenata vektorskog sloja pomoću različitih simbola ovisno o kategoriji tj. vrijednosti atributa. Na taj način je napravljen prikaz vektorskog sloja u šest kategorija tj. šest vrsta simbola koji su bili korišteni za prikaz benzinskih postaja prema vlasniku. Pritom svaki simbol ima svoje atribute poput naziva, vrijednosti i boje te se sukladno njima i

iscrtava. Stvaranje simbola te određivanje načina iscrtavanja je opisano u sljedećem isječku koda dostupnom u Prilogu 1, linija 279-367.

```
#stvorena lista koja će sadržavati kategorije za iscrtavanje
cat_list = []

#definiran prvi simbol
symbol1_label = QString("Tifon")
symbol1_value = QVariant("Tifon")
symbol1_color = QColor(Qt.red)
symbol1 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
symbol1.setColor(symbol1_color)

#definirana prva kategorija za iscrtavanje
cat1 = QgsRendererCategoryV2(symbol1_value, symbol1, symbol1_label)

#prva kategorija dodana na listu za iscrtavanje
cat_list.append(cat1)

#definiran drugi simbol
symbol2_label = QString("OMV")
symbol2_value = QVariant("OMV")
symbol2_color = QColor(Qt.blue)
symbol2 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
symbol2.setColor(symbol2_color)

#definirana druga kategorija za iscrtavanje
cat2 = QgsRendererCategoryV2(symbol2_value, symbol2, symbol2_label)

#druga kategorija dodana na listu za iscrtavanje
cat_list.append(cat2)

#definiran treći simbol
symbol3_label = QString("Lukoil")
symbol3_value = QVariant("Lukoil")
symbol3_color = QColor(Qt.green)
symbol3 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
symbol3.setColor(symbol3_color)

#definirana treća kategorija za iscrtavanje
cat3 = QgsRendererCategoryV2(symbol3_value, symbol3, symbol3_label)

#treća kategorija dodana na listu za iscrtavanje
cat_list.append(cat3)

#definiran četvrti simbol
symbol4_label = QString("EuroPETROL")
symbol4_value = QVariant("EuroPETROL")
symbol4_color = QColor(Qt.cyan)
symbol4 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
symbol4.setColor(symbol4_color)

#definirana četvrta kategorija za iscrtavanje
cat4 = QgsRendererCategoryV2(symbol4_value, symbol4, symbol4_label)

#četvrta kategorija dodana na listu za iscrtavanje
cat_list.append(cat4)

#definiran peti simbol
symbol5_label = QString("Petrol")
symbol5_value = QVariant("Petrol")
symbol5_color = QColor(Qt.magenta)
symbol5 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
symbol5.setColor(symbol5_color)

#definirana peta kategorija za iscrtavanje
cat5 = QgsRendererCategoryV2(symbol5_value, symbol5, symbol5_label)

#peta kategorija dodana na listu za iscrtavanje
cat_list.append(cat5)
```

```

#definiran šesti simbol
symbol6_label = QString("INA")
symbol6_value = QVariant("INA")
symbol6_color = QColor(Qt.yellow)
symbol6 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
symbol6.setColor(symbol6_color)

#definirana šesta kategorija za iscrtavanje
cat6 = QgsRendererCategoryV2(symbol6_value, symbol6, symbol6_label)

#šesta kategorija dodana na listu za iscrtavanje
cat_list.append(cat6)

#definirana vrsta biblioteke za iscrtavanje, iscrtava se po kategorijama
renderer = QgsCategorizedSymbolRendererV2('', cat_list)

#definiran atribut po kojem se iscrtava
renderer.setClassAttribute("Vlasnik")

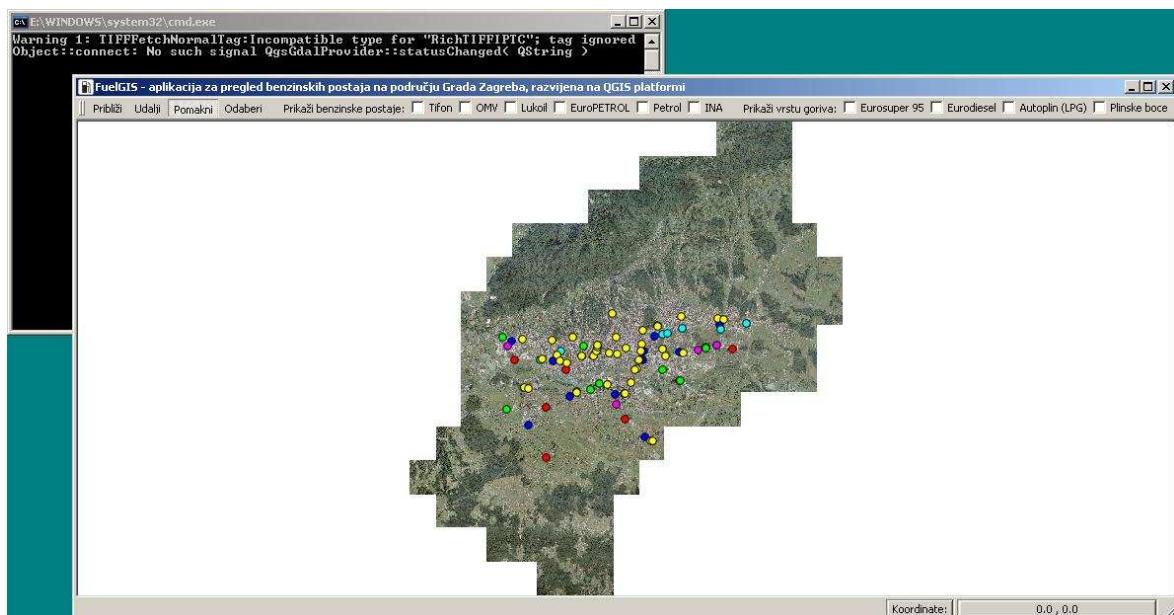
#simboli u vektorskem sloju se iscrtavaju po kategorijama
self.layer2.setRendererV2(renderer)

```

Na početku je napravljena lista s nazivom *cat_list* koja će sadržavati popis kategorija za iscrtavanje. Zatim se pristupa izradi prvog simbola pomoću klase *QgsSymbolV2*, pritom on dobiva naziv, vrijednost, boju te se postavlja za korištenje u prikazu vektorskog sloja. Nakon toga se definira prva kategorija za iscrtavanje koja sadrži prvi simbol te sve njegove prethodno stvorene attribute. Tako definirana prva kategorija dodaje se na listu za iscrtavanje. Navedeni postupak ponavlja se za svih šest simbola tj. šest kategorija. Kada su napravljeni svi simboli te pripadajuće kategorije dodane u listu za iscrtavanje definira se vrsta biblioteke za iscrtavanje. Pomoću klase *QgsCategorizedSymbolRendererV2* određeno je iscrtavanje po kategorijama i to prema onim navedenim u varijabli *cat_list*. Uz to potrebno je navesti naziv atributa kako bi biblioteka za iscrtavanje znala po vrijednostima kojeg atributa se obavlja iscrtavanje. Na kraju je potrebno vektorskem sloju pridružiti odabranu biblioteku za iscrtavanje. Time je dobiven željeni način prikaza benzinskih postaja prema vlasniku, svaki vlasnik imao je prikazane benzinske postaje u različitoj boji. Ovim korakom je ujedno i završena implementacija potrebnih *QGIS* funkcionalnosti te izrada *FuelGIS* aplikacije.

