



OPENLAYERS JA SEN KÄYTTÖ PROJEKTIHALLINTAOHJELMISTOSSA

Samuli Lappi

Opinnäytetyö
Elokuu 2011
Tietotekniikan koulutusohjelma
Ohjelmistotekniikan
suuntautumisvaihtoehto
Tampereen ammattikorkeakoulu

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

SISÄLLYSLUETTELO

1 JOHDANTO.....	4
2 TAUSTAA.....	5
2.1 OpenLayers-kehys.....	5
2.1.1 Kerrokset.....	6
2.1.2 Piirrettävät elementit.....	7
2.2 PlanMan Project –projektinhallintaohjelmisto.....	7
2.3 Ohjelmointityökalut.....	9
3 TYÖN SISÄLTÖ.....	10
3.1 Suunnittelu.....	10
3.2 Toteutus.....	11
3.3 Testaus ja integrointi.....	14
4 LOPPUTULOKSET JA JOHTOPÄÄTELMÄT.....	15
LÄHTEET.....	16
LIITTEET.....	17

LYHENTEIDEN JA MERKKIEN SELITYKSET

API	Application programming interface, eli ohjelmointirajapinta
CSS	Cascading Style Sheet, tyylitiedosto
HTML	Hypertext Markup Language, verkkosivuja varten kehitetty kuvauskieli
HTTP	Hypertext Transfer Protocol, hypertekstin siirtoprotokolla
JS	JavaScript
PHP	Hypertext Preprocessor, verkkosivuja varten kehitetty komentosarjakieli

1 JOHDANTO

Työn tarkoitus oli parantaa PlanMan Project -projektinhallintaohjelman karttakäyttöliittymän graafisuutta. Pääasiallisena työkaluna oli JavaScriptillä tehty avoimen lähdekoodin OpenLayers -kehys, jota käyttäen karttakäyttöliittymä toteutettiin. Olemassa ollut karttakäyttöliittymä oli toiminnallisuudeltaan yksinkertainen: se tuki eri kuvioiden piirtoa ja tallennusta. Projekti oli hyvin asiakaslähtöinen, sillä monet PlanManin asiakkaista olivat toivoneet enemmän graafisuutta karttaan.

PlanMan Project on projektinhallintaohjelmisto, jonka on kehittänyt suomalainen PlanMan Oy. Karttakäyttöliittymän ideana on, että jokaiseen projektissa olevaan resurssiin voidaan liittää koordinaatteja, jotka kuvaavat joko pistettä, viivaa, tai aluetta.

OpenLayers on avoimen lähdekoodin JavaScript-kirjasto, jonka avulla voidaan samalle pohjalle piirtää useiden eri palveluntarjoajien karttoja, sekä erilaisia kuvioita. OpenLayersiä on kuvattu tarkemmin kappaleessa 2.1.

Työssä perehdytään tarkemmin OpenLayersin toimintaan ja ominaisuuksiin, sekä esitellään lopuksi PlanMan Projectiin tehdyt karttakäyttöliittymän parannukset.

2 TAUSTAA

Tässä kappaleessa esitellään OpenLayersistä työssä tarvituista ominaisuuksista keskeiset piirteet, sekä esitellään hieman PlanMan Project -ohjelmaa. Lisäksi käydään läpi työssä käytetyt ohjelmointityökalut.

2.1 OpenLayers-kehys

OpenLayers on oliopohjainen JavaScriptillä toteutettu ohjelmistokehys. OpenLayers tarjoaa helppokäyttöisen ohjelmointirajapinnan, jonka avulla erilaisia toimintoja on helppo toteuttaa. Tämä kehys on avointa lähdekoodia, mikä tarkoittaa sitä, että kuka tahansa voi muokata lähdekoodia omiin tarkoituksiinsa sopivaksi. OpenLayers sopii kuitenkin useimpiin tarkoituksiin sellaisenaan. (OpenLayers: Home)

Monet palvelut, kuten Google ja OpenStreetMap, tarjoavat karttapalveluitaan ilmaiseksi käyttöön. Näihin sisältyy myös rajapinta, jonka avulla karttoja voidaan muokata kuhunkin palveluun sopivaksi. Tarjolla on usein useita karttapohjia, kuten tavallinen kartta, satelliittikuva, ja katunäkymä. Lisäksi tarjotaan monia muita ominaisuuksia, kuten erilaisten kuvioden piirtämistä kartalle, sekä koordinaattien muuttamista osoitteeksi ja päinvastoin. Nämä piirtotyökalut toimivat vain kyseisen palveluntarjoajan kartoille, joten jokaista palveluntarjoajaa kohti pitäisi tällöin tehdä erikseen samat toiminnallisuudet. (OpenStreetMapWiki, Google Maps API Family)

