



**PUUVIRTA – PUUNHANKINNAN
VIRTUAALINEN
OPPIMISYMPÄRISTÖ**

Tilannekohtaisen ohjeen toteutus

Petri Rantala

Opinnäytetyö
Marraskuu 2013
Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikka
Ohjelmistotekniikka

PETRI RANTALA

Puuvirta – Puunhankinnan virtuaalinen oppimisympäristö:
Tilannekohtaisen ohjeen toteutus

Opinnäytetyö 25 sivua
Marraskuu 2013

Tämän opinnäytetyön tarkoituksena oli muuntaa Puuvirta-oppimisympäristön olemassa oleva käyttöohje tilannekohtaiseksi ohjeeksi Puuvirta-asiakasohjelmistoon. Työssä luotiin myös ohjemateriaalille sellainen perusta, jota voitaisiin tulevaisuudessa jatkokehittää niin muiden opiskelijoiden kuin opettajien toimesta. Työ tehtiin oppimisympäristön jatkokehityksen ohessa, jona aikana tutustuttiin myös oppimisympäristön toimintaan ja asiakasohjelmistossa käytettyyn teknologiaan.

Puuvirta-asiakasohjelmisto on C# -ohjelmointikielellä toteutettu .NET-sovellus, jonka toteutuksessa on hyödynnetty myös C++/CLI -kielellä kehitettyjä komponentteja. Ohjemateriaalin perusta luotiin HTML-dokumenttina, jossa hyödynnetään CSS-tekniikkaa sekä JavaScript-kieltä.

Tässä raportissa esitellään Puuvirta-oppimisympäristö, työssä käytetty kehitysympäristö ja teknologiat. Dokumentissa käydään läpi myös suunnittelussa huomioon otetut rajoitukset ja niiden perusteella tehdyt ratkaisut.

Raportin lopussa pohditaan myös opinnäytetyön tuloksia.

ABSTRACT

Tampere University of Applied Sciences
Degree of Computer Science
Software Development

PETRI RANTALA

Puuvirta – Virtual Learning Environment:
Development of context sensitive help

Bachelor's thesis 25 pages
November 2013

The purpose of this thesis was to convert the existing user manual for Puuvirta virtual learning environment in to a context sensitive help function for the Puuvirta client software. A template was also created for the help content, which can be used as a base for further development of the help material by both the students and the teachers.

Technical implementation of this thesis was done alongside with the development of additional features for the learning environment. During this time the functionality, as well as the technologies used in the application were thoroughly examined.

Puuvirta client software is a .NET application that is written in C#, which uses components that are written in C++/CLI. Template for the help content was written as a HTML document that takes advantage of CSS techniques and JavaScript scripting language.

First portion of this document contains an introduction to Puuvirta virtual learning environment and briefly describes the development environment, as well as the technologies used during the project.

The second portion of the document focuses on describing the design and development process of the context sensitive help function. Structure of the client software and the development constraints and requirements set for the project are also examined.

The final portion of the document contains some thoughts about the thesis work in general and the overall progress throughout the project.

Key words: context sensitive help, C#, .NET, HTML, CSS, JavaScript, virtual learning environment

SISÄLLYS

1	JOHDANTO.....	6
2	PUUVIRTA-OPPIMISYMPÄRISTÖ.....	7
3	TYÖKALUT JA TEKNOLOGIA.....	9
	3.1 Visual Studio -kehitysympäristö.....	9
	3.2 .NET Framework -alusta.....	10
	3.3 C#-kieli	10
	3.4 HTML-kieli ja CSS-tyyliohje	10
	3.5 JavaScript-kieli	12
4	TYÖN SUUNNITTELU	14
	4.1 Puuvirta-oppimisympäristön rakenne.....	14
	4.2 Vaatimukset ja rajoitukset	14
	4.3 Työssä käytetty ratkaisu.....	15
	4.4 Hylätyt ratkaisut.....	16
	4.4.1 HelpProvider -luokka.....	16
	4.4.2 PDF- ja docx -tiedostot	16
	4.4.3 CHM -tiedosto.....	17
5	TYÖN TOTEUTUS	18
	5.1 Tilannekohtainen ohjetoiminto.....	18
	5.1.1 XML -asetustiedosto	19
	5.1.2 Toiminnon kehittäminen	19
	5.2 Ohjemateriaalin runko	20
	5.2.1 HTML-dokumentin luonti.....	21
	5.2.2 JavaScript-navigointipalkin ja sivutunnisteen kehitys.....	22
	5.2.3 CSS-tyyliohjeen luonti	23
6	POHDINTA.....	25
	LÄHTEET.....	26

LYHENTEET JA TERMIT

C#	C Sharp -ohjelmointikieli
CSS	Cascading Style Sheets –tyyliohje
HTML	Hyper Text Markup Language –kieli
JavaScript	JavaScript –komentosarjakieli
Oppimisympäristö	Opetuskäyttöön suunniteltu sovellus
.NET	.NET Framework ohjelmistokomponenttikirjasto
TAMK	Tampereen ammattikorkeakoulu
TTY	Tampereen tekninen yliopisto
Widget	Minisovellus

1 JOHDANTO

Tämän työn tarkoituksena on kehittää Puuvirta-oppimisympäristön asiakasohjelmistoon tilannekohtainen ohjetoiminto, jonka avulla käyttäjä pääsee käsiksi suoraan avoimena olevan ikkunan ohjemateriaaliin. Työssä luotiin myös käytettävälle ohjemateriaalille sellainen pohja, jonka muokkaus on helppo toimenpide ja johon talvella 2013 tapahtuva ohjemateriaalin kattava päivitys voidaan toteuttaa.

Tilannekohtaisen ohjeen väliaikaisena sisältönä työssä käytetään Puuvirran alkuperäistä ohjemateriaalia, joka muokataan sen tämän hetkisestä docx/pdf -muotoisesta dokumentista toimimaan ohjemateriaalin uudessa HTML-pohjassa.

Idea tilannekohtaisen ohjeen toteuttamisesta Puuvirta-asiakasohjelmiston päivityksen yhteydessä lähti oppimisympäristön käyttäjiltä saadusta palautteesta. Monet käyttäjistä kokivat uuden työkalun monipuoliseksi ja hyödylliseksi, mutta sen omaksuminen tuntui alkuun hieman vaikealta ja sen käytön tueksi toivottiin lisää koulutusta.

Vaikka tilannekohtainen ohje ei pyri korvaamaan tarvetta lisäkoulutukselle, sen avulla pyritään alentamaan sovelluksen oppimiskynnystä tuomalla käyttäjille saatavilla oleva ohjemateriaali välittömästi.

Tämän raportin toisessa kappaleessa esitellään työn alustana toimiva Puuvirta-oppimisympäristö pääpiirteittäin. Kolmas kappale keskittyy työssä käytetyn kehitysympäristön ja teknologioiden esittelyyn.

