

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Daniel Presl Radi

**Razvoj geografskega informacijskega sistema s
podporo 3D prikazovalnika**

DIPLOMSKO DELO
NA VISOKOŠOLSLEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Rok Rupnik

Ljubljana, 2011

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00036/2010

Datum: 05.10.2010



Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DANIEL PRESL RADI**

Naslov: **RAZVOJ GEOGRAFSKEGA INFORMACIJSKEGA SISTEMA S
PODPORO 3D PRIKAZOVALNIKA
THE DEVELOPMENT OF GEOGRAPHIC INFORMATION SYSTEM
WITH 3D VIEWER CAPABILITY**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V okviru diplomske naloge razvijte geografski informacijski sistem, kjer bo možno prikazati in urejati že pripravljene geografske podatke in njihove atribute v 3D prostoru. Poleg možnosti prikaza statičnih podatkov naj bo na voljo tudi prikaz dinamičnih podatkov, kot so pozicije iz GPS sistemov in ostalih modulov.

Mentor:

doc. dr. Rok Rupnik

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Daniel Presl Radi,

z vpisno številko 63060329,

sem avtor/-ica diplomskega dela z naslovom:

Razvoj geografskega informacijskega sistema s podporo 3D prikazovalnika

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

doc. dr. Rok Rupnik

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 09.03.2011 Podpis avtorja/-ice: _____

ZAHVALA

Zahvaljujem se profesorju dr. Roku Rupniku za mentorstvo, vodenje in strokovne nasvete pri izdelavi diplomske naloge.

Zahvaljujem se tudi vsem sodelavcem iz podjetja Harpha Sea, ki so me bodrili ter z mano neprestano delili njihovo znanje. Iskreno se zahvaljujem za vso podporo pri sodelovanju na različnih razvojnih projektih skozi čas trajanja študija.

Še posebej pa se zahvaljujem svoji družini in dekletu, ki so mi stali ob strani, ter me neprestano spodbujali in podpirali med študijem in pri izdelavi diplomske naloge.

KAZALO

Povzetek	1
Abstract	2
1. Uvod.....	3
2. Opredelitev problema.....	4
2.1 Predstavitev problema	4
2.2 Cilji novega informacijskega sistema.....	4
2.3 Glavne funkcionalnosti	4
3. Geografski informacijski sistem.....	6
3.1 Kaj je GIS.....	6
3.2 Prostorski podatki.....	7
3.3 Koordinatni sistemi in projekcije	8
4. Uporabljena razvojna orodja in knjižnice	9
4.1 Microsoft Visual Studio 2008	9
4.2 OpenSceneGraph.....	9
4.3 wxWidgets.....	9
4.4 Podatkovna baza.....	10
4.4.1 PostgreSQL	10
4.4.2 PostGIS	10
4.4.3 LibPQ	10
4.5 PHP in CodeIngiter	11
5. Razvoj informacijskega sistema.....	12
5.1 Arhitektura informacijskega sistema.....	12
5.2 Podatkovna baza.....	14
5.3 Kontrolni del	16
5.4 3D prikazovalnik.....	18
5.4.1 Zgradba 3D prikazovalnika	19
5.4.2 Nalaganje objektov in risanje slojev	19
5.4.3 Niti risanja in posodabljanja.....	21
5.4.4 Dinamični objekti.....	22
5.4.5 Razvijanje za prihodnost	23
6. Sklepne ugotovitve in nadaljnji razvoj.....	25
Seznam slik	26
Seznam tabel	26
Viri in literatura.....	27

Seznam uporabljenih kratic in simbolov

3D	tridimenzionalen
API	(angl. <i>Application Programming Interface</i>) - aplikacijski programski vmesnik
GIS	(angl. <i>Geographic(al) Information System</i>) - geografski informacijski sistem
GPS	(angl. <i>Global Positioning System</i>) - globalni pozicijski sistem
HTTP	(angl. <i>Hypertext Markup Language</i>) – transportni protokol za hiperbesedilo
IDE	(angl. <i>Integrated Development Environment</i>) - integrirano razvojno okolje
OGC	(angl. <i>Open Geospatial Consortium</i>) – organizacija, ki določa standarde na GIS področju
OpenGL	(angl. <i>Open Graphic Library</i>) – nizko nivojska grafična knjižnica
OSG	(angl. <i>Open Scene Graph</i>) – preprosta grafična knjižnica, ki temelji na OpenGL-ju
PHP	(angl. <i>Hypertext Preprocessor</i>) - programski jezik za razvoj spletnih strani
SRID	(angl. <i>Spatial Reference Identifier</i>) - oznaka prostorskega koordinatnega sistema
SQL	(angl. <i>Structured Query Language</i>) - strukturiran povpraševalni jezik za delo s podatkovno bazo
SUPB	sistem za upravljanje s podatkovno bazo

Povzetek

Vizualizacija in analize prostorskih podatkov predstavljajo pomemben vir informacij, ki so v pomoč pri razumevanju prostorskih pojavov ter vplivajo na sprejemanje prostorskih odločitev. Za učinkovito pridobivanje informacij iz prostorskih podatkov je danes na razpolago mnogo različnih programskih rešitev.

Cilj diplomske naloge je vzpostaviti geografski informacijski sistem, kjer bo možno prikazati in urejati že pripravljene geografske podatke in njihove attribute v 3D prostoru, ki so hranjeni v PostgreSQL podatkovni bazi. Poleg možnosti prikaza statičnih podatkov, bo na voljo tudi prikaz dinamičnih podatkov, kot so pozicije iz GPS sistemov in različnih modulov.

V diplomski nalogi sem najprej na kratko predstavil in opisal pojem geografski informacijski sistem in pomen prostorskih podatkov, kasneje pa opisal še orodja in tehnologije, ki smo jih uporabili pri izdelavi. Diplomsko delo opisuje tako predstavitev, cilje in funkcionalnost sistema, kot tudi probleme in ugotovitve pri samem razvoju informacijskega sistema.

Sistem je funkcionalno razdeljen na dva dela – kontrolni del, s katerim prikazujemo in spreminjamo attribute prostorskih podatkov preko spletnih strani, ter 3D prikazovalnik, v katerem prikazujemo prostorske podatke v 3D prostoru, ter se sprehajamo po terenu.

Ključne besede:

- geografski informacijski sistem
- vizualizacija prostorskih podatkov
- PostgreSQL
- 3D prikazovalnik

Abstract

Visualization and analyses of spatial data represents important source of information, which are very helpful in understanding spatial occurrences and which affects accepting of spatial decisions. For effective extraction of information from spatial data, many different software solutions are available today.

The goal of this bachelor thesis is to establish a geographic information system, where will be possible to show and arrange already prepared geographical data and their attributes in 3D space, which are stored in PostgreSQL database. Beside possibility of displaying static data, we will be possible to display dynamic data, such as the position of GPS systems and other different modules.

In this bachelor thesis I briefly introduced and described the concept of geographic informational system and the importance and meaning of spatial data. Then I described tools and technologies that we used in making this system. This thesis describes presentation, goals and functionalities of the system, as well as problems and conclusions in development of information system.

The system functionality is divided in two parts – the control part, which allow us to show and change attributes of spatial data through web pages. The second part is the viewer in which we present spatial data or spatial information in 3D space.

Keywords:

- geographical information system
- spatial data visualization
- PostgreSQL
- 3D viewer

1. Uvod

V današnjem času imamo na razpolago ogromno podatkov, iz katerih pridobivamo informacije. Vendar je nekatere podatke lažje predstaviti numerično, druge pa vizualno. Dandanes je vizualni prikaz podatkov neprimerno bolj učinkovit kot numerična predstavitev.

To velja predvsem za geografske prostorske podatke, saj le ti vsebujejo geometrične attribute, kot so oblika, velikost in seveda lega na površju. Ravno zato postaja poudarek na vizualni predstavitvi vedno večji, omogoča pa boljše preglednost ter kvalitetnejšo razumevanje informacije. Za vizualizacijo geografskih prostorskih podatkov obstaja mnogo različnih in zelo uporabnih programskih orodij, s katerimi lahko poleg prikaza, še urejamo, analiziramo in zajemamo podatke. Takim orodjem pravimo geografski informacijski sistem.

Kot se dogaja vsem znanostim, se tudi geografija vse bolj povezuje in prepleta z informatiko. Z razvojem informacijske tehnologije se posledično razvijajo tudi geografski informacijski sistemi, ki postajajo izredno pomembni pri sprejemanju številnih odločitev in spremljanju dejavnosti v prostoru.

V podjetju Harpha Sea d.o.o., iz Kopra, pri katerem delam, smo razvili geografski informacijski sistem, ki ima trenutno nekaj manj funkcij, je pa zato bolj specializiran za posebne uporabnike, kot so civilna zaščita, gasilska brigada in mestna občina, saj je razvit po naročilu. Glavna posebnost tega geografskega informacijskega sistema je v tem, da so vsi podatki prikazani v 3D prostoru, s katerim dobimo lepšo in boljše predstavitev, kot če bi podatke vizualizirali v 2D prostoru.

Temelj sistema je podatkovna baza PostgreSQL skupaj s dodatkom za geografske podatke PostGIS, in odprtokodna knjižnica OpenSceneGraph, ki temelji na odprtokodni knjižnici OpenGL. Celoten sistem je napisan v programskem jeziku C++, sestavljen pa je iz dveh posebej ločenih delov, ki sta povezana med sabo samo s podatkovno bazo. Prvi, kontrolni del nam omogoča ogled in spreminjanje atributov prostorskih podatkov, izbiranje slojev in vnašanje novih podatkov v bazo preko protokola HTTP in spletnega jezika PHP. Drugi, prikazovalni del, pa nam omogoča sprehajanje po prostoru ter spremljanje in analiziranje vizualnih podatkov. Celoten prikazovalnik je napisan s pomočjo knjižnice OpenSceneGraph, njegova dobra lastnost pa je v tem, da zna prikazati tudi dinamične objekte. To smo izkoristili v naš prid in v prikazovalniku omogočili spremljanje in spreminjanje objektov (na primer vozila, ljudje, veter itd.) v realnem času.