OpenLayers'in suurin vahvuus on se, että se ei ole sidottu yhden palveluntarjoajan karttapalveluun, vaan sen avulla voidaan ladata kartat mm. Google Mapsista, Bingistä, tai OpenStreetMapsista. Tämä ns. peruskerros on valittavissa ohjelmallisesti, ja käyttäjät voivat itse valita itselleen sopivimman. Peruskerroksen lisäksi voidaan luoda muita kerroksia, joihin on helppo upottaa erilaisia kuvioita ja merkintöjä, ja jotka on helppo piilottaa halutessa (kuva 1). OpenLayersin tarkoitus onkin erottaa itse karttatiedot karttatyökaluista, jolloin yhdelle pohjalle piirretyt elementit toimivat myös muilla karttapohjilla. (OpenLayers: Home)



KUVA 1: Esimerkki OpenLayers -kerrosvalintänäkymästä.

2.1.1 Kerrokset

Kaikki OpenLayersillä näytettävät karttapohjat sekä kuviot piirretään kerroksille. Näitä on useita eri tyyppisiä – eri palveluntarjoajien karttapohjille on jokaiselle oma luokkansa, ja piirrettäville elementeille on kokoelma erilaisia kerrostyyppejä. Kerrostyypit voivat olla tietyille kuvioille, esimerkiksi pisteille, suunniteltuja, tai sitten kaikkia piirrettäviä elementtejä tukevia. (OpenLayers API documentation)

Kaikki OpenLayersin tukemat kerrokset löytyvät `OpenLayers.Layer` -luokan alta. Luokasta löytyy aliluokat jokaista tuettua palveluntarjoajaa kohti erikseen. Lopullista päätöstä käytettäviä karttapohjia varten ei tehty, mutta projektin aikana käytössä oli Googlen, Yahoo!n, sekä OpenStreetMapsin karttapohjat.

Erilaisia piirtoelementtejä varten olevat kerrokset ovat pääperiaatteeltaan kahden tyyppisiä: `vector` ja `marker`. `Vector`-tyyppinen kerros tukee nimensä mukaisesti vektorigrafiikkaa, kun taas `marker`-tyyppinen kerros tukee HTML-kuvaelementtejä. (Overlays – OpenLayers)

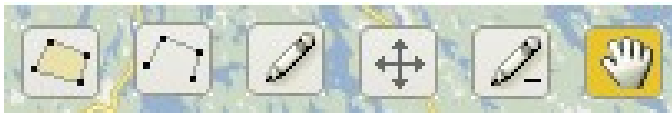
Piirrettäviä elementtejä varten luodut kerrokset ovat tyypiltään `Layer.Vector` -kerroksia. `Vector`-kerros tukee useita erilaisia piirto-ominaisuuksia, joten se oli paras valinta tähän

tarkoitukseen. Kerroksen tyylimäärittelyihin käytettiin `OpenLayers.StyleMap` –luokkaa, joka kerrosta luotaessa annettiin parametrina.

2.1.2 Piirrettävät elementit

`OpenLayers`istä löytyy useita valmiita työkaluja erilaisten elementtien piirtämiseen. Piirtotyökalu löytyy `Control.DrawFeature` -luokasta. Tästä luokasta voidaan luoda instansseja, joille annetaan parametrina halutun piirtotyökalun tyyppi. Nämä työkalut laitetaan työkalupalkkiin, joka on tyypiltään `Control.Panel`.

Karttakäyttöliittymässä käytetty työkalupalkki on tyypiltään `OpenLayers.Control.Panel`. Tähän on upotettu kuusi erilaista työkalua piirtoa ja kartan liikuttamista varten. Piirtotyökaluissa on käytetty luokkaa `Control.DrawFeature`. Tälle annetaan parametrina `Handler` -olio, joka kertoo minkä tyyppistä elementtiä ollaan piirtämässä. Alue on toteutettu `Polygon` -tyyppisenä, reitti `Path` -tyyppisenä, ja piste `Point` -tyyppisenä. Kuvassa 2 on esimerkki työkalupalkista.



KUVA 2: Esimerkki `OpenLayers` työkalupalkista

Kuvioita voidaan lisätä myös ohjelmallisesti. Tämä tapahtuu kutsumalla halutun kerroksen `addFeatures` -funktiota, jolle annetaan parametrina piirrettävä elementti.