Neljännessä kappaleessa käsitellään työn suunnittelua; siinä käydään läpi Puuvirta-asiakasohjelmiston rakenne sekä suunnittelussa huomioon otetut vaatimukset ja rajoitukset. Samassa kappaleessa esitellään myös työssä toteutettu ratkaisu ja käydään läpi suunnittelun aikana hylättyjä ratkaisuja.

Viides kappale käy läpi työn toteutuksen vaiheet. Raportin lopussa pohditaan opinnäytetyön tuloksia ja työprosessin kulkua.

2 PUUVIRTA-OPPIMISYMPÄRISTÖ

Puuvirta-oppimisympäristö kehitettiin Tampereen Ammattikorkeakoulun ja Tampereen teknisen yliopiston hypermedialaboratorion (IISLab / Intelligent Information Systems Laboratory) yhteistyönä MetsäOnline -hankkeessa vuosina 2008–2011.

Hankkeen tavoitteena oli kehittää puunhankinnan oppimisympäristö, jonka avulla puuntoimitusketjun opetuksen laatua ja tehokkuutta voitaisiin parantaa. Oppimisympäristön avulla opetusta voidaan tukea laatimalla erilaisia opetusskenaarioita, sekä luomalla aidon kaltaisia alan ongelmia, joita opiskelijat voivat ratkoa. (Metsä Online –dokumentaatio 2011)

Puuvirta-oppimisympäristö kehitettiin ammattikorkeakouluille, mutta sen sisältö soveltuu myös ammattikouluille ja yliopistoille. Tällä hetkellä oppimisympäristö on opetuskäytössä Tampereen, Mikkelin ja Rovaniemen ammattikorkeakouluissa sekä Helsingin yliopiston metsätieteiden laitoksella. (Puuvirta –projektin sähköpostit 2012)

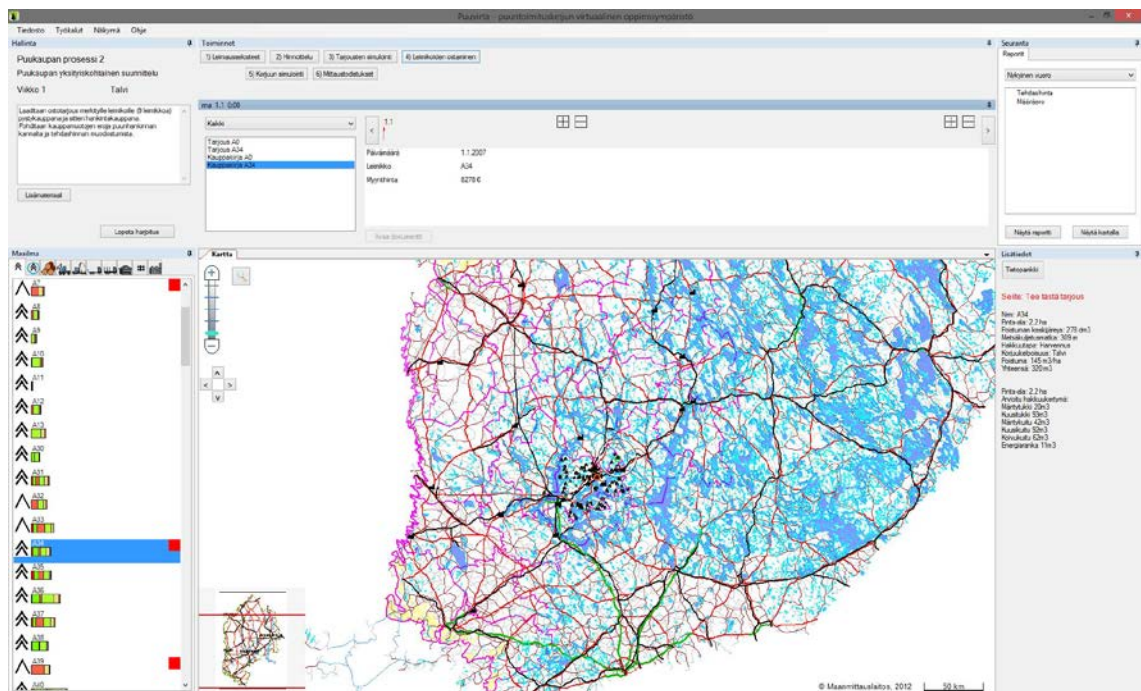
Oppimisympäristö kattaa puunhankinnan eri osavaiheet (puukauppa, korjuu, kuljetus), joita voidaan käsitellä yksityiskohtaisesti tai laajempina ketjuna, joka keskittyy kuvaamaan eri vaiheiden riippuvuussuhteita. Näiden lisäksi oppimisympäristö käsittelee myös hankinnan suunnittelua, joka keskittyy suunnitteluprosessin esittelyyn sekä painottaa aiheelle keskeisten käsitteiden oppimista.

Puuvirta-asiakasohjelmiston avulla opettajat voivat laatia erilaisia harjoituksia, joita opiskelijat voivat suorittaa yksin tai ryhmätöinä riippuen harjoituksen tyypistä.

Harjoituksia suorittaessaan opiskelijoilla on käytössä ns. virtuaalimaailma omine tietoineen ja karttoineen, sekä monipuolisia toimintoja, joiden avulla harjoituksissa voidaan edetä sekä saatavilla olevia tietoja voidaan tulkita ja simuloida.

Kuvassa 1 esitellään Puuvirta-asiakasohjelmiston pääikkuna, joka sisältää yllä mainittuja toimintoja.

Oppimisympäristön jatkokehityksen aikana tein myös muutamia päivityksiä pääikkunassa näkyvään karttaan: selkeämpi aluerajaus, korjauksia kartan värien kustomointiin, sekä korjasin sovelluksessa esiintyneen virheen, jossa tietyssä tapauksessa pääikkunan vasemmassa reunassa sijaitsevassa maailma –valikossa valittu objekti ei näkynyt valittuna kartalla.