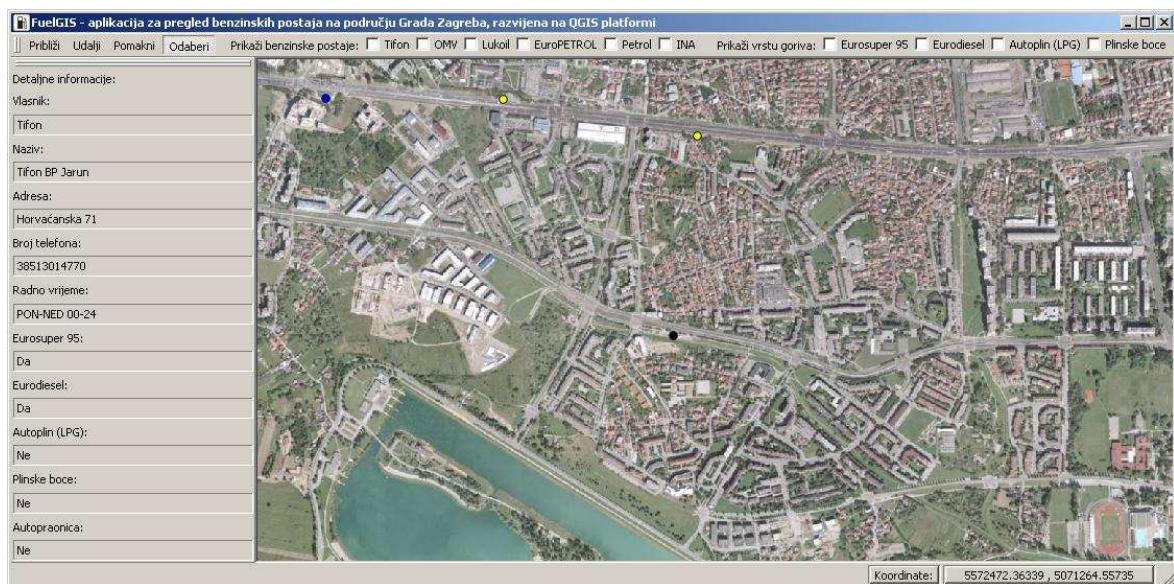
6. Upute za rad s FuelGIS aplikacijom

Kako bi *FuelGIS* aplikacija mogla ispravno raditi moraju biti ispunjeni određeni preduvjeti. Potrebno je instalirati te ispravno podesiti postavke *OSGeo4W* programskog paketa kako bi njegove komponente, potrebne *FuelGIS* aplikaciji, mogle ispravno raditi u *Windows* okruženju. To je moguće napraviti prema uputama opisanim u potpoglavljima 5.1 i 5.2. Također treba voditi računa o lokaciji potrebnih datoteka tj. o njihovim putanjama podešenim u samom kodu *FuelGIS* aplikacije. Navedeno se odnosi na datoteke *Zagreb.tiff*, *benz.shp* i *bp.png* koje služe kao izvori za dodavanje rasterskog i vektorskog sloja te kao ikona glavnog okvira. Ukoliko postoji razlika potrebno je pomoću uređivača teksta ažurirati te putanje kako bi *FuelGIS* aplikacija mogla pristupiti potrebnim datotekama. Uz to *FuelGIS.bat* i *FuelGIS.py* datoteke moraju se nalaziti u istom direktoriju kako bi se pokretanjem *bat* datoteke izvršio napisani kod iz *py* datoteke. Jedino ako su zadovoljeni svi navedeni uvjeti moguće je ispravan rad *FuelGIS* aplikacije na bilo kojem računalu. Pokretanje aplikacije obavlja se pomoću *FuelGIS.bat* datoteke što uzrokuje otvaranje *Python* interpretera te glavnog okvira *FuelGIS* aplikacije (Slika 31.).



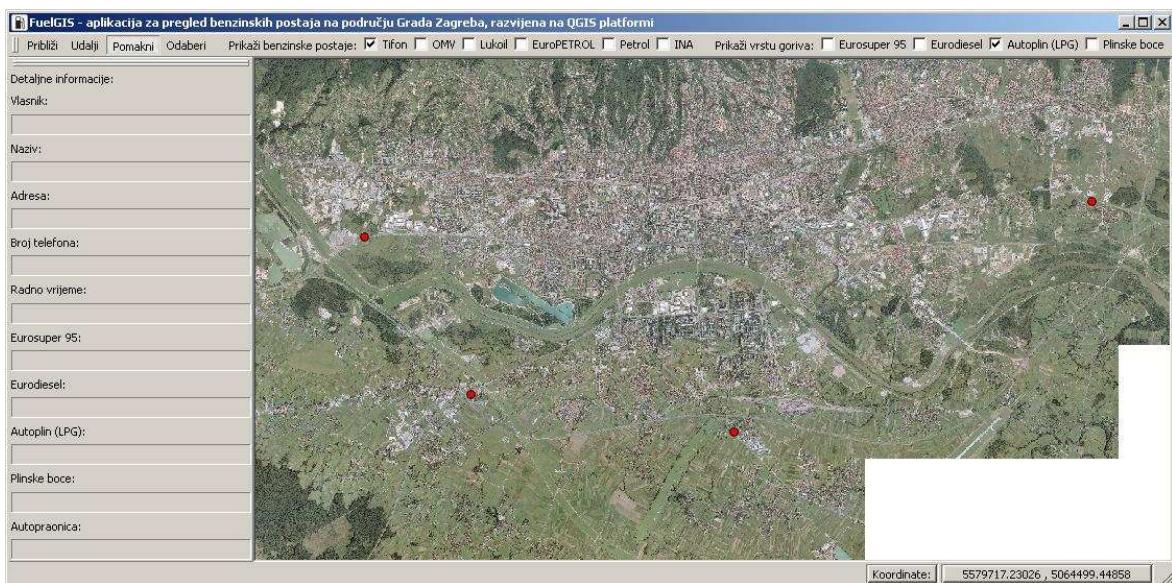
Slika 31. Početni izgled FuelGIS aplikacije

Pri vrhu glavnog okvira nalazi se glavna alatna traka s alatima za interakciju s prikazanim slojevima te aktivacijskim okvirima. Središnji i najveći dio glavnog okvira zauzima podloga za iscrtavanje karte na kojoj su iscrtani rasterski i vektorski slojevi. Rasterski sloj ima ulogu podloge te prikazuje područje Grada Zagreba i pripadajuće okolice. Vektorski sloj prikazuje položaje benzinskih postaja u obliku točaka raznih boja ovisno o vlasniku. Na dnu glavnog okvira nalazi se statusna traka koja služi za prikaz koordinata (izraženih u metrima) kada se mišem pomiče po podlozi za iscrtavanje karte. Pomoću alata na glavnoj alatnoj traci moguće je približavanje, udaljavanje i pomicanje po prikazanim slojevima. Prilikom pokretanja aplikacije automatski se aktivira alat za pomicanje. Pomoću alata za odabir moguće je odabrati željenu benzinsku postaju te o njoj dobiti detaljne informacije. Te informacije se ispisuju na info traci, inače sakrivenoj, koja se prikazuje s lijeve strane glavnog okvira nakon aktivacije alata za odabir (Slika 32.).



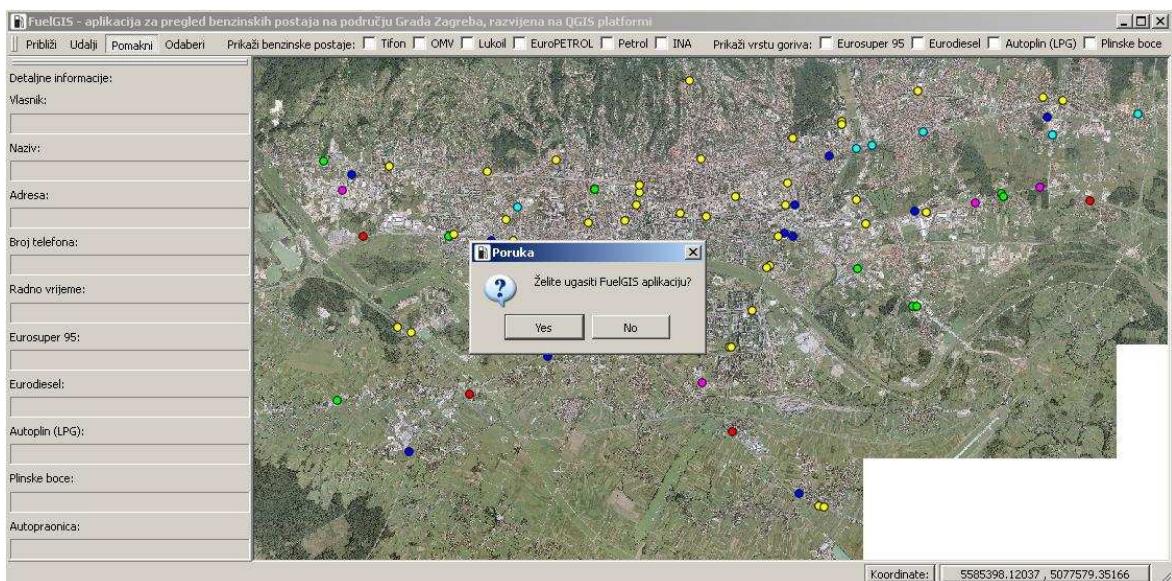
Slika 32. Korištenje alata za odabir

Pritom se odabrana benzinska postaja označava crnom bojom kako bi bila prepoznatljiva u odnosu na okolne benzinske postaje. Aktivacijski okviri služe za prikaz benzinskih postaja prema vlasniku i vrsti goriva. Tako se odabirom odgovarajućih aktivacijskih okvira prikazuje željeni sadržaj. Ako se želi prikazati npr. benzinske postaje naftne kompanije Tifon koje imaju u ponudi autoplin odaberu se odgovarajući aktivacijski okviri te se na podlozi za iscrtavanje karte prikazuju benzinske postaje koje zadovoljavaju navedene uvjete (Slika 33.).