2.2 PlanMan Project –projektinhallintaohjelmisto

PlanMan Project on suomalaisen PlanMan Oy:n kehittämä ja ylläpitämä projektinhallintaohjelmisto. PlanMan Project toimii Windows XP/Vista/7 -ympäristöissä. Ohjelmistossa on hyödynnetty innovatiivisesti karttakäyttöliittymää. Tällaista ratkaisua ei muissa projektinhallintaohjelmissa ole vielä olemassa. (PlanMan Project 2010 tuote-esittely)

PlanMan Projectissa on panostettu runsaasti tietojen visuaaliseen esittämiseen. Pyrkimyksenä on, että mahdollisimman vähällä tiedollakin saadaan piirrettyä runsaasti erilaisia kaavioita ja tehtyä projektia edesauttavia laskelmia. Kilpaileviin tuotteisiin verrattuna PlanMan on onnistunut tässä tavoitteessaan hyvin.

Kuvassa 3 on esitetty kuvankaappaus PlanMan Project -ohjelmistosta. Käyttöliittymän toteutukseen on käytetty Microsoftin työkaluja, joten käyttöliittymän ulkoasu muistuttaa monilta osin Microsoft Office -tuotteiden käyttöliittymää.

Karttakäyttöliittymä on toteutettu ohjelmaan upotetulla selainikkunalla. Karttaliittymän ohjelmakoodit sijaitsevat PlanManin omalla palvelimella, johon tämä selainikkuna ohjataan. Tästä syystä karttakäyttöliittymä vaatii toimiakseen internet-yhteyden. Selainta ei voi ohjata muihin osoitteisiin.

Ohjelmiston lisäksi PlanMan tarjoaa ohjelman käyttöönotto-, perus-, sekä kustomointikoulutusta.

The screenshot displays the PlanMan Project software interface. The main window is divided into several sections:

- Top Menu:** TYÖT, KARTAT, Kuormitus.
- Project List Table:**

Hier	Nimi	Määrä	Yks	Kap pv
1 -1	Sudenkatu joen perkaaminen	30 m		10
2 1.1	Joen perkaaminen	30 m		10
3 -2	Lahdenperänkatu kunnostustöitä	300 m		30
4 2.1	Katuvalojen uusiminen	10 kpl		2
5 2.2	Bussipysäkin siirto	1 kpl		0,33
6 2.3	Puhelinkaapelin uusiminen	20 m		20
7 -3	Takahuhdintie päällystys ja rumpu	300 m		33
8 3.1	Päällystekorjaukset	200 m		40
9 3.2	Tierummun vaihto	20 m		20
10 -4	Hakametsän hallin pysäköintialueen parantaminen	5000 m2		208
11 4.1	Päällystystyöt	3000 m2		300
12 4.2	Autopaikkamaalaukset	500 kpl		100
13 4.3	Suojakaiteet	100 m		33
14 4.4	Liikennemerkit	5 kpl		5
- Map View:** A Google Maps-style map showing a street layout with a highlighted area in grey, likely representing the project site. The map includes street names like Sudenkatu, Kalleenkatu, and Takahuhtitie.
- Resource Table:**

Koodi	Nimi	Kpl	Määrä suhde	Määrä	Yks	Menekki	h	Ke
1 RAM	Rakennusammattimies	1		30 h		0,80	24	
2 RM	Rakennusmies	1		30 h		0,80	24	
- Bottom Panel:** Includes tabs for Mitoitus, Ajoitus, Kustannukset, Kassavirta, Resurssit, Ulkoasu, Lomake, Aikataulukko, Kuvaaja, and Matriisi. A legend at the bottom right shows color-coded boxes for Suunnitelma (red), Tavoite (grey), Toteutuma (black), and Jäpette (yellow).

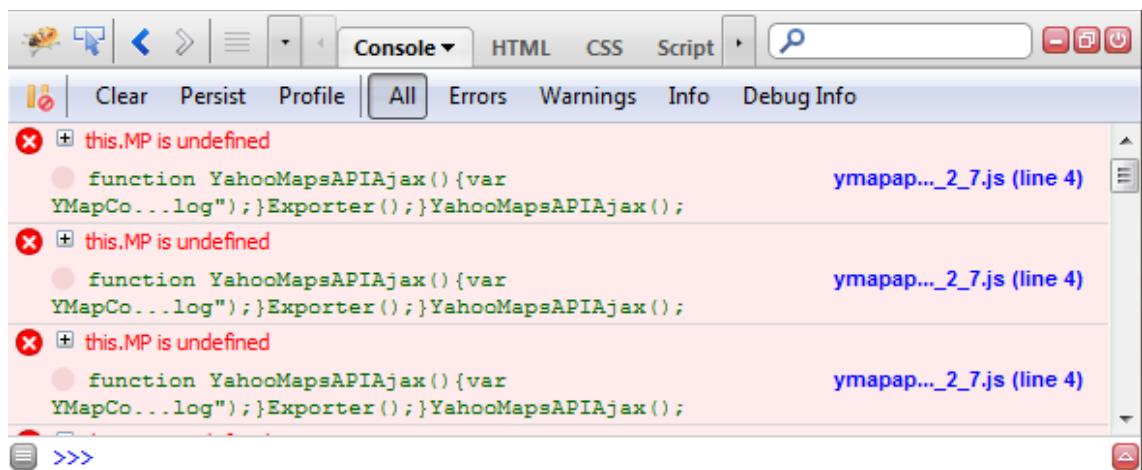
KUVA 3: Kuvankaappaus PlanMan Project -ohjelmistosta