KUVA 1. Puuvirta-asiakasohjelmiston käyttöliittymä

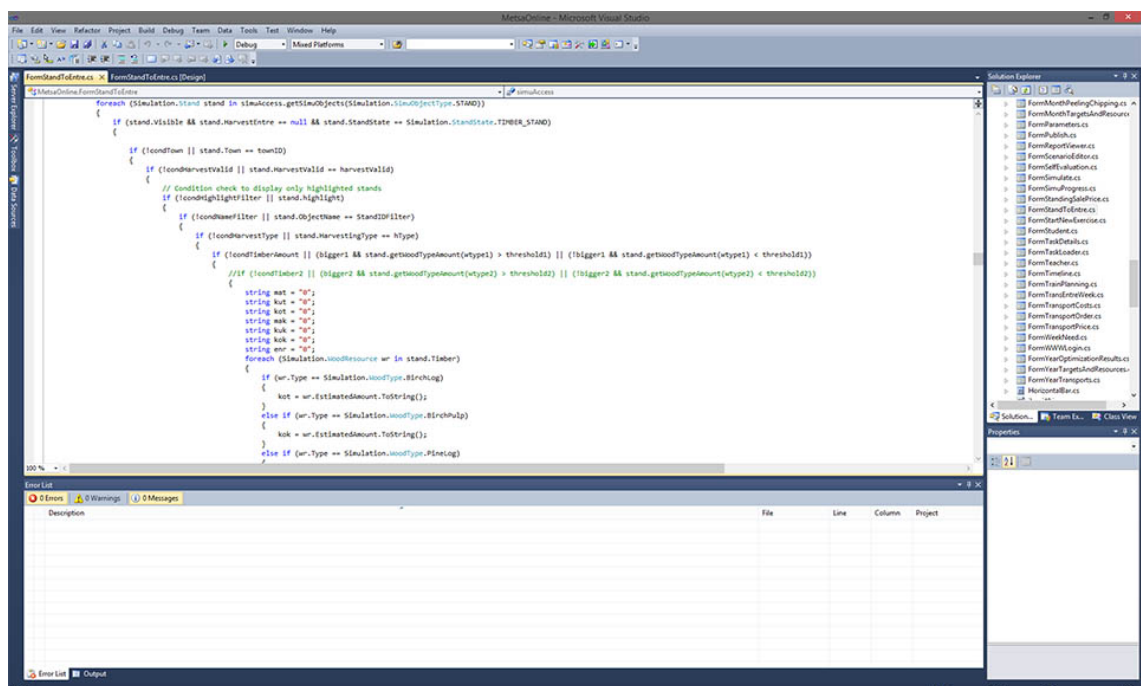
3 TYÖKALUT JA TEKNOLOGIA

Tässä kappaleessa käydään läpi lyhyesti työssä käytetty kehitysympäristö sekä teknologia.

3.1 Visual Studio -kehitysympäristö

Visual Studio on Microsoftin kehittämä integroitu kehitysympäristö, jonka avulla kehittäjä voi luoda erilaisia sovelluksia kuten työpöytäsovelluksia, verkkosovelluksia sekä mobiilisovelluksia. Kehitysympäristö sisältää mm. työkaluja koodin editoimiseen, käyttöliittymien luontiin, koodin kääntämiseen sekä virheiden etsintään. (Microsoft Developer Network – Visual Studio)

Kuvassa 2 esitellään Visual Studion projektinäkömä, jossa on näkyvissä avoinna oleva projekti sekä työkaluja koodin editoimiseen. Kuvassa työn alla on oppimisympäristön jatkokehityksen aikana toteutettu hakutoiminto varannon jako yrittäjille –ikkunaan.



KUVA 2. Visual Studio 2010 Professional -kehitysympäristön käyttöliittymä

Tätä opinnäytetyötä tehdessä käytettiin ensin Visual Studio 2008 -versiota. Kuitenkin projektissa tehtyjen muutosten vuoksi siirryttiin myöhemmin käyttämään Visual Studio 2010 Professional -versiota.

3.2 .NET Framework -alusta

.NET Framework on Microsoftin kehittämä ohjelmistokomponenttikirjasto, joka mahdollistaa monien eri ohjelmointikielien yhteentoimivuuden. Tuetuista ohjelmointikielistä suosituimpien joukkoon kuuluu myös Puuvirrassa käytetty C#.

.NET -alustalle suunnattujen sovellusten kehitysympäristönä toimii Microsoftin Visual Studio. Niiden pääasiallinen kohdealusta on Microsoft Windows –käyttöjärjestelmä; kuitenkin muille alustoille, kuten Linux ja Mac OS, löytyy kolmannen osapuolen toteuttamia ratkaisuja. (Mono -projekti 2012)

3.3 C#-kieli

C# (C Sharp) on Microsoftin kehittämä oliopohjainen ohjelmointikieli, jonka kehitys alun perin tapahtui .NET Frameworkin kehityksen yhteydessä. C# on erityisesti suunnattu .NET Frameworkia hyödyntävien sovellusten kehittämiseen.

Microsoftin tavoitteena oli kehittää moderni ohjelmointikieli, joka on helppokäyttöinen mutta tehokas. (Introduction to the C# language and the .NET Framework 2013)

3.4 HTML-kieli ja CSS-tyyliohje

HTML eli Hypertext Markup Language on merkintäkieli, jota käytetään verkkosivujen rakenteen kuvaamisessa. Kun verkkoselain lukee HTML-dokumenttia, se tulkitsee dokumentin sisällön ja luo sen perusteella käyttäjälle näkyvän verkkosivun.

Yksinkertaistettuna voitaisiin sanoa, että HTML-dokumentit koostuvat tunnisteista, jotka sisältävät rakenteellisia, esityksillisiä ja hypertekstuaalisia elementtejä. Rakenteelliset elementit määrittävät elementin tyypin/tarkoituksen ja esitykselliset miltä sen tulisi näyttää selaimessa. Hypertekstuaaliset elementit toimivat linkkeinä muihin dokumentteihin.

Kuvassa 3 esitellään esimerkki HTML-dokumentin rakenteesta, joka sisältää tunnisteita ja erityyppisiä elementtejä. Esimerkki rakenteesta on <head> -tunnisteen sisältö, joka sisältää kolme rakenteellista elementtiä ja yhden hypertekstuaalisen elementin.

Elementti <title> on rakenteellinen elementti, joka määrittelee dokumentin otsikon. Seuraava elementti on <link>, joka toimii hypertekstuaalisena viittauksena .css tiedostoon.