Slika 33. Prikaz željenog sadržaja

Uz to prelaskom miša preko aktivacijskih okvira za prikaz prema vlasniku dobivaju se dodatne informacije, kojom bojom je označena čija benzinska postaja. Na taj način olakšava se snalaženje među različito prikazanim benzinskim postajama. Za završetak rada tj. zatvaranje *FuelGIS* aplikacije potrebno je odabratи gumb za zatvaranje (X) smješten u gornjem desnom uglu glavnog okvira. Pritom se otvara dijaloški okvir u kojem je potrebno potvrditi izlaz iz aplikacije (Slika 34.).



Slika 34. Izlazak iz FuelGIS aplikacije

Odabirom opcije *Yes* dolazi do zatvaranja *FuelGIS* aplikacije te pripadajućeg *Python* interpretera.

7. Zaključak

Ovim radom je prikazan način izrade jednostavne *GIS* aplikacije, s nazivom *FuelGIS*, pomoću tehnologija otvorenog koda. Pokazane su mogućnosti i potencijal velikog broja aplikacija i alata temeljenih na otvorenom kodu. Utvrđeno je da mogu kvalitetno obavljati sve predviđene zadaće. Pritom posebno treba istaknuti *Python* kao programski jezik koji je prilagođen novim korisnicima tj. početnicima u programiranju. Također i *Quantum GIS* je jedna od najboljih *GIS* aplikacija temeljenih na otvorenom kodu. *Python* i *QGIS* su bili temelj na kojem se pristupilo izradi *FuelGIS* aplikacije. Pritom su detaljno opisani svi potrebni koraci, od prikupljanja podataka koji će se koristiti u *GIS* aplikaciji i njihove obrade do izrade aplikacije tj. pisanja koda. U tim postupcima dolazilo je do raznih problema i prepreka koje su uspješno savladane te je kao rezultat toga stvorena jednostavna, ali ipak za izradu zahtjevna aplikacija. Njene mogućnosti pregleda položajnih i atributnih podataka benzinskih postaja na području Grada Zagreba daju korisnicima sve potrebne informacije kako bi im se olakšao odabir benzinske postaje kojom se žele poslužiti. Treba napomenuti kako je ovaj rad među prvima koji istražuju razvoj *GIS* aplikacija na *Quantum GIS* platformi što se tokom razvoja pokazalo kao hindekap jer je bilo poteškoća s pronalaženjem i korištenjem potrebne dokumentacije. No svejedno se uspjelo ispuniti i premašiti početne ciljeve i očekivanja, što dokazuje da se s dobrom idejom te dosta truda može uz minimalna ulaganja napraviti kvalitetan proizvod koji može biti jednak kvalitetan kao i neusporedivo skuplje komercijalne *GIS* aplikacije.

Za kraj, bitno je napomenuti da je moguće neznatnim izmjenama koda ovakvu vrstu aplikacije prilagoditi i nekim drugim namjenama. Također ju je moguće uz malo više truda nadograditi kako bi se omogućile i neke naprednije *GIS* funkcionalnosti poput ažuriranja atributnih podataka, povezivanja putem Interneta s poslužiteljima na kojima se nalaze baze podataka itd. Na temeljima softvera otvorenog koda *FuelGIS* aplikacija ima veliki potencijal za daljnji razvoj, također može poslužiti kao osnova za neke nove projekte.

8. Prilozi

8.1. Prilog 1 – izvorni kod aplikacije

```

1  # -*- coding: utf-8 -*-
2
3  #unos potrebnih modula
4  import sys
5  from sets import Set
6  from qgis.core import QgsRasterLayer, QgsVectorLayer, QgsMapLayerRegistry,
7  QgsApplication, QgsCoordinateReferenceSystem, QgsFeature, QgsGeometry, QgsSymbolV2,
8  QgsRendererCategoryV2, QgsCategorizedSymbolRendererV2
9  from qgis.gui import QgsMapCanvas, QgsMapCanvasLayer, QgsMapToolPan, QgsMapToolZoom,
10 QgsMapToolEmitPoint
11 from PyQt4.QtGui import QAction, QMainWindow, QApplication, QLabel, QFrame, QStatusBar,
12 QCheckBox, QToolBar, QSplitter, QIcon, QMessageBox, QColor
13 from PyQt4.QtCore import SIGNAL, Qt, QString, QObject, QVariant
14
15 #kasnije se koristi za inicijalizaciju aplikacije
16 import os
17 qgis_prefix = os.getenv("QGISHOME")
18
19 class MapCoords(object):
20
21     def __init__(self, mainwindow):
22         selfmainwindow = mainwindow
23
24         #povezivanje canvasa sa signalom pomaka miša kako bi se odredile koordinate
25         QObject.connect(mainwindow.canvas, SIGNAL("xyCoordinates(const QgsPoint&)"),
26         self.updateCoordsDisplay)
27
28         #definiranje polja za prikaz koordinata
29         self.xy = QLabel("0.0 , 0.0")
30         self.xy.setFixedWidth(200)
31         self.xy.setAlignment(Qt.AlignHCenter)
32         self.xy.setStyleSheet(QFrame.StyledPanel)
33
34         #definiranje oznake polja za prikaz koordinata
35         self.name = QLabel("Koordinate:")
36         self.name.setStyleSheet(QFrame.StyledPanel | QFrame.Raised)
37
38         #dodaj oznaku u statusnu traku
39         selfmainwindow.statusBar().addPermanentWidget(self.name)
40
41         #dodaj polje za prikaz u statusnu traku
42         selfmainwindow.statusBar().addPermanentWidget(self.xy)
43
44         #metoda za ažuriranje prikaza koordinata
45         def updateCoordsDisplay(self, p):
46
47             #koordinate miša prikaži kao string
48             capture_string = QString(str(p.x()) + " , " +str(p.y()))
49
50             #koordinate stavi u polje za prikaz
51             self.xy.setText(capture_string)
52
53             #definiranje glavnog prozora
54             class MyWnd(QMainWindow):
55
56                 def __init__(self):
57                     QMainWindow.__init__(self)
58
59                     #postavi naziv glavnog prozora
60                     self.setWindowTitle(u"FuelGIS - aplikacija za pregled benzinskih postaja na
61                     području Grada Zagreba, razvijena na QGIS platformi")
62
63                     #postavi ikonu glavnog prozora
64                     self.setWindowIcon(QIcon("E:/FuelGIS/bp.png"))