PlanMan Project -ohjelmisto sopii hyvin mm. rakennusalalle, teollisuuteen, sekä tuotantoprojekteihin (PlanMan Project 2010 Tuote-esittely).

2.3 Ohjelmointityökalut

Työssä käytetty ohjelmointikieli on JavaScript. JavaScript ei ole käännettävä kieli, vaan se ajetaan sellaisenaan. Tästä syystä mahdollisten koodivirheiden löytämiseen piti käyttää muita menetelmiä. Päädyttiin käyttämään Mozilla Firefox –selaimelle tehtyä liitännäistä nimeltä Firebug. Firebug havaitsee JavaScript-koodissa olevat virheet ohjelman ajamisen aikana. Kuvassa 4 on esitetty näkymä Firebugin konsoli-ikkunasta. Konsoli kertoo, minkä virheen se on havainnut, missä tiedostossa virhe on, sekä rivinumeron. (What is Firebug?)

Firebugia voidaan käyttää myös mm. CSS-tyylitiedostojen tarkasteluun ja ajonaikaiseen muokkaamiseen, HTTP-kyselyiden ja vastausten tarkastelemiseen, sekä JavaScript muuttujien arvojen muokkaamiseen ajon aikana.



KUVA 4: Näkymä Firebugin konsolista

Koodin käsittelyyn käytettiin ohjelmointiin tarkoitettua ilmaista Notepad++ –ohjelmaa. Se on avoimen lähdekoodin tekstieditori, joka tunnistaa eri ohjelmointikielten rakenteen, ja tukee syntaksin korostusta. Tämä helpottaa ohjelmointityötä huomattavasti. Kuvassa 5 on esitetty näkymä Notepad++ -ohjelmasta. (Notepad++)

Ohjelman testaukseen käytettiin sekä Mozilla Firefox että Internet Explorer –selaimia. Työn vaatimuksena oli, että se toimii täydellisesti Internet Explorerissa, mutta Firefoxille oli saatavilla paremmat työkalut ohjelmointityötä varten. Näin ollen ohjelman testaus suoritettiin ensin Firefoxilla, ja tämän jälkeen varmistettiin toiminta Internet Explorerissa. Ohjelmakoodin toimivuudessa oli yllättävän suuria eroja näiden selainten välillä.

```
97     function init()
98     {
99         debug = document.getElementById('debug');
100        debug.innerHTML = "Debugging enabled<br />";
101        // the map
102        map = new OpenLayers.Map('map',
103            {
104                controls:
105                [
106                    new OpenLayers.Control.Navigation(),
107                    new OpenLayers.Control.PanZoomBar(),
108                    new OpenLayers.Control.KeyboardDefaults()
109                ],
110                numZoomLevels: maxZoom,
111                projection: new OpenLayers.Projection("EPSG:900913"),
112                displayProjection: new OpenLayers.Projection("EPSG:900913"),
113                units: "degrees",
114                numZoomLevels: 18,
115                maxResolution: 156543.0339,
116                maxExtent: new OpenLayers.Bounds(-20037508, -20037508,
117                                                    20037508, 20037508)
118            });
```

KUVA 5: Kuvankaappaus Notepad++ -ohjelmasta

3 TYÖN SISÄLTÖ

PlanMan Project on toteutettu C++ -kielellä, ja karttakäyttöliittymän JavaScript-funktioita kutsutaan suoraan C++:sta. Työssä oli tarkoitus tehdä parannuksia nimenomaan karttakäyttöliittymän JavaScript-osioon. Projektin alussa työn sisältö suunniteltiin yhdessä PlanMan Oy:n kanssa. Työ jaettiin karkeasti kolmeen osaan: suunnitteluun, toteutukseen ja testaukseen.