Sistem je delujoč, a za uporabno delovanje potrebuje že vnaprej pripravljene prostorske podatke v podatkovni bazi, ter sam teren, ki ga naloži kot statičen objekt ob zagonu sistema. Sistem je zasnovan v taki obliki, da je delo z njim zelo preprosto. Uporabnik oziroma administrator mora imeti le znanje za delo s podatkovnimi bazami PostgreSQL, predvsem pa s prožilci, saj se večina poizvedb upravlja z njimi.

V diplomski nalogi si bomo podrobneje ogledali razvoj sistema, arhitekturo podatkovne baze ter uporabljene metode, orodja in tehnologije. Najprej bomo opredelili problemsko področje, zahteve in cilje novega informacijskega sistema, nato pa bomo obdelali malo teoretičnih osnov, saj bomo predstavili pomen geografskega informacijskega sistema in prostorskih podatkov.

2. Opredelitev problema

2.1 Predstavitev problema

Ker se tehnologija in strojna oprema razvijata in obstaja programska oprema, ki zna prikazati geografske podatke na zelo privlačen in interaktiven način, smo se v podjetju Harpha Sea d.o.o odločili, da bomo razvili geografski informacijski sistem, ki sledi tem razvojnim trendom. Trenutno najbolj mamljiva in zanimiva tehnologija je prikaz stvari v 3D tehniki, vendar je strojna opreme šele v razvoju in zato tudi precej draga.

Odločili smo se, da je za začetek potrebno najprej razviti geografski informacijski sistem, ki zna prikazati geografske prostorske podatke v 3D prostoru. Sistem mora delovati hitro, da je lahko konkurenčen drugim rešitvam na trgu, ter mora na enostaven in učinkovit način obvladati ogromno količino prostorskih podatkov. Poleg tega mora biti geografski informacijski sistem prenosen, da se lahko zaganja na različnih računalnikih, nočemo pa, da so podatki podvojeni in med seboj različni. Zato se morajo podatki hraniti na skupni podatkovni bazi ter biti neprestano ažurirani.

Kot GIS sistem mora ta omogočati tudi spremembo atributov prostorskih podatkov, zato je bilo treba razviti tudi kontrolni del sistema, s katerim lahko zelo hitro in spretno spreminjamo podatke v podatkovni bazi. Odločili smo se, da bi bilo za ta del najlažje uporabiti spletni jezik PHP, saj mora pri taki veliki količini podatkov prikazovati attribute dinamično.

Ker je informacijski sistem razvit po naročilu, so hoteli naročniki nekaj posebnega, nekaj kar v drugih GIS sistemih ni. Želeli so nekatere podatke prikazovati dinamično, ter jim tako spreminjat pozicijo, velikost ali barvo.

2.2 Cilji novega informacijskega sistema

Glavni cilj novega informacijskega sistema je priprava celotne, enostavne podatkovne baze ter informacijskega sistema, ki se napaja s podatki iz te podatkovne baze. Podatki se bodo nahajali na enem mestu, zagnanih pa bo več instanc informacijskega sistema. Omogočiti je treba, da so vsi podatki ažurni, dostopni in celoviti.

S standardizacijo podatkovne baze želimo uporabnikom na različnih področjih omogočiti zelo malo spreminjanja podatkovne baze, samo delovanje GIS sistema pa optimizirati ter tako pospešiti dolgotrajne in nepotrebne procese.

Z novim informacijskim sistemom želimo prikazovati in urejati tako statične prostorske podatke kot dinamične objekte. Zato je treba tehtno premisliti in določiti, kateri objekti bodo hranili zgodovino in kateri ne. Objekte bodo lahko tako izbirali in vnašali kar preko 3D pregledovalnika kot preko dinamičnih spletnih strani.

2.3 Glavne funkcionalnosti

Glavne funkcije novega informacijskega sistema, ki smo si jih zastavili:

- povezava na katerokoli že v naprej ustvarjeno in pripravljeno podatkovno bazo,
- izbira koordinatnega sistema in projekcije,
- prikaz prostorskih podatkov v 3D prostoru v obliki slojev,
- izpis vklopljenih in izklopljenih slojev,
- prikaz in izris dinamičnih podatkov,

- možnost sledenja dinamičnih objektov,
- prikaz in hranjenje zgodovine podatkov,
- izbiranje objektov preko 3D prikazovalnika in preko spletnih strani,
- izpis atributov prostorskih podatkov v dinamičnih spletnih strani in
- vnašanje novih atributov preko spletnih strani.

Poleg zgoraj omenjenih, glavnih funkcionalnosti, smo si zadali nalogo, da bo naš nov geografski informacijski sistem interaktiven in, da bo vseboval vsaj nekaj lastnosti, ki jih imajo sistemi za virtualno resničnost.

3. Geografski informacijski sistem

Geografski informacijski sistemi predstavljajo celovito rešitev za zajem, prikazovanje in distribucijo prostorskih podatkov. Izvajanje prostorskih analiz je največja moč prostorskih informacijskih sistemov. Nudijo nam neizčrpen vir izvedenih informacij, s pomočjo katerih lahko izvajamo dela na operativnem področju ali pa so v pomoč strateškim odločitvam na nivoju vodstva [1].

3.1 Kaj je GIS

Geografski informacijski sistem, krajše GIS (angl. *Geographic(al) Information System*), je kompleksen pojem, za katerega se je v svetu vzporedno razvijalo več definicij.

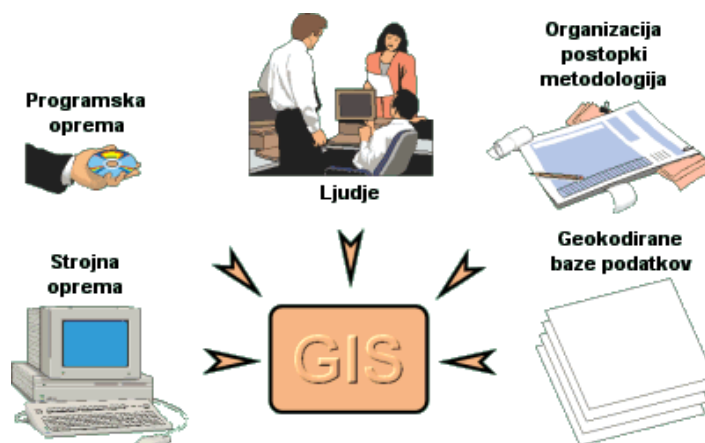
Obstajajo tri definicije tega pomena:

- tehnološka,
- procesno funkcijska in
- vsebinska definicija.

Po tehnološki definiciji je geografski informacijski sistem opredeljen kot zbirka tehnoloških orodij za zbiranje, shranjevanje, iskanje, pretvorbe in prikazovanje prostorskih podatkov za določene namene. Geografski informacijski sistem je sinonim za ustrezno računalniško tehnologijo, metodologijo in orodja, ki so nastala in se razvila z integracijo tehnologije podatkovnih baz ter računalniško podprte kartografije.

Procesno funkcijska definicija poudarja pomen podatkovnih analiz in opredeljuje GIS kot računalniško podprt prostorski informacijski sistem, ki nudi upravljalno, organizacijsko in poslovno osnovo za zajemanje, shranjevanje, iskanje, obdelavo, analiziranje, prikazovanje ter distribucijo prostorskih podatkov.

Vsebinska definicija se osredotoča na sestavne komponente GIS sistema, ki so naslednje: splošna in posebna strojna oprema, sistemska in posebna programska oprema, sistem uporabniških aplikacij, integrirana podatkovna baza prostorskih podatkov ter vzdrževalci in uporabniki informacijskega sistema.



Slika 1: Osnovni elementi, ki tvorijo GIS sistem.

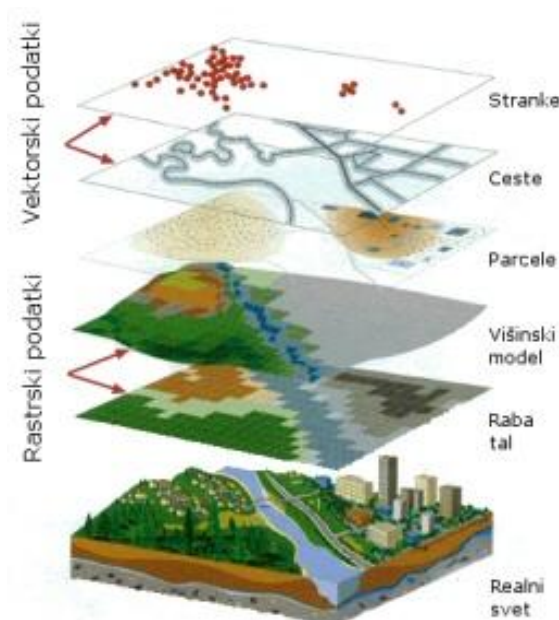
Iz prejšnjih treh definicij lahko zaključimo, da je GIS učinkovito orodje za upravljanje s prostorskimi podatki ter izvajanje prostorskih analiz. Geografski informacijski sistemi združujejo lastnosti geografskih informacijskih sistemov, lahko pa tudi tehničnih digitalnih arhivov in sistemov za upravljanje z dokumenti ter tako predstavljajo integrirano rešitev na

področju upravljanja tehničnih sredstev in tako uporabniku nudijo kakovostno informacijsko podporo pri pridobivanju različnih informacij o tehničnih sredstvih, infrastrukturnih sistemih, nepremičninah, ipd. Naloga GIS je biti v kar največji meri v pomoč uporabniku pri izvajanju vsakodnevnih operativnih nalog kot tudi pri strateških razvojnih odločitvah. Geografski informacijski sistemi so v geografsko lokacijske namene še zlasti uporabni pri:

- projektiranju novih sistemov,
- intervencijskih posegih,
- komercialnih predstavitev (lokacija prostorov, dimenzije, okolje),
- sprejemanju strateških razvojnih odločitev,
- in še na mnogih drugih področjih.