Viimeiseksi <head> -tunniste sisältää kaksi rakenteellista <script> -elementtiä. Ensimmäinen <script> -elementti sisältää viittauksen erilliseen dokumenttiin, joka sisältää komentosarjan. Toinen <script> -elementti sisältää lyhyen komentosarjan.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-GB">
3   <head>
4     <title>Puvvirta - Käyttöohje</title>
5     <link rel="stylesheet" type="text/css" href="style.css" media="screen"/>
6     <script type="text/javascript" src="navmenu.js"></script>
7     <script>
8       function navcheck()
9       {
10        document.getElementById("mainlink10").checked=true
11        }
12    </script>
13  </head>
14  <body>
15    <div id="container">
16      <div id="header">
17        <a href="http://metsonline.projects.tamk.fi"></a>
18      </div>
19      <div id="content-container">
20        <div id="content">
21          <!-- Page Content Here -->
22          <p>Esimerkki</p>
23        </div>
24        <script type="text/javascript">show(navmenu)</script>
25        <script>navcheck()</script>
26      <div id="footer">
27        
28      </div>
29    </div>
30  </body>
31 </html>

```

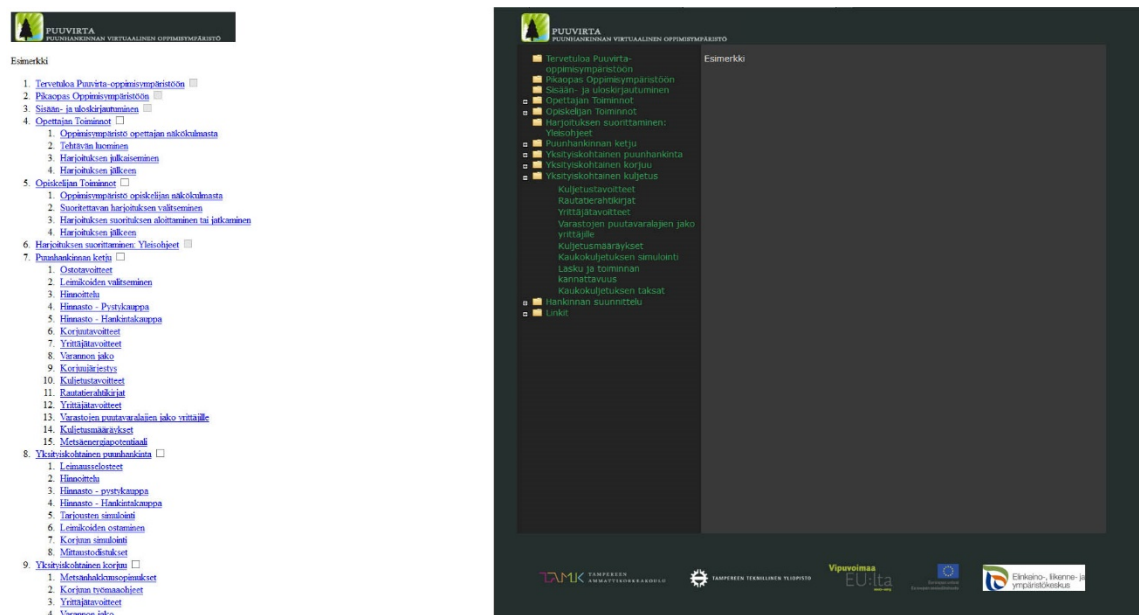
KUVA 3. Esimerkki HTML-dokumentin rakenteesta

Kuitenkin nykyään on suosittua käyttää erillistä CSS (Cascading Style Sheets) -tyyliohjetta HTML-dokumenttien lisänä. Sen tarkoituksena on mahdollistaa erittely, jossa HTML (tai muulla merkintäkielellä toteutettu) -dokumentti käsittelee vain sen sisältämät elementit ja jossa CSS-tyyliohje määrittelee dokumentin ulkonäön ja kuinka verkkoselaimen tulisi taittaa sivusto.

Erillisen .css-tyylitiedoston (tai tiedostojen) käyttö selkeyttää ja nopeuttaa kehitysprosessia huomattavasti. Yhdellä tyylitiedostolla pystytään määrittelemään kokonaisuuden ulkonäkö sen sijaan, että ulkonäön määrittely tehtäisiin jokaiseen HTML-dokumenttiin erikseen.

Kuvassa 4 esitellään, miltä kuvan 3 HTML-dokumentti näyttää verkkoselaimessa ilman CSS-tyyliohjetta ja CSS-tyyliohjeen kanssa.

Vasemmalla puolella kuvaa HTML-dokumentti on avattu verkkoselaimessa ilman CSS-tyyliohjetta, jolloin jokainen elementti näkyy verkkoselaimessa sen tulkitsemassa järjestyksessä ja niiden oletusmuodossa. Kuvan oikealla puolella HTML-dokumentti on avattu verkkoselaimessa CSS-tyyliohjeen kanssa, jonka avulla selain osaa tulkita, missä järjestyksessä ja miten HTML-dokumentin sisältö tulisi näyttää käyttäjälle.



KUVA 4: Esimerkki CSS-tyyliohjeen vaikutuksesta HTML-dokumentin ulkoasuun verkkoselaimessa.

3.5 JavaScript-kieli

JavaScript on tulkittava komentosarjakieli, joka alun perin kehitettiin pääasiassa erilaisten selainskriptien luontiin verkkoympäristössä. Niiden avulla verkkosivustoille voitiin luoda erilaisia toimintoja, joita ei ollut mahdollista toteuttaa pelkän HTML -kielen avulla.

JavaScriptin suosio on kasvanut vuosien mittaan ja nykyään sille löytyy tuki suurimmasta osasta moderneja verkkoselaimia ja sen käyttö on myös yleistynyt verkkoympäristöjen ulkopuolella. JavaScript-kieltä hyödynnetään esimerkiksi sovellusten ominaisuuksien luonnissa (esim. SimCity -pelin (2013) käyttöliittymä on toteutettu JavaScriptin avulla) ja erilaisten widgettien luonnissa. Widgetit ovat minisovelluksia, jotka ovat yleensä

suunniteltu toteuttamaan jokin tietty, mutta pienimuotoinen tehtävä. Esimerkkinä widgetin toiminnasta voitaisiin mainita minisovellus, jonka avulla käyttöliittymään toteutetaan muistutus päivän tapaamisista, jotka ovat merkitty erilliseen kalenterisovellukseen.

Opinnäytetyössäni hyödynsin JavaScript-kieltä ohjemateriaalin navigointipalkin ja sivukohtaisten tunnisteiden luonnissa.

Kielenä JavaScriptin perusteet ovat helposti omaksuttavissa ja sen tulkittava luonne mahdollistaa nopean ja helpon tavan testata erilaisia skriptejä/ohjelmia verkkoselaimen avulla. (W3Schools JavaScript tutorials)