3.1 Suunnittelu

Suunnitteluvaiheessa päätettiin, mitä ominaisuuksia karttakäyttöliittymään piti toteuttaa. Kartalle piirrettävät kuviot – pisteet, viivat sekä alueet – päätettiin värikoodata siten, että nollan prosentin valmius kuvataan punaisella, sadan prosentin valmius vihreällä, ja muut valmiudet keltaisella. Karttakäyttöliittymän JavaScript -koodi ei ota kantaa siihen, mistä prosenttiosuudet lasketaan, vaan ne tarjotaan parametrina C++ -ohjelmasta. Värikoodaus päätettiin toteuttaa kerroksia (layer) käyttäen. Kerroksen elementeille voi asettaa helposti yhtenäisen tyylin, jolloin jokaista elementtiä ei tarvitse tyyllittää erikseen. Lisäksi elementtien piilottaminen käy kerroksia käyttämällä erittäin helposti.

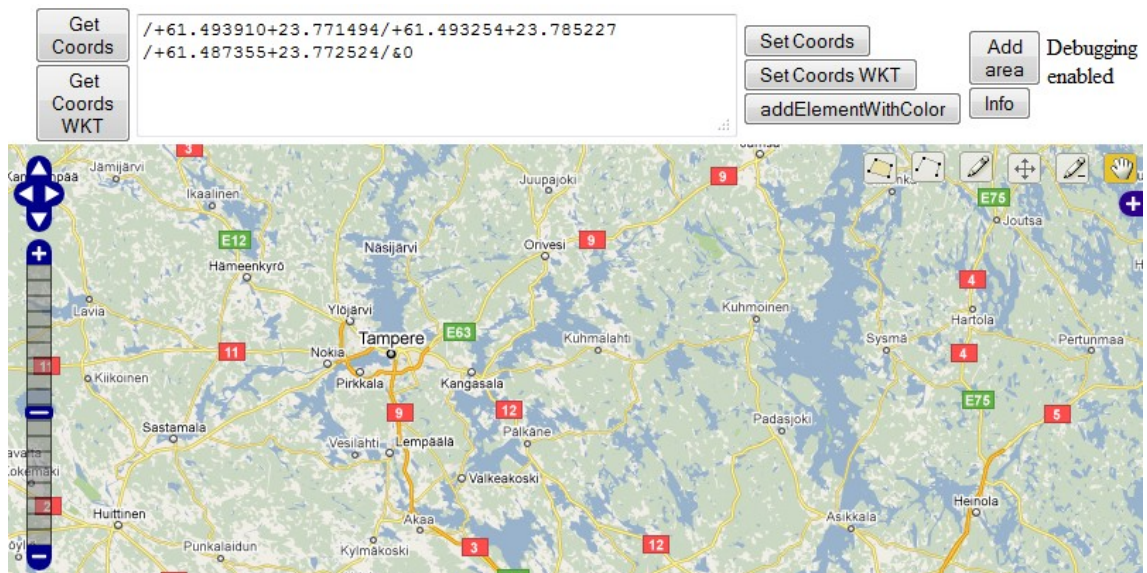
Karttakäyttöliittymään haluttiin lisäksi ominaisuus, jolla käyttäjä voi helposti siirtää kartan keskikohdan haluamaansa paikkaan kirjoittamalla paikan osoitteen hakukenttään. Tämä toiminto edellyttää ns. GeoCoding-ominaisuutta, jota tarjoaa ilmaiseksi mm. Google. Palveluntarjoajan valinta jätettiin tässä vaiheessa avoimeksi.

PlanMan Project -ohjelman laajuuden takia karttakäyttöliittymän ohjelmointivaiheen testauksen suorittaminen itse ohjelmassa ei katsottu olevan tehokasta. Tämän vuoksi päätettiin, että ensimmäiseksi toteutetaan selainpohjainen kehys, joka simuloi PlanMan Projectin lähettämiä JavaScript-kutsuja. JavaScriptin tuli olla yhteensopiva Internet Explorer -selaimen kanssa.

3.2 Toteutus

Toteutusvaiheessa ensimmäiseksi toteutettiin selainpohjainen testiympäristö. Tämän toteutuskielenä käytettiin PHP:tä, joka laitettiin omalle testiserverille. Tästä eteenpäin ei karttakäyttöliittymän kehittäminen poikennut juurikaan tavanomaisesta web-ohjelmoinnista. Testiympäristön tarkoitus oli kutsua JavaScript-funktioita samankaltaisilla parametreilla kuin PlanMan Project. Lisäksi ympäristöön sai helposti upotettua työtä helpottavia toimintoja, kuten elementtien laskeminen ja tyhjentäminen.

Lopullinen testisivu muodostui työkaluosasta ja karttaosasta. Työkaluosaan kuului tekstikenttä, johon syötettiin piirrettävän kuvion koordinaatit sekä prosenttimäärä. Tämän lisäksi siihen kuului elementtien laskenta, kaikkien elementtien poisto ja piirrettyjen elementtien koordinaattien haku tallentamista varten. Karttaosaan piirrettiin itse kartta, täysin samanlaisena kuin se näkyy PlanMan Project -ohjelmassa. Testiympäristön kuvankaappaus löytyy kuvasta 6. Google Mapsin karttaphojien näyttämistä varten piti generoida Google Maps API key, yksityinen avain, joka liitettiin osaksi Google Maps API:n osoitetta.