3.2 Prostorski podatki

Geografski informacijski sistem vsebuje prostorske podatke, ki jih imenujemo sloji. Ti so opredeljeni z geometričnimi in alfanumeričnimi atributi. Med geometrične attribute sodijo oblika, lega in velikost, alfanumerični atributi pa so njihove pripadajoče lastnosti. Prostorski podatki so lahko grafično organizirani v vektorski ali rastrski obliki [2].



Slika 2: Razporeditev vektorskih in rastrskih podatkov.

Vektorska organizacija temelji na treh osnovnih grafičnih gradnikih: točkah, linijah in poligonih. Grafični gradniki so opredeljeni s točkami v koordinatnem sistemu in povezavami med njimi. Na prostorsko opredeljene grafične gradnike se navezujejo ostali opisni ali atributni podatki.

Rastrske podatke si lahko predstavljamo kot sliko. Glavna razlika med rastrskimi podatki in navadnimi slikami je v tem, da so rastrski podatki georeferencirani. Bistveno za rastrske podatke je, da jih lahko zelo učinkovito in nazorno prikažemo grafično.

3.3 Koordinatni sistemi in projekcije

Geografski informacijski sistem uporablja podatkovno bazo PostgreSQL z dodatkom PostGIS. PostGIS baza sledi OGC standardom, vendar jih tudi razširja ter predlaga nove.

Po OGC standardu obstaja v bazi enolična oznaka (ključ) prostorskega koordinatnega sistema. To enolično oznako imenujemo SRID (angl. *Spatial Reference Identifier*), ki določa projekcijo prostorskih podatkov [3]. Paziti moramo, da izberemo oziroma določimo pravilno SRID oznako, ker lahko v nasprotnem primeru pride do nezaželenih premikov v koordinatnem prostoru.

Številka 4326 je na primer oznaka za koordinatni sistem zemljepisnih koordinat z WGS84 projekcijo, njegova širina in dolžina sta podana v geografskih stopinjah in minutah. Številka 3794 pa je oznaka za koordinatni sistem z ETRS89 projekcijo, ki pa ima podano širino in dolžino v metrih. SRID oznaka 0 pa pomeni, da koordinatni sistem oziroma projekcija ni podana, kar lahko povzroča zelo velike napake.

Trenutno obstajajo OGC standardi le za dvodimenzionalne prostorske podatke, vendar PostGIS podpira tudi tridimenzionalne (poleg širine in dolžine, podpira tudi višino).

Pri našem sistemu bomo upoštevali le dvodimenzionalne prostorske podatke – le širino in dolžino, višino pa bomo razbirali iz višin terena.

4. Uporabljena razvojna orodja in knjižnice

Za razvoj našega geografskega informacijskega sistema smo izbrali samo takšna razvojna orodja in tehnologije, ki so prosto dostopna na spletu, ter so v večini primerov platformno neodvisna. Edina posebnost je spodaj omenjeni Microsoft Visual Studio, ki je plačljiv ter deluje samo na operacijskih sistemih Microsoft Windows. Vendar tudi zanj obstaja zamenjava, kot je na primer Eclipse.

4.1 Microsoft Visual Studio 2008

Microsoft Visual Studio je integrirano razvojno okolje (angl. *Integrated development environment - IDE*), ki ga je razvilo podjetje Microsoft. To je aplikacija, ki vsebuje orodja za razvoj in razhroščevanje programske opreme. Omogoča tako razvoj konzolnih, kot tudi grafičnih uporabniških vmesnikov ter spletne strani, spletne aplikacije ter spletne storitve. Programska oprema, razvita s pomočjo Visual Studia, teče samo na Microsoftovih operacijskih sistemih. S pomočjo Visual Studia smo oblikovali celotno aplikacijo, ki je napisana v programskem jeziku C++. Za razvoj aplikacij smo poleg knjižnic, ki jih uporablja Visual Studio, uporabili še spodaj naštetje knjižnice [4].

4.2 OpenSceneGraph

OpenSceneGraph je odprtokodno grafično orodje za razvoj visoko zmogljivih grafičnih aplikacij, kot so simulatorji letenja, igre, virtualne resničnosti in znanstvene vizualizacije. Temelji na konceptu iz SceneGraph, ki zagotavlja predmetno usmerjeno ogrodje na temeljih OpenGL-ja. To omogoča, da razvijalcu ni potrebno izvajanje in optimizacija nizko stopenjskih klicev grafičnemu okolju, ter ponuja veliko dodatnih pripomočkov za hiter razvoj grafičnih aplikacij [5].

Napisan je v celoti v standardnem C++ in OpenGL jeziku, ter tako zagotavlja razvoj knjižnic, ki so popolnoma osredotočene le na potrebe končnih uporabnikov. Ključne prednosti OpenSceneGraph-a so učinkovitost, razširljivost, prenosljivost in produktivnost pri razvoju programske opreme.



Slika 3: Logotip OpenSceneGraph.

4.3 wxWidgets

wxWidgets je knjižnica za izdelavo in oblikovanje grafičnih uporabniških vmesnikov, namenjena pa je razvoju aplikacij, ki so neodvisne od platforme. wxWidgets omogoča pisanje, vodenje in zagon aplikacij na različnih platformah z minimalno ali brez spremembe kode. Zajema sisteme, kot so Microsoft Windows, Mac OS X, Linux in druge [6].

4.4 Podatkovna baza

Ciljni SUPB, na katerem je postavljena GIS baza je PostgreSQL, ki omogoča PostGIS funkcionalnosti, te pa so potrebne za delo z geografskimi in geometrijskimi podatki.

4.4.1 PostgreSQL

PostgreSQL je objektno-relacijski upravljalnik podatkovnih baz, razvit na kalifornijski univerzi na oddelku za računalništvo Berkeley (angl. *Berkeley Computer Science Department*). PostgreSQL je prosta in odprtokodna programska oprema in kot je to pri mnogih odprtokodnih sistemih - jo razvijajo razne globalne skupnosti razvijalcev in podjetij [7].

Podpira velik del standardnega jezika SQL ter ponuja številne funkcije:

- kompleksne poizvedbe,
- tuje ključe,
- sprožilce,
- poglede in
- transakcijsko integriteto.

Da bi PostgreSQL konkuriral svojim tekmecem, ima vgrajeno tudi proceduralno razširitev SQL-a. Podobno kot ima Oracle proceduralni jezik PL/SQL, ima PostgreSQL vgrajen jezik PL/pgSQL. Proceduralni jezik omogoča pisanje kode v blokih, ki se izvedejo na strežniški strani, lahko pa se uporablja tudi za ustvarjanje uporabniško določenih funkcij (kot so poizvedbe in prožilci). S standardno namestitvijo PostgreSQL-a so poleg proceduralnega jezika PL/pgSQL vključeni še proceduralni jeziki PL/Tcl, PL/Perl in PL/Python. Nestandardni jeziki, ki so lahko kasneje nameščeni pa so PL/Lua, PL/php, PL/Ruby, PL/sh, PL/Java in drugi.

4.4.2 PostGIS

PostGIS je odprto-kodna programska oprema, ki dodaja podporo geografskim objektom v PostgreSQL SUPB-ju. V praksi to pomeni, da lahko geografske podatke (kot so točke, linije, poligoni itd.) hranimo v podatkovni bazi kot posebne objekte (angl. *geometry*). Prednost je v tem, da lahko te objekte enostavno prikazujemo na interaktivnih zemljevidih in jih lažje obdelujemo [8].

Namestitev PostGIS je zelo enostavna. To se stori tako, da se obstoječi ali novo kreirani podatkovni bazi v PostgreSQL-ju doda notranje funkcije PostGIS-a. S tem postopkom postane PostgreSQL baza tudi PostGIS, ker lahko na njej uporabljamo vse funkcije, ki jih PostGIS omogoča.

4.4.3 LibPQ

LibPQ je primarna knjižnica, ki smo jo uporabili kot vmesnik med podatkovno bazo PostgreSQL in našo aplikacijo. Omogoča izvajanje SQL stavkov direktno na strežniku, brez omejevanja aplikacije.

4.5 PHP in CodeIgniter

PHP je razširjen odprtokodni programski jezik, ki se uporablja za strežniške uporabe oziroma za razvoj dinamičnih spletnih vsebin. Lahko ga primerjamo z Microsoftovim sistemom ASP, VBScript in JScript, Sunovim sistemom JSP in Java ter sistemom CGI in Perl.

Podoben je običajno strukturiranim programskim jezikom kot sta jezikoma C in Perl, najbolj izkušenim programerjem pa dovoljuje razvijanje zapletenih uporab brez dolgega učenja. Glavni namen pri PHP-ju je izgradnja dinamične spletne strani. Torej, da se stran odziva na uporabnikove vnose podatkov.

CodeIgniter je orodje za ljudi, ki gradijo spletne strani v PHP-ju. Temelji na MVC (Model-View-Controller) arhitekturi in je skladna s standardom deljenega gostovanja računov, različnimi verzijami PHP-ja (vključno s PHP4 in PHP5) in konfiguracijami. Njegov cilj je omogočiti, da se projekti razvijajo veliko hitreje, kot bi lahko, če bi pisali kodo iz nič. Zagotavlja bogat nabor knjižnic za najpogostejša opravila, preprost vmesnik ter logično strukturo za dostop do njih. Z minimizacijo količine kode omogoča da se ustvarjalno osredotočimo na projekt [9].