```

```

60
61     #definiranje prostora za prikaz slojeva tj. map canvasa
62     self.canvas = QgsMapCanvas()
63     self.canvas.setCanvasColor(Qt.white)
64     self.canvas.show()
65
66     #postavi canvas kao centralni dio prozora
67     self.setCentralWidget(self.canvas)
68
69     #definiranje koordinatnog sustava
70     crs = QgsCoordinateReferenceSystem()
71     crs.createFromProj4("+proj=tmerc +lat 0=0 +lon 0=15 +k=0.9999 +x 0=5500000
+y 0=0 +ellps=bessel +towgs84=514.0188,155.448,507.0461,5.6136,3.676,-
11.4667,0.9999979090043 +units=m +no defs")
72
73     #definiranje pojedinih slojeva i njihovo dodavanje u registar slojeva
74     layer1 = QgsRasterLayer("E:/FuelGIS/Zagreb.tif", "Zagreb")
75     self.canvas.setExtent(layer1.extent())
76     QgsMapLayerRegistry.instance().addMapLayer(layer1)
77     self.layer2 = QgsVectorLayer("E:/FuelGIS/benz.shp", "benzinske", "ogr")
78     QgsMapLayerRegistry.instance().addMapLayer(self.layer2)
79
80     #potrebno dodati slojeve u map canvas da bi se prikazali
81     self.canvas.setLayerSet([QgsMapCanvasLayer(self.layer2),
82                             QgsMapCanvasLayer(layer1)])
82
83     #stvaranje statusne trake
84     sb = QStatusBar()
85     sb.setFixedHeight(22)
86     self.setStatusBar(sb)
87     self.mapcords = MapCoords(self)
88
89     #stvaranje akcije s pripadajućim nazivom
90     self.actionZoomIn = QAction(QString(u"Približi"), self)
91     self.actionZoomOut = QAction(QString("Udalji"), self)
92     self.actionPan = QAction(QString("Pomakni"), self)
93     self.actionSelect = QAction(QString("Odaberi"), self)
94
95     #omogućavanje odabira željene akcije
96     self.actionZoomIn.setCheckable(True)
97     self.actionZoomOut.setCheckable(True)
98     self.actionPan.setCheckable(True)
99     self.actionSelect.setCheckable(True)
100
101    #povezivanje akcije s potrebnim signalom kako bi se aktivirala pripadajuća
metoda
102    self.connect(self.actionZoomIn, SIGNAL("triggered()"), self.zoomIn)
103    self.connect(self.actionZoomOut, SIGNAL("triggered()"), self.zoomOut)
104    self.connect(self.actionPan, SIGNAL("triggered()"), self.pan)
105    self.connect(self.actionSelect, SIGNAL("triggered()"), self.selectActivated)
106
107    #stvori QGIS alat za približavanje i poveži s pripadajućom PyQT akcijom
108    self.toolZoomIn = QgsMapToolZoom(self.canvas, False) # false = in
109    self.toolZoomIn.setAction(self.actionZoomIn)
110
111    #stvori QGIS alat za udaljavanje i poveži s pripadajućom PyQT akcijom
112    self.toolZoomOut = QgsMapToolZoom(self.canvas, True) # true = out
113    self.toolZoomOut.setAction(self.actionZoomOut)
114
115    #stvori QGIS alat za pomicanje i poveži s pripadajućom PyQT akcijom
116    self.toolPan = QgsMapToolPan(self.canvas)
117    self.toolPan.setAction(self.actionPan)
118
119    #stvori QGIS alat koji šalje koordinate prilikom svakog klika na canvasu
120    self.clickTool = QgsMapToolEmitPoint(self.canvas)
121
122    #povezivanje alata za klikanje pomoću potrebnog signala s metodom za odabir
pojedine benzinske postaje
123    QObject.connect(self.clickTool, SIGNAL("canvasClicked(const QgsPoint &,
Qt::MouseButton)"), self.selectFeature)
124
125    #aktivacija QGIS alata za pomicanje prilikom pokretanja aplikacije
126    self.pan()

```

```

127      #stvaranje natpisa za glavnu alatnu traku
128      n1 = QLabel(u"      Prikaži benzinske postaje:  ")
129      n2 = QLabel(u"      Prikaži vrstu goriva:  ")
130
131      #stvaranje checkboxova za prikaz po vlasniku
132      cb1 = QCheckBox("Tifon", self)
133      cb1.stateChanged.connect(self.showTifon)
134      cb2 = QCheckBox("OMV", self)
135      cb2.stateChanged.connect(self.showOMV)
136      cb3 = QCheckBox("Lukoil", self)
137      cb3.stateChanged.connect(self.showLukoil)
138      cb4 = QCheckBox("EuroPETROL", self)
139      cb4.stateChanged.connect(self.showEuroPETROL)
140      cb5 = QCheckBox("Petrol", self)
141      cb5.stateChanged.connect(self.showPetrol)
142      cb6 = QCheckBox("INA", self)
143      cb6.stateChanged.connect(self.showINA)
144
145      #stvaranje checkboxova za prikaz po vrsti goriva
146      cb7 = QCheckBox("Eurosuper 95", self)
147      cb7.stateChanged.connect(self.showSuper95)
148      cb8 = QCheckBox("Eurodiesel", self)
149      cb8.stateChanged.connect(self.showDiesel)
150      cb9 = QCheckBox("Autoplín (LPG)", self)
151      cb9.stateChanged.connect(self.showLPG)
152      cb10 = QCheckBox("Plinske boce", self)
153      cb10.stateChanged.connect(self.showBoce)
154
155      #stvaranje dodatnih informacija za korisnika
156      cb1.setToolTip("<b>Tifon benzinske postaje prikazane su crvenom bojom</b>")
157      cb2.setToolTip("<b>OMV benzinske postaje prikazane su plavom bojom</b>")
158      cb3.setToolTip("<b>Lukoil benzinske postaje prikazane su zelenom bojom</b>")
159      cb4.setToolTip("<b>EuroPETROL benzinske postaje prikazane su cijan bojom</b>")
160      cb5.setToolTip("<b>Petrol benzinske postaje prikazane su magenta bojom</b>")
161      cb6.setToolTip("u<b>INA benzinske postaje prikazane su žutom bojom</b>")
162
163      #stvoren set koji će sadržavati uvjete za prikaz bp po vlasniku
164      self.benz_list = Set()
165
166      #definirane varijable s početnom vrijednosti False budući da su checkboxovi za
167      #gorivo u početku isključeni
168      self.super95 = False
169      self.diesel = False
170      self.plin = False
171      self.boce = False
172
173      #stvaranje glavne alatne trake
174      toolbar = QToolBar()
175      self.addToolBar(toolbar)
176
177      #dodavanje akcija, checkboxova i natpisa u glavnu alatnu traku
178      toolbar.addAction(self.actionZoomIn)
179      toolbar.addAction(self.actionZoomOut)
180      toolbar.addAction(self.actionPan)
181      toolbar.addAction(self.actionSelect)
182      toolbar.addWidget(n1)
183      toolbar.addWidget(cb1)
184      toolbar.addWidget(cb2)
185      toolbar.addWidget(cb3)
186      toolbar.addWidget(cb4)
187      toolbar.addWidget(cb5)
188      toolbar.addWidget(cb6)
189      toolbar.addWidget(n2)
190      toolbar.addWidget(cb7)
191      toolbar.addWidget(cb8)
192      toolbar.addWidget(cb9)
193      toolbar.addWidget(cb10)
194
195      #stvaranje natpisa za info traku
196      info = QLabel("Detaljne informacije:")
197      info.setFixedWidth(230)
198      vlasnik = QLabel("Vlasnik:")