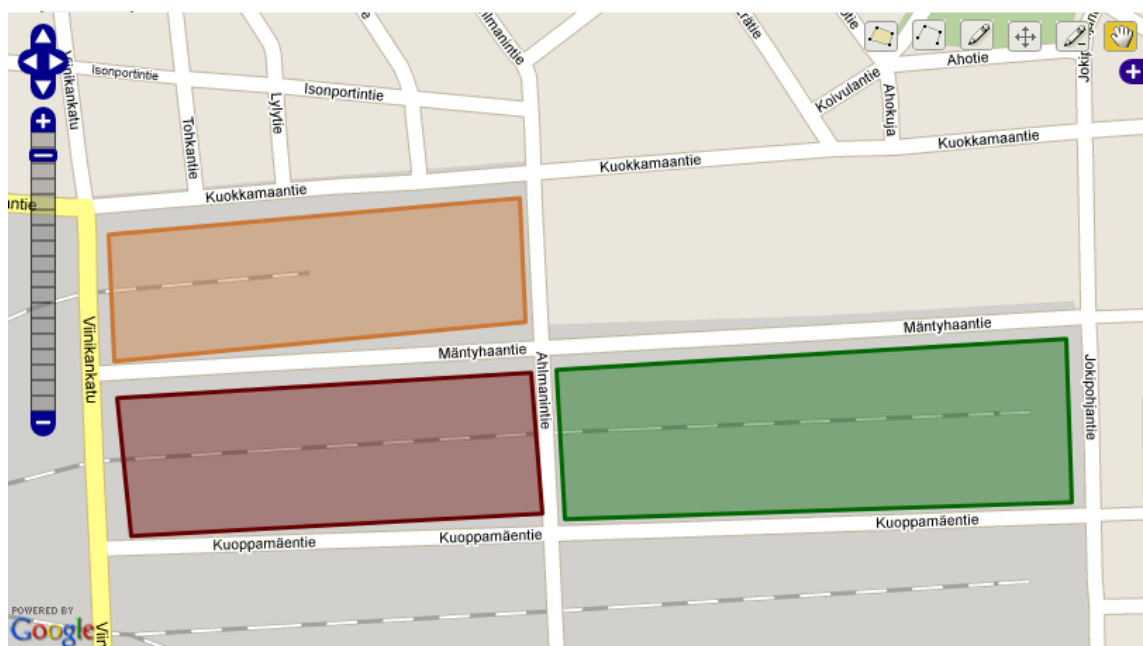
KUVA 6: Kuvankaappaus testiympäristöstä Mozilla Firefox -selaimessa.

Karttaosassa on työkalut elementtien käsittelyä varten. Nämä ominaisuudet ovat pisteen piirto, viivan piirto, kuvion piirto, elementin siirtäminen, ja elementin poisto. Näiden

lisäksi on tarjolla navigointityökalu. Työkalut oli toteutettu OpenLayersin valmiilla työkaluluokilla, ja ne oli sijoitettu kartan oikeaan ylälaitaan paneeliin.

Piirtotyökalut ovat kerroskohtaisia. Tästä syystä työkaluihin piti tehdä joitakin muutoksia. Mikäli käyttäjä valitsee useamman resurssin näytettäväksi kartalla yhtä aikaa, saattavat elementit olla eri kerroksilla, jolloin muokkaustyökalu ei toimisi molemmille kuvioille. Tämä ratkaistiin poistamalla piirtotyökaluja käytöstä seuraavasti: kun yhtään elementtiä ei vielä ole piirretty, kaikki työkalut ovat käytettävissä. Kun yksi elementti on piirretty, poistuu mahdollisuus lisätä uusia elementtejä, sillä jokaisella resurssilla voi olla vain yksi kuvio. Mikäli käyttäjä valitsee ohjelmassa usean eri resurssin näytettäväksi kerralla kartassa, ei piirto- ja poistotyökaluja näytetä lainkaan, vaan pelkästään navigointityökalu.