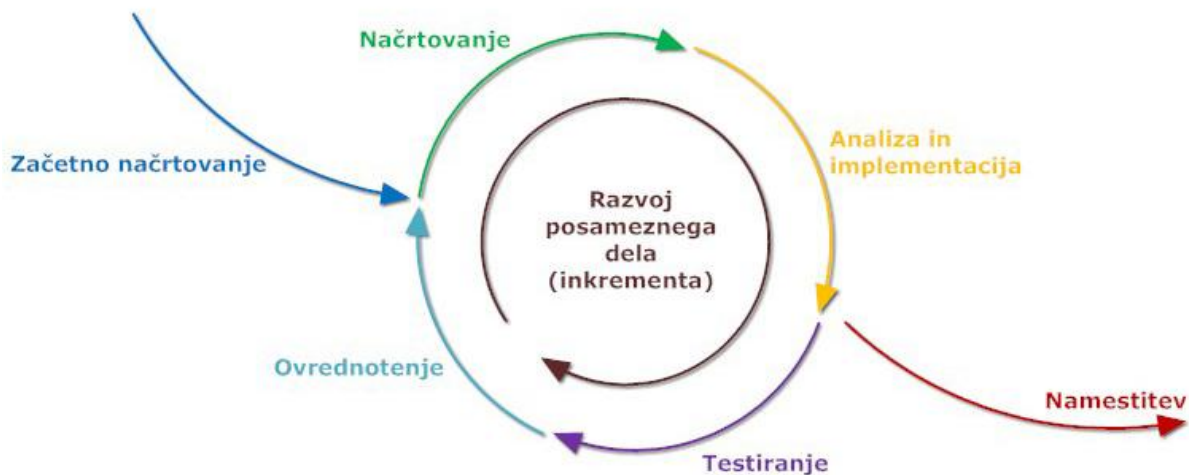
Večji del dela kot je nalaganje knjižnic, pridobivanje podatkov iz modela in sestavljanje pogleda je opravljeno znotraj kontrolerja. Ni potrebno poimenovati modele po tabelah v bazi. Deluje z vsemi strežniki in spletnimi gostovanji, ki zagotavljajo PHP 4.3.2 ali novejšo različico. Za večino spletnih aplikacij zahteva podatkovno bazo. Trenutno so podprte podatkovne baze: MySQL (4.1 +), MySQLi, MS SQL, Postgres, Oracle, SQLite, in ODBC.

5. Razvoj informacijskega sistema

Razvoj informacijskega sistema je potekal v več ponavljajočih se ciklih (iteracijah). Vsak cikel je vseboval manjše faze (inkremente), te pa so doprinesle korenite spremembe tako v podatkovni bazi kot v sami aplikaciji. Za izdelavo našega informacijskega sistema smo uporabili kombinacijo iterativnega in inkrementalnega razvoja. S to strategijo smo si omogočili, da imamo lahko sistem v uporabi, čeprav še ni popolnoma funkcionalen, ter vsako budočo izboljšavo oziroma dopolnitev lahko preprosto vključimo [10].

Iterativni razvoj pomeni odlašanje izvedbe za določen čas, da bi hitreje napredovali in se kasneje vračali k istemu problemu, kjer bi dosežene rezultate izboljšali oziroma popravili. Tak pristop je težje obvladljiv, saj je pri planiranju težko oceniti koliko iteracij bo potrebno za doseg zastavljenega cilja.

Inkrementalni razvoj je strategija napredovanja v majhnih korakih, da bi prišli do ustreznih rezultatov. Problematiko moramo najprej razdeliti, ki jih nato rešujemo vzporedno ali izmenično. Ko posamezen podproblem rešimo, ga testiramo in vključimo v obstoječi sistem.



Slika 4: Shema iterativno-inkrementalnega razvoja.

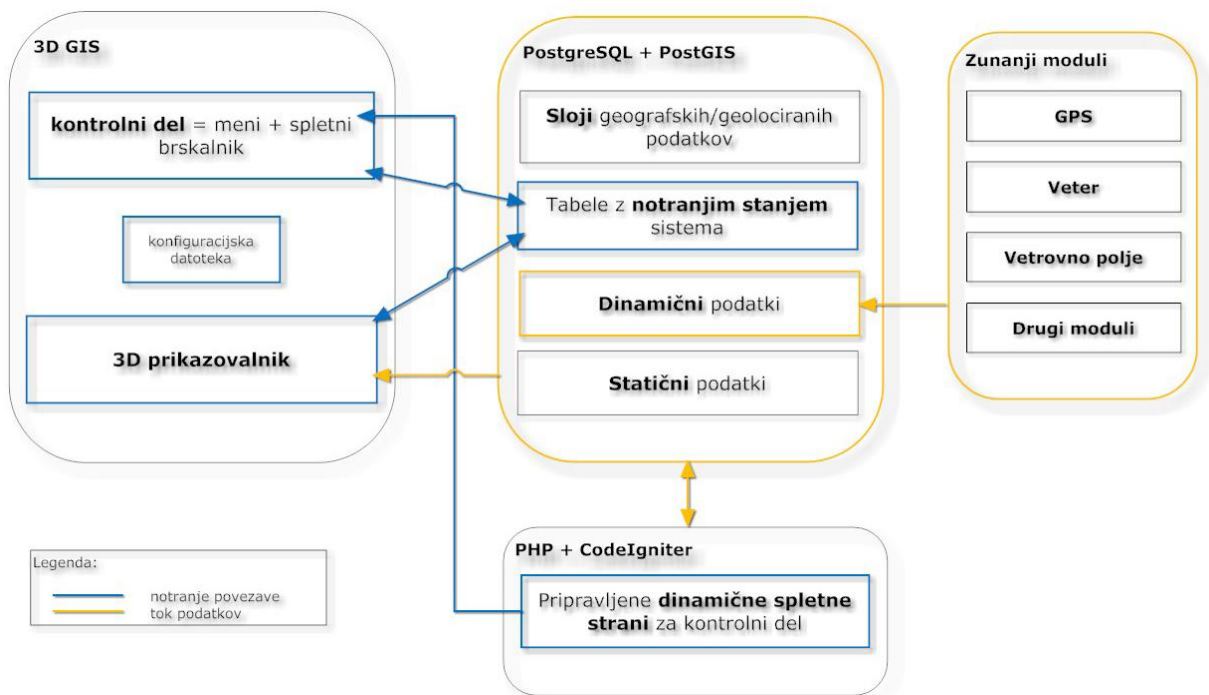
5.1 Arhitektura informacijskega sistema

Razvoj informacijskega sistema smo razdelili na tri dele, tako bodo tudi opisni v nadaljevanju diplomske naloge. Vendar smo vse dele razvijali vzporedno, saj so vsi med sabo tesno povezani ter odvisni drug od drugega.

V prvem delu je bilo treba razviti temelj informacijskega sistema, to je podatkovna baza, ki ne sme biti specializirana, mora pa se poljubno širiti glede na število prostorskih podatkov.

V drugem delu smo razvijali kontrolni del, s katerim je bilo mogoče spreminjati in pregledovati atribute in podatke na zelo hiter način ter dinamično. Za ta del smo si zamislili kar spletni brskalnik in spletne strani, ki so napisane s programskim jezikom PHP in JavaScript.

V zadnjem, tretjem delu smo pripravili prikazovalni del (v nadaljevanju 3D prikazovalnik), ki mora znati prikazovati različne prostorske podatke ter različne dinamične objekte. Seveda pa mora imeti vse lastnosti, kot jih imajo vsi ostali GIS sistemi.



Slika 5: Arhitektura in tok podatkov GIS sistema.

Kontrolni del in 3D prikazovalnik za svoj zagon potrebuje konfiguracijsko datoteko. Z njo določimo parametre podatkovne baze, ki so nujni za povezavo med njo in programi. Poleg nastavitve podatkovne baze, se v njej nahajajo drugi parametri, kot so SRID oznaka projekcije geometrijskih podatkov, način zagona 3D prikazovalnika, ime terena, začetna pozicija kamere, proxy nastavitve ter opis levega in desnega klika miške. Kot posebnost v konfiguracijski datoteki, so zapisani tudi statični gumbi. Njihov pomen, bomo razložili v nadaljevanju diplomske naloge.

Primer konfiguracijske datoteke:

```
[Program config]
Name=SeaSovzo
Type=1                #1 .. celozaslonski način, 2 .. ptičji pogled
Terrain=teren.ive    #ime terena
StartPos=401015.0;45148.0;3.0    #začetna pozicija kamere
SRIDout=3794        #SRID oznaka izhodnih podatkov

[Server config] #nastavitve povezave s strežnikom
Host=localhost
Port=5432
Database=testna_baza
Username=user
Password=user
HTTP=localhost:8010    #spletni naslov dinamičnih strani
```

```
[Proxy settings] #proxy nastavitve
Proxy=proxy.test.si
ProxyPort=80
NoProxiesOn=localhost, 127.0.0.1 #strani, na katere proxy ne vpliva

[Mouse config] #nastavitve gumbov miške (opis;tip_objekta)
LeftDoubleClick=Izpis atributov na tocki klika;objekti
RightDoubleClick=Izbira lokacije za novo intervencijo;intervencija

[Static buttons config] #seznam statičnih gumbov (opis;tip_dogodka)
Static_0=3D
Static_1=3D - vozila;vozilo
Static_2=3D - osebe;oseba
```

5.2 Podatkovna baza

Temelj celotnega informacijskega sistema je njegova podatkovna baza in podatki shranjeni v njej. Ciljni SUPB na katerem je postavljena GIS podatkovna baza je PostgreSQL, ki omogoča PostGIS funkcije, te pa so potrebne za delo z geometrijskimi podatki.

Začetno načrtovanje podatkovne baze je bila eno od najtežjih nalog samega razvoja informacijskega sistema. Ustvariti smo morali podatkovno bazo, ki jo je moč enostavno širiti z novimi prostorskimi podatki, mora pa biti enostavna, hitra in tako zastavljena, da bi ustrezala različnim uporabam našega sistema. Poleg tega smo morali načrtovali podatkovno bazo tako, da bi bilo v njej shranjeno začetno stanje - tako kontrolnega dela kot 3D prikazovalnika.