```

```

199     naziv = QLabel("Naziv:")
200     adresa = QLabel("Adresa:")
201     broj_tel = QLabel("Broj telefona:")
202     radno_vrijeme = QLabel("Radno vrijeme:")
203     eurosuper = QLabel("Eurosuper 95:")
204     eurodiesel = QLabel("Eurodiesel:")
205     autoplin = QLabel("Autoplín (LPG):")
206     plinske_boce = QLabel("Plinske boce:")
207     autopraonica = QLabel("Autopraonica:")
208
209     #stvaranje praznih polja na info traci za prikaz info o bp
210     self.vlasnik2 = QLabel("")
211     self.vlasnik2.setFixedHeight(20)
212     self.vlasnik2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
213     self.naziv2 = QLabel("")
214     self.naziv2.setFixedHeight(20)
215     self.naziv2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
216     self.adresa2 = QLabel("")
217     self.adresa2.setFixedHeight(20)
218     self.adresa2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
219     self.broj_tel2 = QLabel("")
220     self.broj_tel2.setFixedHeight(20)
221     self.broj_tel2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
222     self.radno_vrijeme2 = QLabel("")
223     self.radno_vrijeme2.setFixedHeight(20)
224     self.radno_vrijeme2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
225     self.eurosuper2 = QLabel("")
226     self.eurosuper2.setFixedHeight(20)
227     self.eurosuper2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
228     self.eurodiesel2 = QLabel("")
229     self.eurodiesel2.setFixedHeight(20)
230     self.eurodiesel2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
231     self.autoplín2 = QLabel("")
232     self.autoplín2.setFixedHeight(20)
233     self.autoplín2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
234     self.plinske_boce2 = QLabel("")
235     self.plinske_boce2.setFixedHeight(20)
236     self.plinske_boce2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
237     self.autopraonica2 = QLabel("")
238     self.autopraonica2.setFixedHeight(20)
239     self.autopraonica2.setFrameStyle(QFrame.Panel | QFrame.Sunken)
240
241     #stvaranje vertikalnog graničnika
242     splitter = QSplitter(Qt.Vertical)
243
244     #stvaranje info trake
245     self.infoToolbar = QToolBar()
246
247     #staviti info traku s lijeve strane
248     self.addToolBar(Qt.ToolBarArea(Qt.LeftToolBarArea), self.infoToolbar)
249
250     #sakriti info traku
251     self.infoToolbar.setVisible(False)
252
253     #dodavanje graničnika u info traku
254     self.infoToolbar.addWidget(splitter)
255
256     #dodavanje natpisa i pripadajućih polja u graničnik
257     splitter.addWidget(info)
258     splitter.addWidget(vlasnik)
259     splitter.addWidget(self.vlasnik2)
260     splitter.addWidget(naziv)
261     splitter.addWidget(self.naziv2)
262     splitter.addWidget(adresa)
263     splitter.addWidget(self.adresa2)
264     splitter.addWidget(broj_tel)
265     splitter.addWidget(self.broj_tel2)
266     splitter.addWidget(radno_vrijeme)
267     splitter.addWidget(self.radno_vrijeme2)
268     splitter.addWidget(eurosuper)
269     splitter.addWidget(self.eurosuper2)
270     splitter.addWidget(eurodiesel)
271     splitter.addWidget(self.eurodiesel2)

```

```

272     splitter.addWidget(autopl1n)
273     splitter.addWidget(self.autopl1n2)
274     splitter.addWidget(plinske_boce)
275     splitter.addWidget(self.plinske_boce2)
276     splitter.addWidget(autopraonica)
277     splitter.addWidget(self.autopraonica2)
278
279     #stvorena lista koja će sadržavati kategorije za iscrtavanje
280     cat_list = []
281
282     #definiran prvi simbol
283     symbol1_label = QString("Tifon")
284     symbol1_value = QVariant("Tifon")
285     symbol1_color = QColor(Qt.red)
286     symbol1 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
287     symbol1.setSymbol(symbol1_color)
288
289     #definirana prva kategorija za iscrtavanje
290     cat1 = QgsRendererCategoryV2(symbol1_value, symbol1, symbol1_label)
291
292     #prva kategorija dodana na listu za iscrtavanje
293     cat_list.append(cat1)
294
295     #definiran drugi simbol
296     symbol2_label = QString("OMV")
297     symbol2_value = QVariant("OMV")
298     symbol2_color = QColor(Qt.blue)
299     symbol2 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
300     symbol2.setSymbol(symbol2_color)
301
302     #definirana druga kategorija za iscrtavanje
303     cat2 = QgsRendererCategoryV2(symbol2_value, symbol2, symbol2_label)
304
305     #druga kategorija dodana na listu za iscrtavanje
306     cat_list.append(cat2)
307
308     #definiran treći simbol
309     symbol3_label = QString("Lukoil")
310     symbol3_value = QVariant("Lukoil")
311     symbol3_color = QColor(Qt.green)
312     symbol3 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
313     symbol3.setSymbol(symbol3_color)
314
315     #definirana treća kategorija za iscrtavanje
316     cat3 = QgsRendererCategoryV2(symbol3_value, symbol3, symbol3_label)
317
318     #treća kategorija dodana na listu za iscrtavanje
319     cat_list.append(cat3)
320
321     #definiran četvrti simbol
322     symbol4_label = QString("EuroPETROL")
323     symbol4_value = QVariant("EuroPETROL")
324     symbol4_color = QColor(Qt.cyan)
325     symbol4 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
326     symbol4.setSymbol(symbol4_color)
327
328     #definirana četvrta kategorija za iscrtavanje
329     cat4 = QgsRendererCategoryV2(symbol4_value, symbol4, symbol4_label)
330
331     #četvrta kategorija dodana na listu za iscrtavanje
332     cat_list.append(cat4)
333
334     #definiran peti simbol
335     symbol5_label = QString("Petrol")
336     symbol5_value = QVariant("Petrol")
337     symbol5_color = QColor(Qt.magenta)
338     symbol5 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
339     symbol5.setSymbol(symbol5_color)
340
341     #definirana peta kategorija za iscrtavanje
342     cat5 = QgsRendererCategoryV2(symbol5_value, symbol5, symbol5_label)
343
344     #peta kategorija dodana na listu za iscrtavanje

```

```

345     cat list.append(cat5)
346
347     #definiran šesti simbol
348     symbol6_label = QString("INA")
349     symbol6_value = QVariant("INA")
350     symbol6_color = QColor(Qt.yellow)
351     symbol6 = QgsSymbolV2.defaultSymbol(self.layer2.geometryType())
352     symbol6.setColor(symbol6_color)
353
354     #definirana šesta kategorija za iscrtavanje
355     cat6 = QgsRendererCategoryV2(symbol6_value, symbol6, symbol6_label)
356
357     #šesta kategorija dodana na listu za iscrtavanje
358     cat list.append(cat6)
359
360     #definirana vrsta biblioteke za iscrtavanje, iscrtava se po kategorijama
361     renderer = QgsCategorizedSymbolRendererV2('', cat_list)
362
363     #definiran atribut po kojem se iscrtava
364     renderer.setClassAttribute("Vlasnik")
365
366     #vektorskog sloju se pridružuje odabrana biblioteka za iscrtavanje
367     self.layer2.setRendererV2(renderer)
368
369     #metoda za odabir bp i prikaz detaljnih informacija o njoj u info traci
370     def selectFeature(self, point):
371
372         #napravi točku od koordinata koje su poslane od strane kontrole
373         pntGeom = QgsGeometry.fromPoint(point)
374
375         #napravi buffer oko te točke od 5 piksela ovisno o stupnju zooma (ovisno koliko
376         #je stvarnih jedinica u jednom pikselu)
376         pntBuff = pntGeom.buffer( (self.canvas.mapUnitsPerPixel() * 5),0)
377
378         #stvori granični okvir za krug tj. buffer
379         rect = pntBuff.boundingBox()
380
381         #vektorski sloj je pružatelj podataka
382         dp = self.layer2.dataProvider()
383
384         #dohvati sve attribute
385         allAttr = dp.attributeIndexes()
386
387         #definirana lista za bp
388         selectList = []
389
390         #benzinska postaja je klasa QgsFeature
391         bp = QgsFeature()
392
393         #odaberi bp s ovim atributima koristeći granični okvir
394         dp.select([3,4,5,6,7,8,9,10,11,12], rect)
395         while dp.nextFeature(bp):
396
397             #ako se geometrija bp siječe s geometrijom buffera stavi na listu
398             if bp.geometry().intersects(pntBuff):
399                 selectList.append(bp.id())
400
401             #izvuci podatke
402             attrs = bp.attributeMap()
403
404             #prikaži željene podatke
405             vlasnik = attrs[3]
406             self.vlasnik2.setText(vlasnik.toString())
407             naziv = attrs[4]
408             self.naziv2.setText(naziv.toString())
409             adresa = attrs[5]
410             self.adresa2.setText(adresa.toString())
411             broj_tel = attrs[6]
412             self.broj_tel2.setText(broj_tel.toString())
413             radno_vrijeme = attrs[7]
414             self.radno_vrijeme2.setText(radno_vrijeme.toString())
415             eurosuper = attrs[8]
416             self.eurosuper2.setText(eurosuper.toString())