Piirrettäviä elementtejä varten luotiin kolme kerrosta alkuperäisessä toteutuksessa olleen yhden kerroksen sijaan. Kerrosten näkyvyys on valittavissa kerrosvalitsimella (layer switcher), joka on yksi OpenLayers kirjaston perusosia. Kerrokset nimettiin 0%, 10-90% ja 100% kerroksiksi. Näistä voidaan kerralla näyttää niin monta kuin halutaan, tai piilottaa kaikki. Kun sivu ladataan, nämä kerrokset alustetaan, ja niille asetetaan tyyli. Tyyli sisältää värin lisäksi tiedot viivojen paksuudesta, sekä alueiden täyttövärin läpinäkyvyydestä. Nämä ominaisuudet ovat kaikille kerroksille samat. Värin asetettiin niin, että 0% kerros on punainen, 10-90% kerros keltainen ja 100% kerros vihreä.



KUVA 7: Kolme eri kerroksille piirrettyä aluetta karttapohjalla

Kuvassa 7 on esitetty esimerkkinäkymä karttapohjasta, johon on piirretty kolme kuviota eri kerroksille. Kuvassa 8 taas on esitetty koodiesimerkki kerrosten ja tyylimäärittelyiden tekemisestä. Kun elementin piirtävää funktiota kutsutaan, sille annetaan parametrina piirrettävän elementin prosenttiarvo, ja elementti sijoitetaan sen perusteella oikealle kerrokselle. Piirto tapahtuu automaattisesti. Kerrokset ovat OpenLayers kirjaston Vector-kerroksia.

```

styleBad = new OpenLayers.StyleMap(OpenLayers.Util.applyDefaults(
    {
        fillColor: "#660000", // punaisen värin heksa-arvo
        strokeColor: "#660000"
    },
    OpenLayers.Feature.Vector.style["default"])
);
styleNeutral = new OpenLayers.StyleMap(OpenLayers.Util.applyDefaults(
    {
        fillColor: "#cc7733", // keltaisen värin heksa-arvo
        strokeColor: "#cc7733"
    },
    OpenLayers.Feature.Vector.style["default"])
);
styleGood = new OpenLayers.StyleMap(OpenLayers.Util.applyDefaults(
    {
        fillColor: "#006600", // vihreän värin heksa-arvo
        strokeColor: "#006600"
    },
    OpenLayers.Feature.Vector.style["default"])
);

vectorsBad = new OpenLayers.Layer.Vector("0%", {styleMap: styleBad});
vectorsNeutral=new OpenLayers.Layer.Vector("10%-90%", {styleMap: styleNeutral});
vectorsGood = new OpenLayers.Layer.Vector("100%", {styleMap: styleGood});

map.addLayers([vectorsBad, vectorsNeutral, vectorsGood]);

```

KUVA 8: Koodiesimerkki kerrosten luomisesta ja niiden tyylimäärittelyistä.

Elementtien muokkaustyökalulla voi siirtää kaikkia elementin pisteitä, kuten esimerkiksi kulmia, vapaasti kartalla. Poistotyökalu poistaa koko elementin. Ongelmaksi muodostui se, että piirtotyökalut ovat kerroskohtaisia, joten jos kuvioita oli eri kerroksilla, ei muokkaus- ja poistotyökaluja voida käyttää. Ongelma ratkaistiin siten, että työkalut poistettiin käytöstä mikäli elementtejä on näkyvässä useampi. Kun kartta

ladataan, sille annetaan parametrina prosenttiarvo, ja sen mukaan alustetaan piirtotyökalut oikealle kerrokselle.

3.3 Testaus ja integrointi

Ohjelmiston testaus suoritettiin kehitystyön ohessa. Erillistä testaussuunnitelmaa ei koettu tarpeelliseksi tehdä. Ohjelman osien toimivuus testattiin selaimen tehdyssä testiympäristössä, joka vastasi toiminnaltaan PlanMan Projectin toimintoja. Toimivuus testattiin sekä Mozilla Firefox- että Internet Explorer –selaimissa.

Integraatiotyö itse ohjelmistoon ei kuulunut projektiin, vaan se jätettiin myöhemmin toteutettavaksi.

4 LOPPUTULOKSET JA JOHTOPÄÄTELMÄT

OpenLayers on toiminnoiltaan erittäin monipuolinen ohjelmistokehys. Sen toiminnoista löytyy lähes kaikki samat ominaisuudet, jotka ovat myös palveluntarjoajien omissa rajapinnoissa. OpenLayersin heikko kohta on kuitenkin dokumentaatio – vaikka kaikkien luokkien kuvaukset ovat dokumentaatiossa olemassa, ne ovat suurelta osin puutteellisia, eivätkä siis riitä toimintojen täydelliseen ymmärtämiseen.

Lopputuloksena työssä saatiin aikaiseksi eri väristen kerroksien tuki karttakäyttöliittymään. Alkuvaiheessa suunniteltua osoitteen perusteella tehtävää kartan keskittämistä ei voitu toteuttaa, sillä Googlen karttojen ilmaiskäyttö on rajoitettua, eikä haluttu ottaa sitä riskiä, että ilmaiset latauskerrat menisivät täyteen, jolloin Googlen kartat eivät toimisi lainkaan.