Osnovna podatkovna baza je sestavljena iz desetih tabel, ki so ključni za samo delovanje informacijskega sistema. Vsaka pa ima svojo funkcijo:

- **events**

V tabelo *events* se shranjujejo podatki, ko uporabnik dva-krat klikne v 3D prikazovalniku. V tistem trenutku se zapišejo podatki o času, ko je uporabnik kliknil na teren, pozicija na katero je kliknil (koordinate x, y in z), vrsta klika (levi ali desni gumb miške) ter na kateri sloj je kliknil. V tej tabeli se podatki samo dodajajo, ker želimo hraniti zgodovino klikov.

- **events_types**

Ta tabela je šifrant za tabelo *events*. V njej so zapisani vsi možni tipi dogodkov, ki se lahko zgodijo ob dvojnem kliku na teren v 3D prikazovalniku.

- **geometry_columns**

Geometry_column je interna tabela PostGIS dodatka ter je obvezna za normalno delovanje podatkovne baze. Vsakič ko v podatkovno bazo dodamo nov sloj, se v tabelo *geometry_column* samodejno zapišejo podatki o shemi, v kateri je sloj, ime tabele, ime stolpca z geometrijo, SRID oznaka ter tip geometrije.

- **dyn_models**

To je šifrant, v katerem so zapisane poti do 3D modelov za vsak tip pozicij posebej.

- **dynamic_objects**

Ta tabela, je osrednja tabela za prikaz dinamičnih objektov. V njej hranimo pozicijo objekta, dodatno višino, smer, v katero je objekt obrnjen, hitrost premika, povečavo, opis ki se izriše v 3D prikazovalniku, višina po našem digitalnem reliefu, enolična oznaka id, ter čas, ko je bil objekt nazadnje osvežen.

- **layers**

Tabela imenovana *layers* predstavlja šifrant vseh možnih slojev, ki jih lahko prikažemo. V njej se hrani enolična identifikacijska oznaka, ime, barva, velikost ter ime tabele

vsakega sloja posebej. V njej je zapisana tudi pot sloja, ki ga naložimo lokalno, ter imena atributov, ki jih bomo izpisali.

- **spatial_ref_sys**

To je še ena interna tabela PostGIS dodatka. V njej so shranjene različne projekcije, s katerimi lahko določimo koordinatni sistem geometrijskih prostorskih podatkov.

- **application_state**

Kot že samo ime tabele pove, se v njej hrani stanje celotne aplikacije. Hrani se identifikacijska številka sistema, s katero brskamo po ostalih tabelah, tip programa, podatek, ali spremljamo kakšen dinamičen objekt ali ne, številko projekcije (SRID) geometrijskih podatkov, kam naj se obrne kamera ter kateri naslov naj se odpre v kontrolnem delu.

- **windows_state**

V tabeli *windows_state* se hrani začetno in trenutno stanje povezav oziroma oken, ki jih lahko kontrolni del odpre. V njej se hrani tako pozicija okna, kot njena velikost, URL naslov ter ime, ki se izpiše na zaslonu.

- **layers_state**

Zadnja a nič manj vredna tabela se imenuje *layers_state*. V njej so shranjene vse spremembe pri spreminjanju sloja za prikaz. Da ne bi prišlo do napak, moramo vedeti, da spremembe, ki se zgodijo pri spreminjanju atributov oziroma pozicij geometrij, se v tej tabeli ne hranijo. Hrani se samo sprememba barve in velikost sloja, ter podatek, ali je sloj sploh vklopljen, izklopljen ali pa prozoren.

Naša podatkovna baza ima nekaj za podatkovne baze zelo nenavadnih lastnosti. Ena od teh je, da večina sprememb (tako vnosi, kot brisanje in spreminjanje), ki se zgodijo nad podatki, spremljajo številni sprožilci, ki vplivajo na druge entitete glede na različne lastnosti vnosov. Primer dveh tabel, v katere vnašamo podatke le s pomočjo prožilcev so naslednje:

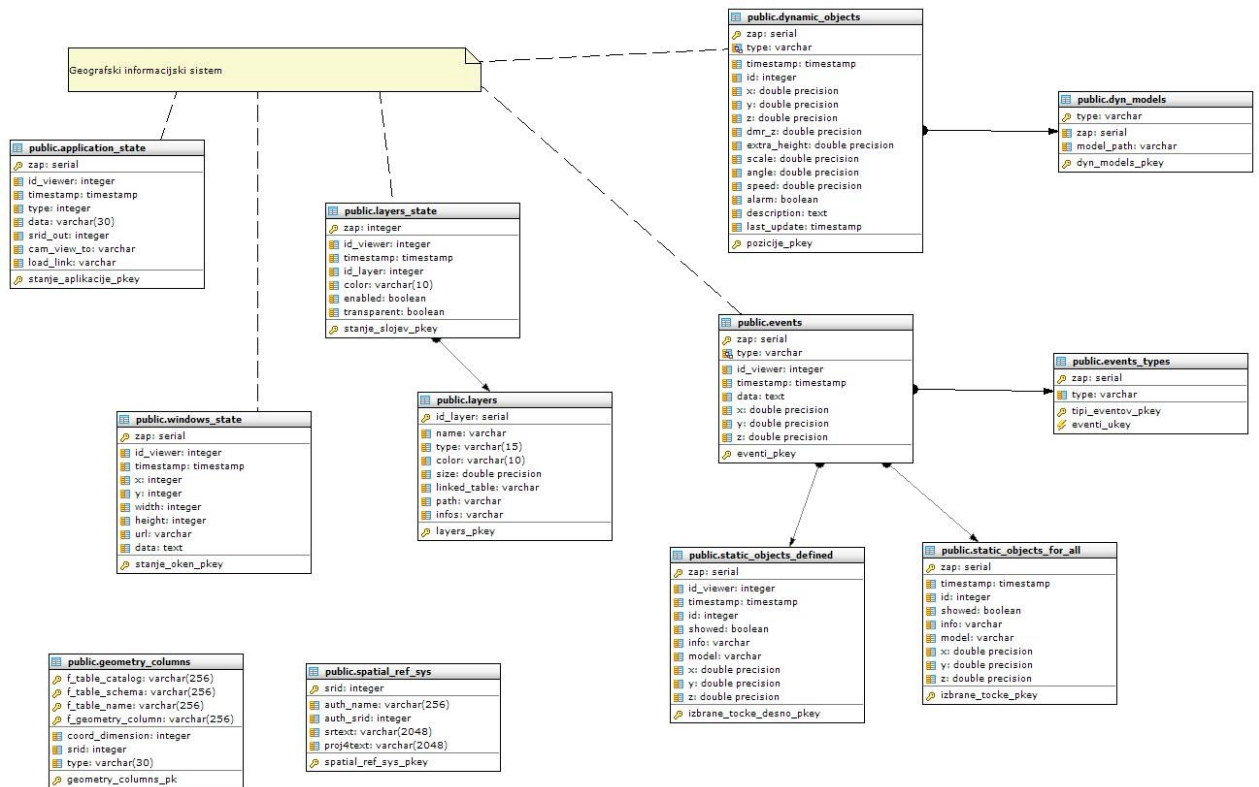
- **static_objects_defined**

V tabelo *static_objects_defined* ne vpisujemo podatke direktno iz same aplikacije, vendar se podatki s pomočjo prožilcev vpisujejo oziroma posodablajo preko tabele *events*. Tabela je namenjena podatkom ob dogodku, ko uporabnik klikne na enega od gumbov miške (to določimo s prožilcem). Podatke preusmerjamo v to tabelo, ker ne želimo hraniti njihove zgodovine.

- **static_objects_for_all**

V nasprotnem primeru, ko uporabnik klikne z drugim gumbom miške, se podatki kopirajo v tabelo *static_objects_for_all*. Razlika med zgornjima dvema tabelama je v tem, da se v tabelo *static_objects_for_all* shranjuje zgodovina vpisov; vsi podatki, ki se vpišejo, so neodvisni od identifikacijske številke sistema. Podatki se izrišejo v vseh zagnanih instancah informacijskega sistema.

Druga posebna lastnost je predvsem ta, da tabele znotraj podatkovne baze večinoma niso povezane med seboj, vendar so vse povezane z našo aplikacijo. Izjeme so le šifranti. V našem primeru je zato toliko težje narisati kakršenkoli logični oziroma konceptualni model.



Slika 6: Shema podatkovne baze.

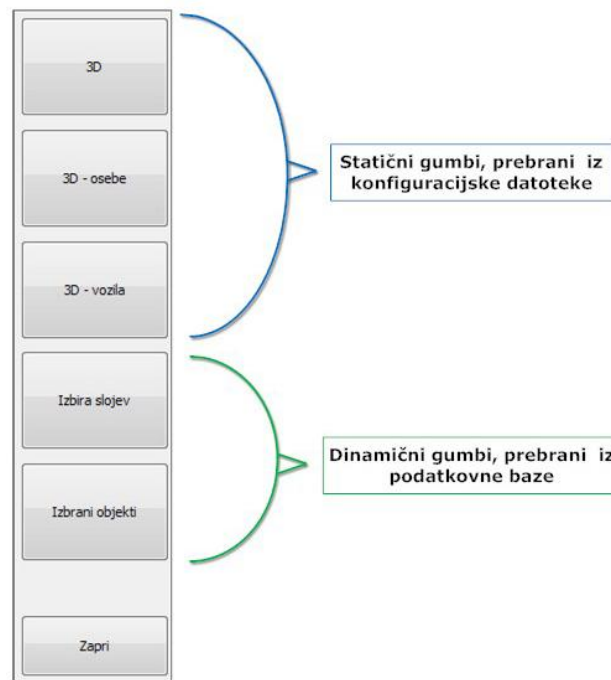
Kombinacija orodji PostgreSQL in PostGIS, se je v nadaljevanju izkazala kot odlična rešitev za razvoj podatkovnega modela. Glavni razlog tiči že v zasnovi PostgreSQL, ki v primerjavi s podobnimi komercialnimi rešitvami enakovreden, v nekaterih primerih pa celo prekaša konkurente. Poleg tega obstaja vrsta orodij, s katerimi lahko upravljamo podatkovno bazo in manipuliramo s podatki (na primer phpPgAdmin, pgAdmin III, Quantum GIS, itd.). Vsa orodja so platformno neodvisna in na voljo brezplačno.