4 TYÖN SUUNNITTELU

Tässä kappaleessa käydään läpi työn suunnittelun kannalta oleelliset tiedot pääpiirteittäin sekä käsitellään työhön valittua ratkaisua ja perusteellaan, miksi kyseinen ratkaisu valittiin.

4.1 Puuvirta-oppimisympäristön rakenne

Puuvirta-oppimisympäristö koostuu kahdesta pääosiesta: loppukäyttäjien asiakasohjelmistosta sekä palvelimesta, joka toimii tietovarastona ja kattaa käyttäjien hallintaan tarvittavat työkalut.

Puuvirta-asiakasohjelmisto on .NET -sovellus, joka on pääasiassa toteutettu C#-ohjelmointikielellä. Asiakasohjelmistossa on käytetty TTY:n hypermedialaboratorion luomaa työkalupohjaista oppimisympäristökehystä, joka on toteutettu C++/CLI-ohjelmointikielellä.

Asiakasohjelmiston käyttöliittymä on toteutettu graafisten käyttöliittymien luontiin tarkoitetulla Microsoftin kehittämällä Windows Forms (winforms) –rajapinnalla. Käyttöliittymän suunnittelusta ja toteutuksesta vastasi TTY:n hypermedialaboratorio.

4.2 Vaatimukset ja rajoitukset

Työn toteutus jaettiin kahteen osaan: Asiakasohjelmistoon toteutettavaan ohjetoimintoon ja ohjelmamateriaalille luotavaan runkoon. Työn toteutukselle asetettiin seuraavat vaatimukset ja rajoitukset:

- 1) Ohjelmamateriaalin tulisi olla helposti muokattavissa ja siihen tehtävien muutosten tulisi olla toteutettavissa ilman syvempää ohjelmistotekniikan tuntemusta.
- 2) Ohjelmamateriaalin tulisi sisältää sisällysluettelo ja ohjelmamateriaaliin pitäisi pystyä lisäämään linkkejä.
- 3) Ohjelmamateriaalin tulisi sijaita palvelimella, mutta tarvittaessa se voitaisiin myös sisällyttää sovelluksen asennuspakettiin.

Tilannekohtaisen ohjetoiminnon kutsumiseen määriteltiin F1-näppäin, joka on yleisesti käytetty näppäin ohjetoiminnoille. Näiden vaatimusten lisäksi ohjetoiminnon koodin toteutuksessa tulisi ottaa huomioon mahdollinen jatkokehitys muiden opiskelijoiden toimesta.

4.3 Työssä käytetty ratkaisu

Ohjemateriaalille säädettyjen vaatimusten perusteella sen toteutukseen valittiin HTML-kieli, jonka avulla ohjemateriaalille rakennettaisiin runko. Yhtenä työn vaatimuksista oli ohjemateriaalin helppo muokkaus ja siitä johtuen rungon ulkoasun asetukset ulkoistettiin erilliseen CSS-tyyliohjeeseen.

Mikäli ohjemateriaalin sisältöön halutaan tehdä muutoksia, niiden tekeminen onnistuu `<p>`, `
`, `<h1>`, `<h2>`, `` ja `<a href>` peruselementtien avulla. Muutosten tekemistä tukemaan luotiin erillinen HTML-dokumentti, joka toimii esimerkkinä rungon rakenteesta ja sisältää ohjeita elementtien käytöstä.

Ohjemateriaalin sisällysluettelon toteutukseen valittiin JavaScript -kieli, jonka avulla voidaan toteuttaa HTML-dokumentista erillään oleva sisällysluettelo. Erillisen sisällysluettelon etuna on ylläpidon helppous, sillä siihen tehtävät muutokset ovat näkyvissä heti kaikilla ohjemateriaalin sivuilla.

JavaScript -kieli valittiin sisällysluettelon toteutukseen, koska se toimii riippumatta siitä, luetaanko HTML-dokumenttia paikalliselta levyltä tai verkkosivustolta.

Ohjemateriaali on tarkoitettu ns. sisäiseksi dokumentaatioksi, joten JavaScriptin valintaa tässä tapauksessa voidaan perustella myös sillä, että HTML-dokumentin luonnissa ei tarvitse ottaa huomioon hakukoneoptimointia ja kohdeympäristössä voidaan varmistaa, että JavaScriptin suoritus on sallittua ohjemateriaalin sijainnista.

Asiakasohjelmiston ohjetoiminnon toteutukseen valittiin System.Diagnostics -nimiavaruus, jonka Process -luokan avulla päästään käsiksi erilaisiin prosessikomentoihin.

Ohjetoiminnon asetusten toteuttamiseen valittiin XML-kielellä toteutettava asetustiedosto, jonka tulkitsemiseen tarvittavat toiminnot voidaan toteuttaa System.Xml -nimiavaruuden luokkien avulla.

4.4 Hylätyt ratkaisut