```

```

417         eurodiesel = attrs[9]
418         self.eurodiesel2.setText(eurodiesel.toString())
419         autoplin = attrs[10]
420         self.autoplin2.setText(autoplin.toString())
421         plinske_boce = attrs[11]
422         self.plinske_boce2.setText(plinske boce.toString())
423         autopraonica = attrs[12]
424         self.autopraonica2.setText(autopraonica.toString())
425
426         #čim nađeš prvu točku (bp), završi s petljom
427         break
428
429     #ako ni jedna točka (bp) nije pronađena, počisti prethodne informacije
430     if len(selectList) == 0:
431         self.vlasnik2.clear()
432         self.naziv2.clear()
433         self.adresa2.clear()
434         self.broj_tel2.clear()
435         self.radno_vrijeme2.clear()
436         self.eurosuper2.clear()
437         self.eurodiesel2.clear()
438         self.autoplin2.clear()
439         self.plinske_boce2.clear()
440         self.autopraonica2.clear()
441
442     #odaberi pronađenu točku (bp)
443     self.layer2.setSelectedFeatures(selectList)
444
445     #metode pojedinih QGIS alata za interakciju s canvasom
446     def zoomIn(self):
447
448         #uključi QGIS alat približavanje
449         self.canvas.setMapTool(self.toolZoomIn)
450
451         #isključi gumb "Odaberi"
452         self.actionSelect.setChecked(False)
453
454         #isključi QGIS alat za odabir
455         self.canvas.unsetMapTool(self.clickTool)
456
457     def zoomOut(self):
458
459         #uključi QGIS alat za udaljavanje
460         self.canvas.setMapTool(self.toolZoomOut)
461
462         #isključi gumb "Odaberi"
463         self.actionSelect.setChecked(False)
464
465         #isključi QGIS alat za odabir
466         self.canvas.unsetMapTool(self.clickTool)
467
468     def pan(self):
469
470         #uključi QGIS alat za pomicanje
471         self.canvas.setMapTool(self.toolPan)
472
473         #isključi gumb "Odaberi"
474         self.actionSelect.setChecked(False)
475
476         #isključi QGIS alat za odabir
477         self.canvas.unsetMapTool(self.clickTool)
478
479     #metoda za aktivaciju QGIS alata "Odaberi"
480     def selectActivated(self):
481
482         #otkriwanje pošiljatelja signala
483         button_odaberi = self.sender()
484
485         #ako je uključen gumb "Odaberi"
486         if button_odaberi.isChecked():
487
488             #isključi ostale QGIS alate
489             self.canvas.unsetMapTool(self.toolZoomIn)

```

```

490         self.canvas.unsetMapTool(self.toolZoomOut)
491         self.canvas.unsetMapTool(self.toolPan)
492
493         #uključi QGIS alat za odabir
494         self.canvas.setMapTool(self.clickTool)
495
496         #prikaži info traku
497         self.infoToolbar.setVisible(True)
498
499         #ponovno iscrtavanje canvasa kako bi se prikazala info traka
500         self.canvas.refresh()
501
502     #ovisno o statusu pojedinog checkboxa šalju se uvjeti setu koji koristi metoda za
503     #prikaz
504     def showTifon(self, state):
505
506         #ako je uključen checkbox
507         if state == Qt.Checked:
508             #dodaj uvjet u set
509             self.benz_list.add("Tifon")
510
511         #ako je isključen checkbox
512         else:
513             #ukloni uvjet iz seta
514             self.benz_list.remove("Tifon")
515
516         #pozovi metodu za prikaz
517         self.showbenz()
518
519     def showOMV(self, state):
520
521         #ako je uključen checkbox
522         if state == Qt.Checked:
523             #dodaj uvjet u set
524             self.benz_list.add("OMV")
525
526         #ako je isključen checkbox
527         else:
528             #ukloni uvjet iz seta
529             self.benz_list.remove("OMV")
530
531         #pozovi metodu za prikaz
532         self.showbenz()
533
534     def showLukoil(self, state):
535
536         #ako je uključen checkbox
537         if state == Qt.Checked:
538             #dodaj uvjet u set
539             self.benz_list.add("Lukoil")
540             #pozovi metodu za prikaz
541             self.showbenz()
542
543         #ako je isključen checkbox
544         else:
545             #ukloni uvjet iz seta
546             self.benz_list.remove("Lukoil")
547
548         #pozovi metodu za prikaz
549         self.showbenz()
550
551     def showEuroPETROL(self, state):
552
553         #ako je uključen checkbox
554         if state == Qt.Checked:
555             #dodaj uvjet u set
556             self.benz_list.add("EuroPETROL")
557
558         #ako je isključen checkbox
559         else:
560             #ukloni uvjet iz seta
561             self.benz_list.remove("EuroPETROL")

```

```

562     #pozovi metodu za prikaz
563     self.showbenz()
564
565     def showPetrol(self, state):
566
567         #ako je uključen checkbox
568         if state == Qt.Checked:
569             #dodaj uvjet u set
570             self.benz_list.add("Petrol")
571
572         #ako je isključen checkbox
573         else:
574             #ukloni uvjet iz seta
575             self.benz_list.remove("Petrol")
576
577         #pozovi metodu za prikaz
578         self.showbenz()
579
580     def showINA(self, state):
581
582         #ako je uključen checkbox
583         if state == Qt.Checked:
584             #dodaj uvjet u set
585             self.benz_list.add("INA")
586
587         #ako je isključen checkbox
588         else:
589             #ukloni uvjet iz seta
590             self.benz_list.remove("INA")
591
592         #pozovi metodu za prikaz
593         self.showbenz()
594
595     #ovisno o statusu pojedinog checkboxa šalju se vrijednosti pripadajućih varijabli
596     #metodi za prikaz
597     def showSuper95(self, state):
598
599         #ako je uključen checkbox
600         if state == Qt.Checked:
601             #varijabla ima vrijednost True
602             self.super95 = True
603
604         #ako je isključen checkbox
605         else:
606             #varijabla ima vrijednost False
607             self.super95 = False
608
609         #pozovi metodu za prikaz
610         self.showbenz()
611
612     def showDiesel(self, state):
613
614         #ako je uključen checkbox
615         if state == Qt.Checked:
616             #varijabla ima vrijednost True
617             self.diesel = True
618
619         #ako je isključen checkbox
620         else:
621             #varijabla ima vrijednost False
622             self.diesel = False
623
624         #pozovi metodu za prikaz
625         self.showbenz()
626
627     def showLPG(self, state):
628
629         #ako je uključen checkbox
630         if state == Qt.Checked:
631             #varijabla ima vrijednost True
632             self.plin = True
633
634         #ako je isključen checkbox

```

```

634     else:
635         #varijabla ima vrijednost False
636         self.plin = False
637
638     #pozovi metodu za prikaz
639     self.showbenz()
640
641 def showBoce(self, state):
642
643     #ako je uključen checkbox
644     if state == Qt.Checked:
645         #varijabla ima vrijednost True
646         self.boce = True
647
648     #ako je isključen checkbox
649     else:
650         #varijabla ima vrijednost False
651         self.boce = False
652
653     #pozovi metodu za prikaz
654     self.showbenz()
655
656 #metoda za prikaz bp po vlasniku i vrsti goriva
657 def showbenz(self):
658
659     #upit na početku ne sadrži nikakve uvjete
660     upit = ''
661
662     #ako je duljina seta veća od 0 tj. ako u setu postoji kakav uvjet
663     if len(self.benz_list)>0:
664
665         #definirana lista koja će sadržavati nazine kao vrijednosti atributa
666         bp_lista=[]
667
668         #za svaku bp u setu
669         for benz in self.benz_list:
670
671             #uzmi naziv bp i dodaj " " jer je naziv vrijednost atributa
672             naziv_benz = '"' + benz + '"'
673
674             #dodaj naziv u obliku vrijednosti atributa na listu
675             bp_lista.append(naziv_benz)
676
677         #stavi zarez između svakoga elementa u listi
678         odabrane_bp = ','.join(bp_lista)
679
680         #stvoren upit prema odabranim bp s liste
681         upit = "Vlasnik in (" + odabrane_bp + ")"
682
683     #ako varijabla ima vrijednost True
684     if self.super95 == True:
685
686         #ako već postoji kakav uvjet u upitu
687         if upit != '':
688             #unesi u upit dodatni uvjet
689             upit = upit + ' and "Eurosuper" = "Da"'
690
691         #ako ne postoji nikakav uvjet u upitu
692         else:
693             #unesi u upit novi uvjet
694             upit = upit + '"Eurosuper" = "Da"'
695
696     #ako varijabla ima vrijednost True
697     if self.diesel == True:
698
699         #ako već postoji kakav uvjet u upitu
700         if upit != '':
701             #unesi u upit dodatni uvjet
702             upit = upit + ' and "Eurodiesel" = "Da"'
703
704         #ako ne postoji nikakav uvjet u upitu
705         else:
706             #unesi u upit novi uvjet