Glavna prednost sistema za upravljanje s podatkovnimi bazami PostgreSQL (PostGIS) pred drugimi sistemi, ki prav tako nudijo podporo shranjevanja geografskih podatkov lahko v nekaj točkah sklenemo tako:

- je brezplačen in odprtokoden,
- nudi odlično komercialno poslovno podporo,
- slovi po stabilnosti in zanesljivosti,
- je prilagodljiv in razširljiv,
- je platformno neodvisen,
- je razvit za okolja z veliko količino podatkov in
- je enostaven za upravljanje.

5.3 Kontrolni del

S pomočjo kontrolnega dela pregledujemo, urejemo ter dodajamo nove attribute izbranim geometrijskim objektom. Z njim spreminjamo tudi delovanje 3D prikazovalnika, izris slojev ter navigacijo po terenu. Razdeljen je na dva dela in sicer na meni ter spletni brskalnik. Napisan je v programskem jeziku C++ s pomočjo knjižnice wxWidgets, ker lahko z njo enostavno in hitro oblikujemo vmesnik.

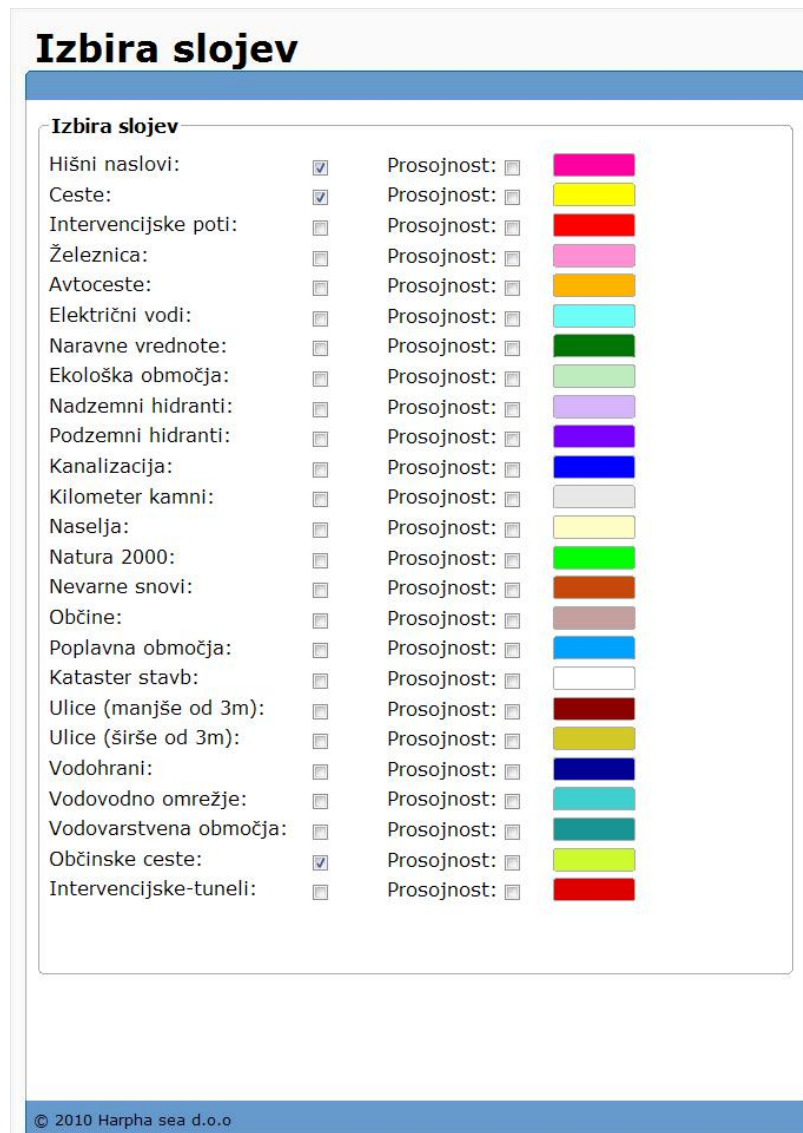


Slika 7: Primer kontrolnega dela – meni.

Meni je zgrajen samo iz enostavnih velikih gumbov, ki stojijo na levem delu zaslona, ne moremo mu spreminjati ne velikosti ne pozicije. Število in zaporedje gumbov je odvisno od podatkovne baze ter od konfiguracijske datoteke.

Kot že omenjeno: v konfiguracijski datoteki so zapisani tudi statični gumbi, ki spreminjajo samo način delovanja 3D prikazovalnika. Z njimi se ne odpre spletni brskalnik, klik na njih vpliva na tabelo *application_state* in način navigacije. Na primer s klikom na gumb 3D-vozilo, spremenimo delovanje navigacije v 3D prikazovalniku tako, da sledi samo objektom, ki so tipa vozilo. Število in postavitev dinamičnih gumbov pa sistem pridobi iz tabele *windows_state*. Ko kliknemo na enega od dinamičnih gumbov se nam odpre vgrajen spletni brskalnik, pot spletne strani pa pridobi iz podatkovne baze.

S spletnim brskalnikom prikazujemo že vnaprej pripravljene dinamične spletne strani. Nima nobene orodne vrstice za brskanje, ki omogočajo vnašanje spletnih naslovov, saj te dobi izključno iz podatkovne baze. Spletne strani so napisane v programskem jeziku PHP, ter s pomočjo JavaScripta in CodeIngiterja. Spletni brskalnik ne bi bil dovršen, če ne bi omogočal proxy nastavitve, ki so tudi zapisane v konfiguracijski datoteki.



Slika 8: Spletni brskalnik s že pripravljeno spletno stranjo za izbiranje slojev.

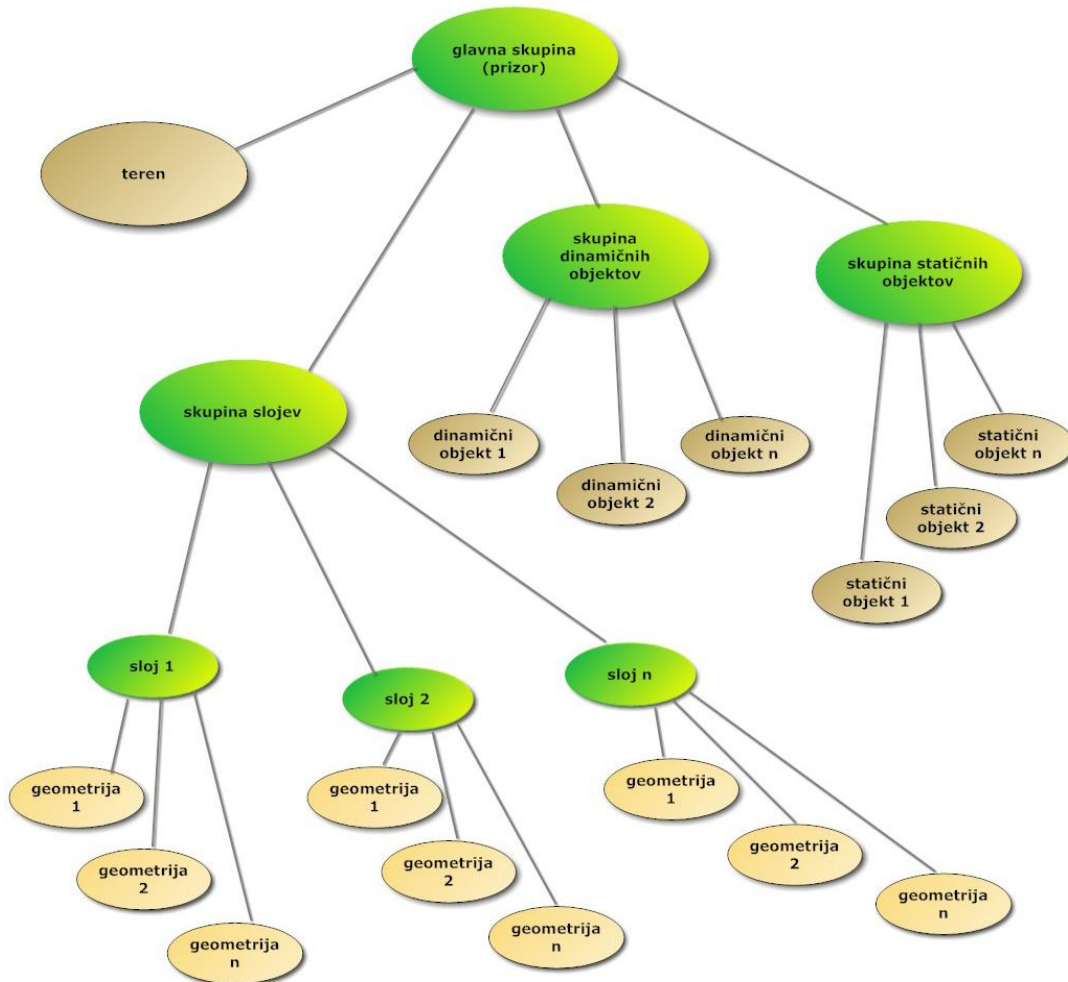
Razlog, da smo izbrali spletne strani za spreminjanje ter dodajanje atributov je bil zelo enostaven. Želeli smo ustvariti dinamični vmesnik, ki ga je enostavno spreminjati in dopolnjevati ter še lažje upravljati. Poleg tega pa spletne strani delujejo na vsakem računalniku, prikaz podatkov je hiter in dinamičen. Držali smo se načela »napiši manj, naredi več« (angl. »Write less, do more«) [11].

5.4 3D prikazovalnik

Če je podatkovna baza temelj celotnega informacijskega sistema, je 3D prikazovalnik osrednji in najpomembnejši del našega sistema. Preko njega želimo prikazati prostorske podatke na zanimiv in interaktiven način. Razvoj je tekel hitro in enostavno, saj je knjižnica OpenSceneGraph, s katero je razvit 3D prikazovalnik, zelo učinkovita, za naše potrebe pa dovolj izpopolnjena ter lahka za uporabo.