Työn suunnittelun aikana tutkittiin monia eri tapoja ja teknologioita, joiden avulla tilannekohtainen ohjetoiminto ja ohjemateriaali olisi voitu toteuttaa. Tässä osiossa käsitellään muutamia ratkaisuja, jotka hylättiin suunnittelun aikana.

4.4.1 HelpProvider -luokka

HelpProvider -luokka on Windows Forms -rajapinnan tarjoama komponentti, jota voidaan käyttää sovelluksen ohjetoimintojen sekä muiden aputoimintojen luonnissa.

HelpProvider -luokka tukee HTML- sekä CHM -tiedostoja, joiden avulla tilannekohtaisen ohjetoiminnon ohjemateriaali voidaan toteuttaa. Tämän lisäksi HelpProvider -luokan avulla voidaan toteuttaa tooltip-tyyppisiä vihjeitä.

Suunnittelun aikana ohjetoiminnon ensimmäinen prototyyppi toteutettiin käyttäen HelpProvider -luokkaa. Myöhemmässä vaiheessa kehitystä HelpProvider -luokka korvattiin Process -luokan Start -metodilla, joka takaa paremman yhteensopivuuden eri tiedostomuotojen kanssa.

4.4.2 PDF- ja docx -tiedostot

Suunnitteluvaiheen alkupuolella tutkittiin, olisiko ohjemateriaalin runkona mahdollista käyttää Microsoft Word:lla tuotettuja .docx -tiedostoja tai Adoben kehittämää .pdf -tiedostomuotoa.

Idea tiedostomuotojen tutkimiseen lähti ohjemateriaalin sisällöntuottajien suunnalta, sillä niiden luominen ja muokkaaminen olisi erittäin yksinkertaista tuttujen sovellusten avulla. Ongelmaksi molempien tiedostomuotojen kohdalla osoittautui tilannekohtaisen ohjeen toteutus, jossa dokumentti tarvitsisi avata tietystä kohdasta riippuen siitä, mistä sovelluksen ikkunasta ohjetoimintoa kutsuttiin.

Ominaisuuden toteutus olisi mahdollista, mutta se olisi hyvin hankalaa ja sen toimivuutta jokaisessa oppimisympäristön käyttökohteessa ei voitaisi taata ellei .docx- ja .pdf -formaatteja lukevista sovelluksista olisi käytössä täsmälleen samoja versioita.

4.4.3 CHM -tiedosto

CHM eli Microsoft Compiled HTML Help -tiedostomuoto on tarkoitettu ohjemateriaalin luontiin. CHM-dokumentti koostuu HTML-sivuista, sisällysluettelosta ja navigointiin tarkoitetuista työkaluista.

Ohjemateriaalin ominaisuuksien osalta CHM (sekä uudemmat CHM:n korvanneet Microsoft help -tiedostomuodot) täyttävät sille asetetut vaatimukset sisällön osalta. Ongelmaksi CHM -tiedostojen kohdalla kuitenkin muodostui niiden kehitys ja päivitys.

CHM -tiedostojen kehitykseen käytetään erillisiä työkaluja, joista helppokäyttöisimmät ovat maksullisia ja ammattilaiskäyttöön suunnattuja. Ohjemateriaalin päivityksen kannalta erillisen työkalun vaatiminen tuottaisi turhaa työtä; etenkin pienien muutosten (kuten linkkien lisäämisen yms.) osalta.

5 TYÖN TOTEUTUS

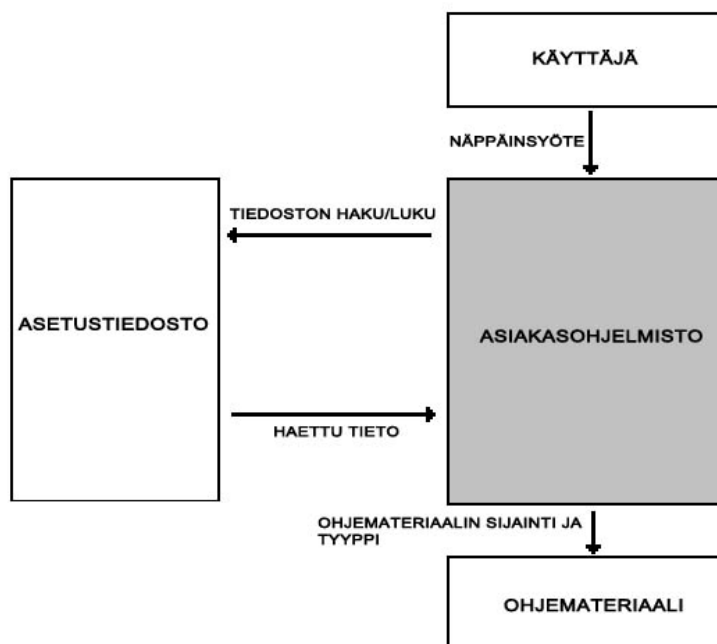
Tässä luvussa käydään läpi työn toteutuksen vaiheet ja selitetään, kuinka työssä luodut komponentit toimivat keskenään.

5.1 Tilannekohtainen ohjetoiminto

Tilannekohtainen ohjetoiminto jakautuu kolmeen pääosiioon: Asetustiedostoon, ohjetoimintoon ja ohjemateriaaliin.

Kaaviossa 1 havainnollistetaan ohjetoiminnon pääosioiden keskinäistä vuorovaikutusta. Käyttäjän painaessa näppäimistön F1-näppäintä, asiakasohjelmisto rekisteröi tulevan syötteen ja käynnistää asetustiedoston lukuprosessin. Lukuprosessi etsii asiakasohjelmiston asennuskansiossa sijaitsevan asetustiedoston ja lukee asetustiedoston sisältämät ohjemateriaalin parametrit.

Luettujen tietojen perusteella asiakasohjelmisto avaa käyttäjälle asetustiedostossa määritellyn ohjemateriaalin, sille määritellyllä oletusohjelmistolla (esim. HTML-dokumentit niille määritellyllä oletusselaimella.)



KAAVIO 1: Tilannekohtaisen ohjetoiminnon pääosioiden vuorovaikutus

5.1.1 XML -asetustiedosto

XML-kielellä luodun asetustiedoston tarkoituksena on mahdollistaa ohjemateriaalin räätälöinti. Asetustiedosto koostuu kolmesta pääosiosta: Ohjetyypin määrittämisestä, ohjemateriaalin sijainnista ja tilannekohtaisen ohjemateriaalin sijainneista. Kuvassa 5 esitetään esimerkki XML –asetustiedoston edellä kuvatussa rakenteesta.