```

```

707             upit = upit + '"Eurodiesel" = "Da"'
708
709     #ako varijabla ima vrijednost True
710     if self.plin == True:
711
712         #ako već postoji kakav uvjet u upitu
713         if upit != '':
714             #unesi u upit dodatni uvjet
715             upit = upit + 'and "Autoplín" = "Da"'
716
717         #ako ne postoji nikakav uvjet u upitu
718     else:
719         #unesi u upit novi uvjet
720         upit = upit + '"Autoplín" = "Da"'
721
722     #ako varijabla ima vrijednost True
723     if self.boce == True:
724
725         #ako već postoji kakav uvjet u upitu
726         if upit != '':
727             #unesi u upit dodatni uvjet
728             upit = upit + 'and "Plínske bo" = "Da"'
729
730         #ako ne postoji nikakav uvjet u upitu
731     else:
732         #unesi u upit novi uvjet
733         upit = upit + '"Plínske bo" = "Da"'
734
735     #postavi zadanom sloju upit s odabranim uvjetima
736     self.layer2.setSubsetString(upit)
737
738     #ponovno iscrtavanje slojeva u QgsMapCanvas-u
739     self.canvas.refresh()
740
741     #metoda za izlazak iz aplikacije
742     def closeEvent(self, event):
743
744         #definiranje prozora s porukom te mogućim izborima
745         reply = QMessageBox.question(self, "Poruka",
746             u"Želite ugasiti FuelGIS aplikaciju?", QMessageBox.Yes |
747             QMessageBox.No, QMessageBox.Yes)
748
749         #ako se odabere "Yes" aplikacija se gasi
750         if reply == QMessageBox.Yes:
751             event.accept()
752
753         #ako se odabere "No" aplikacija i dalje ostaje upaljena
754         else:
755             event.ignore()
756
757     #povezivanje aplikacije s PyQt4-om
758     app = QApplication(sys.argv)
759
760     #potrebno je inicijalizirati aplikaciju u QGIS okolini
761     QgsApplication.setPrefixPath(qgis_prefix, True)
762     QgsApplication.initQgis()
763
764     #prikaz glavnog prozora
765     w = MyWnd()
766     w.show()
767
768     #pokreni aplikaciju
769     retval = app.exec_()
770
771     #počisti nakon izvršavanja aplikacije
772     QgsApplication.exitQgis()
773
774     #zatvorи aplikaciju
775     sys.exit(retval)

```

8.2. Sadržaj priloženog optičkog medija (DVD-a)

Na priloženom mediju pohranjeni su podaci korišteni pri izradi diplomskog rada i svi postignuti rezultati. Logički su organizirani prema smislu (Tablica 5).

RB.	Mapa/ Datoteka	Opis sadržaja
1.	Diplomski_rad.doc/pdf	Tekst diplomskog rada
2.	Aplikacija/ FuelGIS.bat	Bat datoteka za pokretanje FuelGIS aplikacije
3.	Aplikacija/ FuelGIS.py	Python skripta, sadrži kod FuelGIS aplikacije
4.	Podaci/ Zagreb.tiff	Digitalni ortofoto Grada Zagreba iz 2006. godine
5.	Podaci/ Zagreb.tfw	Datoteka za georeferenciranje DOF-a Grada Zagreba
6.	Podaci/ Zagreb.tiff.ovr	OVR datoteka, sadrži 8 rezolucija piramide za zadovoljavajuću brzinu navigacije po <i>tiff</i> datoteci.
7.	Podaci/ bp.png	Rasterska datoteka, sadrži ikonu glavnog okvira FuelGIS aplikacije
8.	Podaci/ benz.shp	Shape datoteka, sadrži geometriju benzinskih postaja
9.	Podaci/ benz.shx	Indeks datoteka, sadrži indekse benzinskih postaja koji se koriste prilikom pretraživanja
10.	Podaci/ benz.dbf	Atributna datoteka, sadrži atribute benzinskih postaja
11.	Podaci/ benz.prj	Projekcijska datoteka, sadrži koordinatni sustav i projekcijske parametre za prikaz geometrije
12.	Podaci/ benz.qpj	QGIS projekcijska datoteka s koordinatnim sustavom i projekcijskim parametrima za prikaz geometrije
13.	Podaci/ benz.csv	CSV datoteka, sadrži prikupljene podatke

Tablica 5. Sadržaj priloženog optičkog medija

Literatura:

Miler, M., Odobašić, D. (2010): Baze prostornih podataka – Vježba 1, upute za vježbu 1 iz kolegija Baze prostornih podataka, str. 2-3, Sveučilište u Zagrebu - Geodetski fakultet, Zagreb.

Downey, A., Elkner, J., Meyers, C. (2002): How to Think Like a Computer Scientist – Learning with Python, Green Tea Press, Wellesley, Massachusetts - USA.

POPIS URL-ova:

URL 1. GIS, http://hr.wikipedia.org/wiki/Geografski_informacijski_sustav, (27. 04. 2012.)

URL 2. Google Earth, <http://www.google.com/earth/download/ge/agree.html>,
(29. 03. 2012.)

URL 3. Microsoft Excel, http://hr.wikipedia.org/wiki/Microsoft_Excel, (06. 05. 2012.)

URL 4. CSV, http://en.wikipedia.org/wiki/Comma-separated_values, (06. 05. 2012.)

URL 5. Quantum GIS, http://en.wikipedia.org/wiki/Quantum_GIS, (07. 05. 2012.)

URL 6. Quantum GIS, <http://www.osgeo.org/qgis>, (07. 05. 2012.)

URL 7. GNU Opća javna licenca, <http://dokumentacija.linux.hr/GPL.html>, (07. 05. 2012.)

URL 8. Quantum GIS, <http://hub.qgis.org/projects/quantum-gis/wiki/Download>,
(01. 04. 2012.)

URL 9. Open Source Geospatial Foundation (OSGeo),
http://en.wikipedia.org/wiki/Open_Source_Geospatial_Foundation, (08. 05. 2012.)

URL 10. ESRI Shape, http://en.wikipedia.org/wiki/ESRI_shape, (11. 05. 2012.)

URL 11. JPG, <http://en.wikipedia.org/wiki/JPEG>, (12. 05. 2012.)

URL 12. TIFF, http://en.wikipedia.org/wiki/Tagged_Image_File_Format, (12. 05. 2012.)

URL 13. GIMP, <http://en.wikipedia.org/wiki/GIMP>, (12. 05. 2012.)

URL 14. GIMP, <http://www.gimp.org/downloads>, (02. 04. 2012.)

URL 15. Python, <http://zetcode.com/tutorials/pythontutorial/python>, (03. 04. 2012.)

URL 16. Qt, <http://www.qgisworkshop.org/html/workshop/qt.html>, (03. 04. 2012.)

URL 17. PyQt4, <http://zetcode.com/tutorials/pyqt4/introduction>, (03. 04. 2012.)

URL 18. PyQt4, <http://en.wikipedia.org/wiki/PyQt>, (15. 05. 2012.)

URL 19. Signali i slotovi, <http://doc.qt.nokia.com/4.7/signalsandslots.html>, (05. 04. 2012.)

URL 20. API, <http://hr.wikipedia.org/wiki/API>, (16. 05. 2012.)

URL 21. QGIS API dokumentacija, <http://www.qgis.org/api>, (04. 04. 2012.)

URL 22. Sublime Text 2, <http://www.sublimetext.com/2>, (04. 04. 2012.)

URL 23. HAK-ova interaktivna karta Republike Hrvatske, <http://map.hak.hr>, (20. 03. 2012.)

URL 24. Tifon d.o.o., <http://www.tifon.hr>, (29. 03. 2012.)