5.4.1 Zgradba 3D prikazovalnika

Zgradba prikazovalnika je zelo podobna razvejanemu drevesu. Na vrhu je glavno vozlišče oziroma skupina, ki se določi kot glavni prizor, nanj pa se obešajo skupine objektov ali pa kar posamezni objekti. Celotno hierarhijo, ne glede na to, ali vsebuje samo statične ali tudi dinamične objekte, smo pripeli na glavni pogon prikazovalnika, ki omogoča celotno delovanje [12].



Slika 9: Zgradba 3D prikazovalnika.

5.4.2 Nalaganje objektov in risanje slojev

Knjižnica OpenSceneGraph že sama posebej vključuje velik nabor dodatnih modulov, s katerimi smo si olajšali razvoj sistema. Eden od teh modulov se imenuje *osgDB*, ki nam omogoča enostavno in preprosto nalaganje objektov v 3D prikazovalniku. Modul *osgDB* se uporablja za nalaganje in shranjevanje objektov ter tako nudi podporo za širok nabor 2D in 3D datotečnih formatov.

Opis	Končnica	Modul zna brat	Modul zna pisat
3DC point cloud reader	.3dc, .asc	da	ne
3D studio	.3ds	da	da
Quake3 BSP	.bsp	da	ne

COLLADA 1.4.x	.dae	da	da
Autodesk DXF Reader	.dxf	da	ne
Autodesk FBX	.fbx	da	da
Open Inventor format	.iv, .wrl	da	da
Native OSG binary	.ive	da	da
Logo Database	.logo	da	ne
Lightwave Objects	.lwo, .lw, .geo	da	ne
Lightwave Scene	.lws	da	ne
Quake MD2	.md2	da	ne
Alias Wavefront	.obj	da	da
Multigen Openflight	.flt	da	da
Native OSG ascii	.osg	da	da
ESRI Shapefile	.shp	da	ne
Stereolithography file	.stl, .sta	da	ne
VRML2	.wrl	da	ne
Window bitmap	.bmp	da	da
DirectX image format	.dss	da	da
GIF Image	.gif	da	ne
Jpeg Image	.jpeg, .jpg	da	da
PIC image	.pic	da	ne
Portable Network Graphics	.png	da	da
TIFF image	.tiff, .tiff	da	da

Tabela 1: Nekateri podprti tipi datotečnih formatov v OSG.

Ta modul smo uporabili tudi za nalaganje terena ter ostalih dinamičnih 3D objektov. Enostavnost in preprostost njegovega delovanja pokaže že samo število vrstic programske kode:

```
int main( int, char** )
{
    osgViewer::Viewer viewer;

    //branje terena iz lokalne datoteke
    osg::Node* teren = osgDB::readNodeFile( 'teren.ive' );

    viewer.setSceneData( teren );
    return viewer.run();
}
```

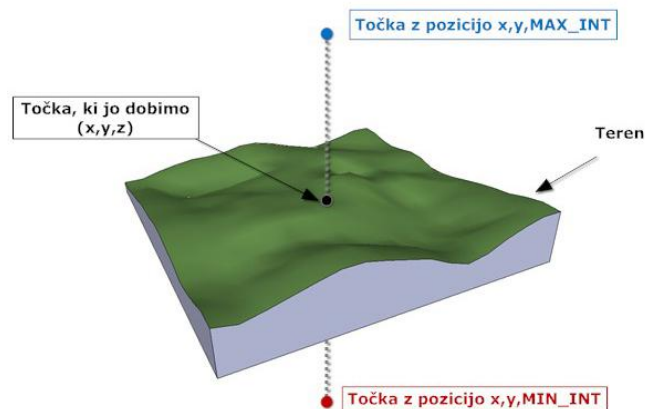
Nalaganje slojev iz baze pa je čisto druga zgodba. Vsi geometrijski prostorski podatki so v podatkovni bazi shranjeni v binarni obliki, niso pa primerni za nalaganje v našem 3D prikazovalniku. Zato smo morali vsakega posebej narisat - in ne naložit.

S pomočjo različnih, že vgrajenih funkcij v PostGIS dodatku, smo vse podatke iz baze prejeli v obliki teksta, ter jih razčlenili s pomočjo razčlenjevalnika (angl *parser*), ki smo ga razvili. Koordinate smo nato shranjevali v geometrije, te pa izrisali na površju.



Slika 10: Izris slojev na površju.

Do problema je prišlo, ko smo se zavedli, da večina prostorskih podatkov v bazi vsebuje le dve koordinati (x in y), višine pa niso podane. Problem smo rešili tako, da smo višine pobirali kar iz višin terena. V dodatnem modulu *osgUtil*, znotraj knjižnice OpenSceneGraph, obstaja razred (angl. *Intersector*), ki vrača koordinate, ko se dva objekta sekata. Tako smo ustvarili izpeljan razred, ki skozi dano točko potegne premico, s to seka teren, vrne pa točko sečišča. Iskanje presečišča je zelo zamudna in obremenjujoča funkcija, glede na velikost podatkov pa se izvaja samo pri nalaganju slojev.



Slika 11: Grafični prikaz delovanja *Intersectorja*.

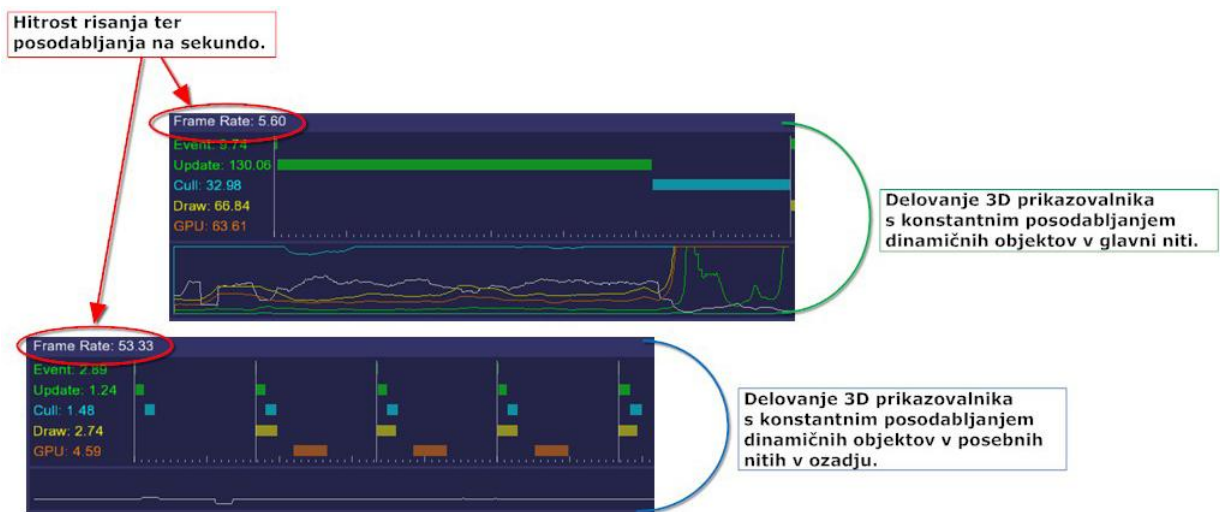
5.4.3 Niti risanja in posodabljanja

Čeprav zelo napredna in izpopolnjena knjižnica, ima OpenSceneGraph še veliko pomanjkljivost. Za spreminjanje in izrisovanje objektov na zaslon, prečka vse objekte v hierarhiji ter pregleda spremembe in dogodke. To pa se dogaja tako pogosto, kot se osvežuje monitor računalnika, v povprečju se to zgodi 60 krat na sekundo. Da bi bilo še težje, vse to opravlja v glavni in edini niti, ta pa je razvijalcem skrita. Da bi informacijski sistem opravljal vse funkcije, ki smo si jih zadali, bi moral dinamične objekte posodabljati v realnem času.

Med razvijanjem je v tem delu prišlo do težko rešljivega problema. Ko smo hoteli posodobiti vse objekte v 3D prostoru, se je prikazovalnik zelo upočasnil ter tako postal neuporaben. Razlog za upočasnitev je bila konstantna komunikacija z podatkovno bazo, ki je omejevala čas izrisa. Prišli smo do zaključka, da moremo uvesti druge, posebne niti. Tako smo poleg glavne uvedli še dodatne niti, v katerih komuniciramo s podatkovno bazo ter posodabljammo objekte.

S pomočjo knjižnice Boost smo ustvarili tri dodatne niti. V vsaki se posodablja za njih pripravljeni objekti:

- nit za posodabljanje slojev,
- nit za posodabljanje izbranih točk ter stanja aplikacije in
- nit za posodabljanje dinamičnih objektov.



Slika 12: Primerjava hitrosti risanja v eni niti ter v več nitih.

S tako strukturo smo dosegli, da komunikacija med sistemom in podatkovno bazo ne ovira risanja in hitrosti 3D prikazovalnika. Hitrost je vidna tudi iz zgornje slike.

5.4.4 Dinamični objekti

V 3D prikazovalniku lahko prikazujemo tudi dinamične objekte. To so objekti, ki se konstantno spreminjajo, njihove podatke pa pridobivajo iz zunanjih modulov, kot so GPS sistemi, vetrovni sistemi ter drugi. Dinamičnim objektom lahko spreminjamo pozicijo, jih povečujemo, določamo hitrost premika, obračamo v poljubne smeri ter spreminjamo napis in barvo napisa, ki se nahaja nad njimi. Spreminjaje samega objekta ni mogoče, saj za našo uporabo to ne pride v poštev.