Viime aikoina kilpailijat ovat kirineet kiinni Googlen suosiota karttapalvelujen ykkösenä. Tämä trendi voi aiheuttaa ohjelmistokehittäjille päänvaivaa karttapalvelujen tarjoajaa valittaessa. OpenLayers tarjoaa hyvän mahdollisuuden jättää palveluntarjoajan valinta myöhemmäksi, tai jopa käyttäjän itse valitsemaksi. (Google vs Yahoo vs Bing vs Mapquest)

Työtä kirjoitettaessa ei ole vielä tiedossa, miten karttaliittymän parannusten integraatio on onnistunut tai millainen vastaanotto asiakkailta on ollut tähän.

LÄHTEET

Sähköiset lähteet

1. PlanMan Project 2010 tuote-esittely [pdf-dokumentti]. [viitattu 8.4.2011].
Saatavissa: <http://www.planman.fi/pdf/PlanMan%20Project%202010%20Tuote-esittely.pdf>
2. OpenLayers: Home [www-sivu]. [viitattu 10.4.2011].
Saatavissa: <http://openlayers.org/>
3. Google vs Yahoo vs Bing vs Mapquest? [www-sivu]. [viitattu 27.4].
Saatavissa: <http://www.jlramirez.info/2010/07/google-vs-yahoo-vs-bing-vs-mapquest-what-do-people-use-2/>
4. Overlays – OpenLayers [www-sivu]. [viitattu 27.4].
Saatavissa: <http://docs.openlayers.org/library/overlays.html>
5. Map Features – OpenStreetMap Wiki [www-sivu]. [viitattu 31.5].
Saatavissa: http://wiki.openstreetmap.org/wiki/Map_Features
6. Google Maps API Family [www-sivu]. [viitattu 1.6].
Saatavissa: <http://code.google.com/intl/fi/apis/maps/index.html>
7. OpenLayers API documentation [www-sivu]. [viitattu 1.6].
Saatavissa: <http://dev.openlayers.org/releases/OpenLayers-2.10/doc/apidocs/>
8. What is firebug? [www-sivu]. [viitattu 1.6].
Saatavissa: <http://getfirebug.com/whatisfirebug>
9. Notepad++ [www-sivu]. [viitattu 1.6].
Saatavissa: <http://notepad-plus-plus.org/>

LIITTEET

1. Tiivistelmä
2. Abstract

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikan suuntautumisvaihtoehto

LAPPI, SAMULI: OpenLayers ja sen käyttö projektinhallintaohjelmistossa

Opinnäytetyö 18 s., liitteet 2 s.
Elokuu 2011

Tämän opinnäytetyön tarkoituksena oli kehittää PlanMan Project – projektinhallintaohjelmiston karttakäyttöliittymän graafisuutta. Tavoitteena oli mahdollistaa kartalle piirrettävien kuvioiden värikoodaus, sekä mahdollistaa kartan siirtäminen käyttäjän antaman osoitteen perusteella. Karttakäyttöliittymä on toteutettu JavaScript-ohjelmointikielellä, käyttäen avoimen lähdekoodin OpenLayers-ohjelmistokehystä. Kartan ideana on, että jokaiseen projektissa olevaan resurssiin voidaan liittää koordinaatteina joko piste, viiva tai alue.

Työn alkuosassa käydään läpi OpenLayer-kehityksen ominaisuuksia ja toimintaa, sekä esitellään työssä käytettyjä työkaluja. Tämän jälkeen esitellään ohjelmakoodiin tehtyjä muutoksia.

Työn lopputuloksena saatiin karttakäyttöliittymään kuvioiden värikoodaus. Kuviot väritetään esimerkiksi niitä koskevan resurssin valmiusasteen mukaan.

Asiasanat: kartta, käyttöliittymä

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in computer engineering
Option of software engineering

LAPPI, SAMULI: OpenLayers and it's usage in project management software

Bachelor's thesis 18 pages, appendices 2 pages
August 2011

The purpose of this project was to enhance the visual feedback of map interface in a project management software called PlanMan Project. The goal was to allow color coding of different drawable map elements and to center the map according to an address given by a user. The map interface is implemented using JavaScript-language and an open source framework OpenLayers. The idea of the map is that every resource in the project may hold coordinate data which contains either a dot, line, or an area.

This thesis goes through the functionality and features of OpenLayers-framework and the tools used in the project. After the introduction the changes to the code are explained.

As a result of the project, the color coding of the map interface was finished. The shapes on the map are colored i.e. according to the readiness of the corresponding resource.

Keywords: map, user interface