URL 25. OMV Hrvatska d.o.o., <http://www.omv.hr>, (29. 03. 2012.)

URL 26. Lukoil Hrvatska d.o.o., <http://www.lukoil.hr>, (30. 03. 2012.)

URL 27. Euro – PETROL d.o.o., <http://www.europetrol.hr>, (30. 03. 2012.)

URL 28. Petrol Hrvatska d.o.o., <http://www.petrol.hr>, (30. 03. 2012.)

URL 29. INA – Industrija nafte d.d., <http://www.ina.hr>, (31. 03. 2012.)

URL 30. Parametri Hrvatskog državnog koordinatnog sustava (HDKS), http://spatial-analyst.net/wiki/index.php?title=MGI / Balkans_coordinate_systems, (01. 04. 2012.)

URL 31. Quantum GIS User Guide - Version 1.7.0 Wroclaw,
[*http://download.osgeo.org/qgis/doc/manual/qgis-1.7.0_user_guide_en.pdf*](http://download.osgeo.org/qgis/doc/manual/qgis-1.7.0_user_guide_en.pdf) (01. 04. 2012.)

URL 32. Prikaz povezanosti Pythona, Qt-a i QGIS-a,
[*http://www.qgisworkshop.org/html/workshop/python_in_qgis_intro.html*](http://www.qgisworkshop.org/html/workshop/python_in_qgis_intro.html), (04. 04. 2012.)

URL 33. Python IDLE, [*http://en.wikipedia.org/wiki/IDLE_%28Python%29*](http://en.wikipedia.org/wiki/IDLE_%28Python%29), (25. 05. 2012.)

URL 34. Bat datoteka, [*http://workingstaff.blogspot.com/2011/02/sto-je-bat-datoteka.html*](http://workingstaff.blogspot.com/2011/02/sto-je-bat-datoteka.html),
(25. 05. 2012.)

Popis slika:

Slika 1. Princip rada signala i slotova (URL 30).....	19
Slika 2. Pisanje koda u Sublime Text 2.....	22
Slika 3. Preuzimanje podataka s HAK-ove interaktivne karte	24
Slika 4. Google Earth – primjer položajnog odstupanja benzinske postaje	25
Slika 5. Unos položajnih i atributnih podataka u CSV datoteku	27
Slika 6. Dijaloški okvir za definiranje koordinatnog sustava	28
Slika 7. Dijaloški okvir za stvaranje sloja iz CSV datoteke	30
Slika 8. Odabir koordinatnog sustava za sloj benz	31
Slika 9. Sloj benz učitan u QGIS-u	31
Slika 10. Stvaranje ESRI Shape datoteke	32
Slika 11. Dodavanje vektorskog sloja u QGIS	33
Slika 12. Prikaz rasterskog i vektorskog sloja u QGIS-u	34
Slika 13. GIMP - postavke za konverziju iz jpg u tiff datoteku	35

Slika 14. Izrada piramide u QGIS-u	36
Slika 15. Prikaz povezanosti Pythona, Qt-a i QGIS-a (URL 32)	39
Slika 16. Početni izbornik OSGeo4W instalacije	41
Slika 17. Izbornik s mogućnošću odabira načina instalacije	41
Slika 18. Izbornik s mogućnošću odabira mjesta i direktorija instalacije	42
Slika 19. Izbornik s mogućnošću odabira mjesta i direktorija za preuzimanje datoteka potrebnih pri instalaciji	43
Slika 20. Izbornik za odabir vrste Internet konekcije	43
Slika 21. Informativni prozor – preuzimanje instalacijskih informacija	44
Slika 22. Izbornik s instalacijskim popisom	45
Slika 23. Informativni prozor – preuzimanje potrebnih datoteka	46
Slika 24. Informativni prozor – instalacija odabranih komponenti	46
Slika 25. Završni izbornik OSGeo4W instalacije	47
Slika 26. Forma za uređivanje sistemskih varijabli	48
Slika 27. Forma za dodavanje korisničkih varijabli	48
Slika 28. Command Prompt prozor	49
Slika 29. Python interpreter u cmd prozoru	50
Slika 30. Elementi sučelja FuelGIS aplikacije	53
Slika 31. Početni izgled FuelGIS aplikacije	77
Slika 32. Korištenje alata za odabir	78
Slika 33. Prikaz željenog sadržaja	79

Slika 34. Izlazak iz FuelGIS aplikacije	79
---	----

Popis tablica:

Tablica 1. Popis obaveznih ESRI Shape datoteka	13
Tablica 2. Popis opcionalnih ESRI Shape datoteka	13
Tablica 3. Popis prikupljenih atributa za svaku benzinsku postaju	26
Tablica 4. Rezultati obrade rasterske datoteke Zagreb	37
Tablica 5. Sadržaj priloženog optičkog medija	93

ŽIVOTOPIS

EUROPEAN
CURRICULUM VITAE
FORMAT



OSOBNE OBAVIJESTI

Ime	KLOBUČAR, NENAD
Adresa	DEMERJE, GORANA ŠIVAKA 53, 10251 HRVATSKI LESKOVAC, REPUBLIKA HRVATSKA
Telefon	091 592 7305
Faks	-
E-pošta	nklobucar@geof.hr , geodet20@net.hr
Državljanstvo	Hrvatsko
Datum rođenja	26. 03. 1986.

RADNO ISKUSTVO

- Datum (od – do) -
- Naziv i sjedište tvrtke zaposlenja -
- Vrsta posla ili područje -
- Zanimanje i položaj koji obnaša -
- Osnovne aktivnosti i odgovornosti -

ŠKOLOVANJE I IZOBRAZBA

- Datum (od – do) rujan 2005 – srpanj 2010
- Naziv i vrsta obrazovne ustanove Geodetski fakultet, Sveučilište u Zagrebu
- Osnovni predmet /zanimanje Preddiplomski studij geodezije i geoinformatike
- Naslov postignut obrazovanjem Sveučilišni prvostupnik inženjer geodezije i geoinformatike
- Stupanj nacionalne kvalifikacije (ako postoji) -

- Datum (od – do) rujan 2001 – lipanj 2005
- Naziv i vrsta obrazovne ustanove I. tehnička škola Tesla
- Osnovni predmet /zanimanje Informatika i računarstvo
- Naslov postignut obrazovanjem Elektrotehničar
- Stupanj nacionalne kvalifikacije (ako postoji) -

• Datum (od – do)	rujan 1993 – lipanj 2001
• Naziv i vrsta obrazovne ustanove	Osnovna škola Brezovica
• Osnovni predmet /zanimanje	-
• Naslov postignut obrazovanjem	-
• Stupanj nacionalne kvalifikacije (ako postoji)	-

OSOBNE VJEŠTINE I SPOSOBNOSTI

Stečene radom/životom, karijerom, a koje nisu potkrnjepljene potvrdama i diplomama.

MATERINSKI JEZIK	HRVATSKI
DRUGI JEZICI	
• sposobnost čitanja	ENGLESKI, NJEMAČKI
• sposobnost pisanja	IZVRSNO
• sposobnost usmenog izražavanja	OSNOVNO
• sposobnost čitanja	IZVRSNO
• sposobnost pisanja	OSNOVNO
• sposobnost usmenog izražavanja	OSNOVNO
ORGANIZACIJSKE VJEŠTINE I SPOSOBNOSTI	-
TEHNIČKE VJEŠTINE I SPOSOBNOSTI <i>S računalima, posebnim vrstama opreme, strojeva, itd.</i>	Napredno znanje o računalima i pratećoj opremi: građa, rad i održavanje Napredno poznавање Windows platformi te Office programskog paketa Programiranje u Python programskom jeziku Iskustvo u radu s komercijalnim i otvorenim GIS alatima: Geomedia, Quantum GIS
UMJETNIČKE VJEŠTINE I SPOSOBNOSTI <i>Glazba, pisanje, dizajn, itd.</i>	-
DRUGE VJEŠTINE I SPOSOBNOSTI <i>Sposobnosti koje nisu gore navedene.</i>	SVJEDODŽBA IZ KOMUNIKACIJSKIH VJEŠTINA – STRUČNO USAVRŠAVANJE
VOZAČKA DOZVOLA	Da, B kategorije
DODATNE OBAVIJESTI	-
DODATCI	-