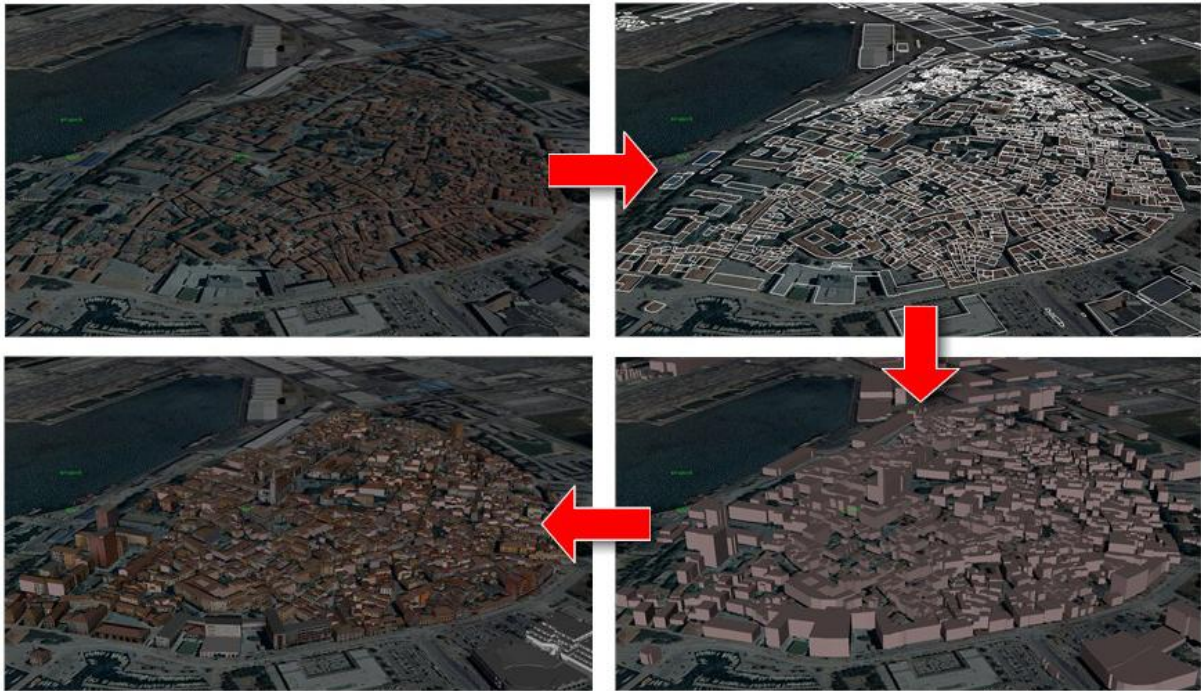


Slika 13: Primer dinamičnih objektov – oseba in vozilo.

5.4.5 Razvijanje za prihodnost

Da bi držali obljubo ter izpolnili zadano nalogo, smo poskušali narediti še korak naprej pri interaktivni predstavitvi prostorskih podatkov. Po dolgotrajnem in temeljitem spoznavanju orodja OpenSceneGraph, smo razvili poseben program izven sistema, s katerim lahko izvozimo podatke iz podatkovne baze. Te pa nato v drugačni obliki uvozili v naš informacijski sistem, kot lokalni sloj. To smo storili predvsem, da bi 3D prikazovalnik pridobil navdih virtualnega sveta.

Iz podatkovne baze smo vzeli podatke katastra stavb in njihove višine ter jih izvozili v obliki kubov. Da bi še bolj popestrili vizualno predstavitev, smo na stranice prilepili fotografije fasad, uredili obliko streh, bolj razgibane stavbe pa celo samo modelirali ter vstavili v sloj. Omejitev pri izdelavi takega posebnega sloja je predvsem v tem, da izdelava še ni avtomatsko vključena v sam informacijski sistem ter, da za pripravo potrebujemo veliko časa in potrpljenja.



Slika 14: Razvoj 3D prikazovalnika.

Pri razvoju in spoznavanju orodij smo naleteli še na eno zelo zanimivo lastnost, ki lahko dopolni naš geografski informacijski sistem. Ugotovili smo namreč, da tudi v skupnosti OpenSceneGraph stremijo v prihodnost pri prikazu v 3D tehnologiji. OpenSceneGraph lahko prikaže 3D prostor tudi v različnih stereo načinih, ki se jih danes uporablja pri 3D sprejemnikih. Omogoča naslednje zelo zanimive načine:

- anaglični način (angl. *anaglyphic* – najbolj znan stereo način, ko za gledanje potrebujemo rdeče-modra očala)
- način štirikratnega medpomnenja (angl. *quad buffer* - ta način uporabljajo določene grafične kartice za pretvorbo iz 2D načina v 3D, obvezna pa je uporaba mežikajočih očal)
- navpični in vodoravni prelom slike (angl. *horizontal/vertical split* - tehnologija, ki se uporablja v 3D sprejemnikih)

Čeprav ima orodje precejšnjo podporo za tako tehnologijo, je strojna oprema še vedno preveč draga. V našem 3D prikazovalniku smo omogočili samo prvega od zgoraj naštetih načinov, saj so rdeče-modra očala danes zelo poceni.

6. Sklepne ugotovitve in nadaljnji razvoj

Namen diplomske naloge je bil predstaviti, kakšne so možnosti za razvoj in implementacijo geografskega informacijskega sistema z novimi in zanimivimi tehnologijami, kot je prikaz v 3D prostoru. Cilj je bil izdelati informacijski sistem, ki je lahko zelo interaktiven, zanimiv, poseben, enostaven za uporabo ter konkurenčen ostalim rešitvam na že tako nasičenem trgu. Upam da se bo dejansko uporabljal in ne bo zavržen ...

Skozi izdelavo tega sistema sem se spoznal z doslej meni neznanimi tehnologijami in orodji, kot je PostgreSQL SUPB ter knjižnicami za virtualno resničnost, kot je OpenSceneGraph. Upam da mi bo pridobljeno znanje še kdaj koristilo, ter se naprej učil in nadgrajeval svoje znanje.

Sam razvoj sistema je potekal brez težav, zataknilo pa se je pri malenkostih, ki smo jih rešili po tehtnem premisleku. Najzahtevnejša je bila implementacija večnitnega delovanja 3D prikazovalnika, saj - kot smo že omenili - OpenSceneGraph trenutno podpira le eno nit in še to sam upravlja v ozadju. Rešitev z našimi dodatnimi nitmi je trenutno najboljša, upam pa, da bodo v skupnosti OpenSceneGraph še naprej razvijali to tehnologijo, ter tako tudi sami razvili možnost dodajanja poljubnega števila niti - brez omejevanja risanja.

Kot vsak razvijalec tudi sam vem, da je moč na tem geografskem informacijskem sistemu še vedno mnogokaj spremeniti, izboljšati ali dopolniti. Večina ostalih GIS sistemov, ki jih lahko kupimo na trgu, omogoča spreminjanje prostorskih podatkov kar iz zemljevida, česar naš sistem trenutno ne omogoča. V prihodnosti, upam, bomo sistem še bolj optimizirali in obogatili z novimi funkcijami, ter ga tako pripravili, da bo še bolj konkurenčen.

Upajmo, da se bo tudi 3D tehnologija in strojna oprema še naprej razvijala in pocenila, da bodo vsi kupci imeli možnost pregledovanja prostorskih podatkov v 3D prostoru, ter tako potovali v virtualnem svetu kar iz naslonjača pred monitorji.

Seznam slik

Slika 1: Osnovni elementi, ki tvorijo GIS sistem.....	6
Slika 2: Razporeditev vektorskih in rastrskih podatkov.....	7
Slika 3: Logotip OpenSceneGraph.....	9
Slika 4: Shema iterativno-inkrementalnega razvoja.....	12
Slika 5: Arhitektura in tok podatkov GIS sistema.....	13
Slika 6: Shema podatkovne baze.....	16
Slika 7: Primer kontrolnega dela – meni.	17
Slika 8: Spletni brskalnik s že pripravljeno spletno stranjo za izbiranje slojev.	18
Slika 9: Zgradba 3D prikazovalnika.....	19
Slika 10: Izris slojev na površju.	21
Slika 11: Grafični prikaz delovanja Intersectorja.....	21
Slika 12: Primerjava hitrosti risanja v eni niti ter v več nitih.....	22
Slika 13: Primer dinamičnih objektov – oseba in vozilo.....	23
Slika 14: Razvoj 3D prikazovalnika.....	24

Seznam tabel

Tabela 1: Nekateri podprti tipi datotečnih formatov v OSG.....	20
--	----

Viri in literatura

[1] What is GIS?

Dostopno na: <http://www.gis.com/content/what-gis>

[2] Geographic information system - Wikipedia, the free encyclopedia.

Dostopno na: http://en.wikipedia.org/wiki/GIS_data

[3] OGC Standards and Specifications.

Dostopno na: <http://www.opengeospatial.org/standards>

[4] Microsoft Visual Studio - Wikipedia, the free encyclopedia.

Dostopno na: http://en.wikipedia.org/wiki/Microsoft_Visual_Studio

[5] OpenSceneGraph – osg.

Dostopna na: <http://www.openscenegraph.org/projects/osg>

[6] wxWidgets: A portable C++ and Python GUI toolkit.

Dostopno na: http://docs.wxwidgets.org/stable/wx_contents.html

[7] PostgreSQL - Wikipedia, the free encyclopedia.

Dostopno na: <http://en.wikipedia.org/wiki/PostgreSQL>

[8] PostGIS : Home.

Dostopno na: <http://postgis.refrations.net/>

[9] Table of Contents : CodeIgniter User Guide.

Dostopno na: http://codeigniter.com/user_guide/toc.html

[10] Iterative and incremental development - Wikipedia, the free encyclopedia.

Dostopno na: http://en.wikipedia.org/wiki/Iterative_and_incremental_development

[11] jQuery: The Write Less, Do More, JavaScript Library.

Dostopno na: <http://jquery.com/>

[12] OpenSceneGraph Quick Start Guide. Dostopno na:

http://www.lulu.com/items/volume_51/767000/767629/3/print/OSGQSG.pdf