```

1 <?xml version="1.0" encoding="iso-8859-1" ?>
2 <helpconf>
3
4 <!--Toggle for the context sensitive help: 0 to Disable, 1 to Enable-->
5 <ContextSensitiveHelp>1</ContextSensitiveHelp>
6
7 <!--If the context sensitive help is disabled, add the help file path or URL below-->
8 <helpfile>InsertLocalPathHere</helpfile>
9
10 <!--Page specific path/URL for context sensitive help-->
11 <BuyApprovalURL>http://metsaonline.projects.tamk.fi/manual/formbuyapproval.html</BuyApprovalURL>
12 <BuyOfferSimuURL>http://metsaonline.projects.tamk.fi/manual/formbuyoffersimu.html</BuyOfferSimuURL>
13 <BuyPricingURL>http://metsaonline.projects.tamk.fi/manual/formbuypricing.html</BuyPricingURL>
14 <BuyPricing_yURL>http://metsaonline.projects.tamk.fi/manual/formbuypricing_y.html</BuyPricing_yURL>
15 <BuySelectionURL>http://metsaonline.projects.tamk.fi/manual/formbuyselection.html</BuySelectionURL>
16 <BuyStandInfoURL>http://metsaonline.projects.tamk.fi/manual/formbuystandinfo.html</BuyStandInfoURL>

```

KUVA 5: Esimerkki XML –asetustiedoston sisällöstä

Ohjetyypin määrittämisen avulla käyttäjä voi määrittellä onko asiakasohjelmistossa käytössä tilannekohtainen ohjetoiminto vai käyttääkö sovellus yhdestä tiedostosta koostuvaa ohjemateriaalia.

Mikäli käyttäjä valitsee käytettäväksi vain yhden ohjetiedoston, sen sijainti voidaan määrittää asetustiedostossa.

Asetustiedosto sisältää myös muokattavan listan tilannekohtaisten ohjemateriaalien sijainneista. Listan nimeämispolitiikkana käytettiin asiakasohjelmiston sisäisiä englanninkielisiä lomakenimiä.

5.1.2 Toiminnon kehittäminen

Toiminnon kehittäminen alkoi XML-lukijan luonnilla, sen kehityksessä hyödynnettiin .NET Frameworkin System.Xml nimiavaruuden tarjoamaa XmlTextReader-luokkaa. XmlTextReader-luokan tukena käytettiin XmlNodeType-viittauksia, joiden avulla XML-lukija pystyy tulkitsemaan miten sen tulisi toimia asetustiedoston sisältämien erityyppisten XML-elementtien kanssa.

Tarkoituksena oli luoda lukija, joka pystyisi tulkitsemaan asetustiedostosta tarvittavat asetukset ja väliaikaisesti tallentamaan ne muistiin string-muotoisena muuttujana.

Asetustiedoston lukemisen lisäksi tarvittaisiin ominaisuus, joka avaisi asetustiedostossa määritellyn ohjemateriaalin. Tämä ominaisuus toteutettiin System.Diagnostics -nimiavaruuden Process -luokan Start -metodilla.

Process.Start -metodin avulla sovellus voi käynnistää käyttöjärjestelmästä löytyviä prosesseja. Määrittelemällä metodin parametriksi ohjemateriaalin sijainnin ja tiedoston nimen, voidaan metodin avulla kutsua sille tiedostotyyppille määritettyä oletussovellusta.

Esimerkiksi verkko-osoitteet ja HTML-dokumentit avautuvat käyttöjärjestelmän oletusselaimella.

XML-lukijasta ja Process.Start -metodista luotiin silmukoita, joiden avulla asiakasohjelmisto pystyy tulkitsemaan määritellyt asetukset ohjemateriaalille ja sen perusteella avaamaan oikean tyyppisen ohjemateriaalin käyttäjälle.

Tilannekohtaisen ohjetoiminnon kutsuminen tapahtuu painamalla F1-näppäintä näppäimistöllä. Tämän toteuttamiseksi jokaiseen sovelluksen lomakkeista tehtiin kaksi muutosta; toimintoikkunat muutettiin tunnistamaan näppäimistöltä saapuva syöte ja niihin lisättiin tapahtuma, joka aktivoituu kun F1-näppäintä painetaan ikkunan ollessa aktiivisena.

5.2 Ohjemateriaalin runko

Ohjemateriaalin runko koostuu kolmesta osasta: HTML-dokumentista, CSS-tyyliohjeesta ja JavaScript -kielellä luodusta navigointipalkista.

HTML-dokumentti toimii rungon perustana, se sisältää dokumentin rakenteen muodostavia elementtejä sekä ohjemateriaalin sisällön. HTML-dokumentti sisältää myös viittauksen CSS-tyyliohjeeseen sekä `<script>` -elementtejä, joiden avulla HTML-dokumentissa viitataan erilliseen JavaScript -kielellä toteutettuun koodiin.

JavaScript -kielellä toteutettu navigointipalkki kirjoitetaan HTML-dokumenttiin, kun sitä kutsutaan HTML-dokumentissa olevan `<script>` -elementin avulla. CSS-tyyliohjeessa määritellään dokumentin ulkoasu ja kuinka selaimen tulisi taittaa dokumentti.

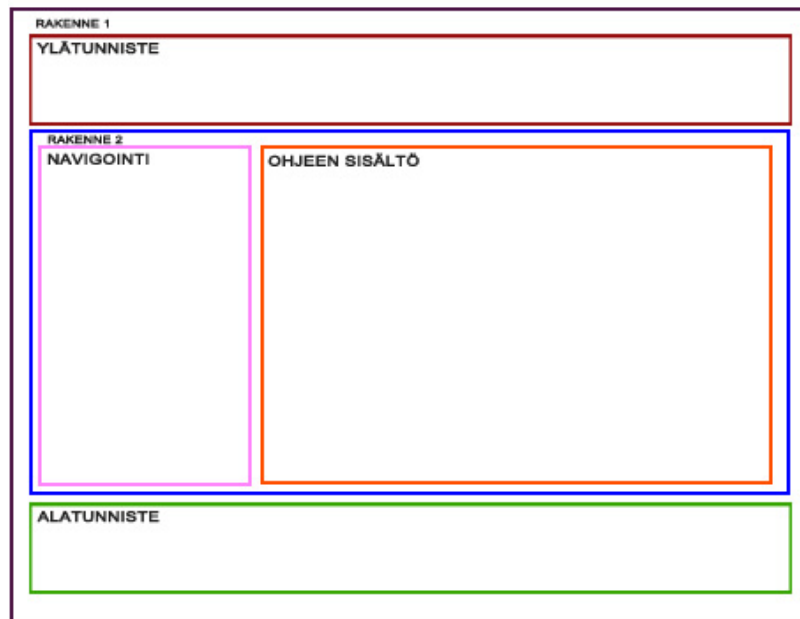
5.2.1 HTML-dokumentin luonti

HTML-dokumentin luonnissa on pyritty pitämään sen sisältö selkeänä ja helposti ymmärrettävänä myös henkilöille, joille HTML-kieli ei ole ennestään tuttu. Dokumentti keskittyy rakenteen ja sisällön määrittämiseen, eikä siinä käytetä dokumentin ulkoasua määritteleviä esityksellisiä elementtejä.

Dokumentti alkaa dokumentin tyyppin määrittelyllä ja se sisältää HTML-dokumenteille ominaisen rakenteen joka koostuu `<html>` -tunnisteen sisään rakennetuista `<head>` ja `<body>` -tunnisteista.

Dokumentin `<head>` -tunnisteessa määritellään dokumentin otsikko ja se sisältää myös kaksi `<script>` -elementtiä, joista ensimmäisen avulla dokumentissa viitataan ulkoiseen JavaScript -tiedostoon ja toisen avulla määritetään navigointipalkin sivutunniste. Tunniste sisältää myös viittauksen CSS-tyyliohjeeseen, joka vastaa dokumentin ulkoasun määrittelystä.

Dokumentin `<body>` -tunniste sisältää sisäkkäin rakennettuja `<div>` -elementtejä, jotka määrittelevät dokumentin rakenteen. Elementeistä muodostuu tyypillinen verkkosivustoissa käytetty rakenne, joka sisältää ylätunniste, sisältö ja alatunniste -osiot. Kaaviossa 2 havainnollistetaan edellä kuvattua dokumentin rakennetta.



KAAVIO 2: Ohjemateriaalin rakenne `<div>` -elementteinä.

Ylätunniste ja alatunniste -osiot sisältävät `` -kuvaelementtejä, joista osa toimii myös linkkeinä. Sisältö -osio kattaa ohjemateriaalin sisällön sekä se sisältää `<script>` -elementin, jonka avulla kutsutaan JavaScript -navigointipalkin piirtofunktiota.

Asiakasohjelmiston jokaiselle toimintoikkunalle on luotu sille tarkoitettu HTML-dokumentti, joka sisältää toimintoikkunan tilannekohtaisen ohjemateriaalin.

5.2.2 JavaScript-navigointipalkin ja sivutunnisteen kehitys

Ohjemateriaalissa sisällysluettelona toimiva navigointipalkki on JavaScript -kielen avulla luotu taulukko, joka sisältää HTML-elementtejä. Navigointipalkki rakentuu yhdestä `<div>` -elementistä, jonka sisältö koostuu `` ja `` -listaelementeistä, jotka sisältävät `<a href>` -linkkielementtejä.

Navigointipalkki sisältää myös funktion, jonka avulla taulukon sisältö voidaan piirtää siihen HTML-dokumenttiin, josta sen skriptiä kutsutaan. Kuvassa 6 esitellään navigointipalkin koodi pääpiirteittäin, sekä navigointipalkin piirtofunktio.

```

var navmenu = new Array();

navmenu[0] = '<div id="navigation">';
navmenu[1] = '<ol class="navtree">';
.
.
.
navmenu[108] = '</div>';

function show(i)
{
  for (x in i)
  {
    document.write(i[x]+'\\n')
  }
}

```

KUVA 6. Navigointipalkin sisältö pääpiirteittäin ja sen piirtofunktio.

JavaScript –kielen avulla ohjemateriaalille kehitettiin myös sivutunniste, joka toimii yhteistyössä CSS-tyyliohjeen kanssa. Ohjemateriaalissa olevaan navigointipalkkiin on toteutettu kansiorakenne CSS-tyyliohjeen avulla, sivutunnisteen tarkoitus on kertoa selaimelle mihin kansioon kukin ohjemateriaalin sivuista kuuluu.

Sivutunnisteen toimintaa voidaan havainnollistaa seuraavalla esimerkillä: käyttäjän avatessa ”Ostotavoitteet” –ikkunan ohjemateriaalin, sivutunniste kertoo selaimelle JavaScriptin `document.getElementById` –metodin avulla, että kyseessä on ohjemateriaalin sivu, joka kuuluu ”Hankinnan suunnittelu” –osioon. Tästä johtuen ohjemateriaalin avautuessa selain on myös laajentanut näkyviin kaikki ”Hankinnan suunnittelu” –kansioon kuuluvat linkit navigointipalkissa.

5.2.3 CSS-tyyliohjeen luonti

CSS-tyyliohje kehitettiin vastaamaan projektin tavoitetta siitä, että ohjemateriaali olisi helposti muokattavissa ja sen muokkauksessa ei tarvittaisi laajempaa tietoa ohjelmoinista. Tyyliohjeeseen on ulkoistettu kaikki ohjemateriaalin ulkoasusta vastaavat määrittelyt. Näihin lukeutuvat mm. sivun taiton asetukset, sivuilla käytetyt värit, fonttien määrittelyt ja rivien tasaus/välit.

Ohjemateriaalin navigointipalkille on myös toteutettu CSS:n avulla kansiorakenne, jonka tarkoituksena on helpottaa navigointia käyttäjän näkökulmasta sekä helpottaa uusien sivujen lisäämistä kehittäjän näkökulmasta.

Tyyliohjeen luonnissa on pyritty käyttämään CSS-tekniikoita, jotka ovat yhteensopivia kaikkien modernien verkkoselainten kanssa.

Kuvassa 7 esitellään ohjemateriaalin ulkoasu käyttäjälle näkyvässä muodossa. Kuva sisältää alustavat yleisohjeet harjoitusten suorittamisesta Puuvirta –oppimisympäristössä.

PUUVIRTA
PUUNHANKINNAN VIRTUAALINEN OPPIMISYMPÄRISTÖ

- Tervetuloa Puuvirta-oppimisympäristöön
- Pikaopas Oppimisympäristöön
- Sisään- ja uloskirjautuminen
- ▣ Opettajan Toiminnot
- ▣ Opiskelijan Toiminnot
- Harjoituksen suorittaminen: Yleisohjeet
- ▣ Puunhankinnan ketju
- ▣ Yksityiskohtainen puunhankinta
- ▣ Yksityiskohtainen korjuu
- ▣ Yksityiskohtainen kuljetus
- ▣ Hankinnan suunnittelu
- ▣ Linkit

Harjoituksen suorittaminen: Yleisohjeet

Suoritusnäkymä:

Suoritusnäkymän yläosan vasemmassa reunassa näkyy harjoituksen perustiedot, sekä painikkeet harjoituksen tallentamiseen ja lopettamiseen. Yläosan keskiosassa ovat toimintopainikkeet, sekä syntyneet dokumentit. Oikeassa reunassa on lista suorituksen tietoja koostavista raporteista. Raportin voi avata valitsemalla **Näytä raportti**.

Suoritusnäkymän alaosassa näytetään oppimisympäristön virtuaalimaailma. Keskellä on kartta, jota voi zoomata ja siirtää. Kartan vasemmalla puolella on tyyppinsä mukaan välilehdille jaettu listaus kaikista virtuaalimaailman kohteista. Kohteen valitsemalla se näytetään kartalla, ja kohteen tarkat tiedot oikeassa reunassa.

Harjoituksen suorittaminen:

- Harjoitus suoritetaan viikko ja vaihe kerrallaan. Tietty viikko ja vaihe suoritetaan yläreunan toimintopainikkeiden avulla.
- Toiminnot käydään läpi numerojärjestyksessä.
- Kaikki toiminnot käydään läpi.
- Jos palaat uudestaan jo suoritettuun toimintoon, on suositeltavaa käydä uudelleen läpi kaikki väliin jäävät toiminnot.
- Toimintopainikkeista avautuu uusi ikkuna, jossa itse toiminto suoritetaan.
- Jatkaaksesi seuraavaan vaiheeseen, paina Tallenna harjoitus- painiketta.

Harjoituksen suorittamisen yhteydessä käytetään monissa yhteyksissä vihreä/keltainen /punainen-korostuksia, joiden merkitys on yleensä:

- *vihreä*: tiedot oikein, ei korjattavaa
- *keltainen*: tiedot lähes oikein, (mahdollisesti) hieman korjattavaa
- *punainen*: tiedot väärin tai niissä on suuri poikkeama, korjattavaa

TAMK TAMPEREEN AMMATTIKORKEAKOULU

TAMPEREEN TEKNILLINEN YLIOPISTO

Vipuvoimaa EU:lta

EUROOPAN UNIONIN EUROOPAN MAAILMANLAATU

Elinkeino-, liikenne- ja ympäristökeskus

KUVA 7. Ohjemateriaalin ulkoasu verkkoselaimessa

6 POHDINTA

Opinnäytetyön ja Puuvirta –oppimisympäristön parissa työskentely tarjosi mielenkiintoisen näkökulman opetukseen käytettävistä virtuaaliympäristöistä.

Ennen kuin aloitin opinnäytetyöni tekemisen, olin jo työskennellyt Puuvirta – oppimisympäristön parissa muutaman kuukauden. Tuona aikana tutustuin ensin ympäristön toimintaan, sen dokumentaatioon sekä lähdekoodin, jonka jälkeen keskityin lähinnä paikantamaan ja korjaamaan erilaisia virhetilanteita, joita oppimisympäristön käytön aikana oli esiintynyt.

Virheiden korjauksen ohella olin toteuttanut myös muutaman lisäominaisuuden oppimisympäristöön, joka antoi hyvän lähtökohdan opinnäytetyöni toteutukseen.

Tilannekohtaisen ohjetoiminnon kehittäminen oli opettavainen prosessi, jona aikana tutustuin moniin eri tapoihin, joilla ohjetoimintoja toteutetaan ja tutkin mikä olisi paras tapa toteuttaa ominaisuus Puuvirta -oppimisympäristöön.

Ohjemateriaalin pohjan toteutus tarjosi myös omat haasteensa: kuinka se voitaisiin toteuttaa mahdollisimman yksinkertaiseksi tulevien muutosten kannalta? Kuinka kattava dokumentaatio / ohjeistus pohjan muokkaukseen tarvitsisi kirjoittaa? Näiden kysymysten lisäksi oli otettava huomioon käyttäjäystävällisyys sekä muut vaatimukset.

Kokonaisuudessaan projekti oli varsin mielenkiintoinen kokemus, sillä siinä yhdistyi tasapuolinen määrä tutkimista, suunnittelua ja toteutusta.

LÄHTEET

Metsä Online –projektin sisäinen dokumentaatio. 2011.

Puuvirta –projektin sähköpostiviestit. 2012, 2013.
jukka.tohu@tamk.fi, marko.moisio@tut.fi

MetsäOnline –sivusto. 2009. Luettu 20.5.2013.
<http://metsaonline.projects.tamk.fi/>

Introduction to the C# language and the .NET Framework. 2013. Luettu 25.5.2013.
<http://msdn.microsoft.com/en-us/library/vstudio/z1zx9t92.aspx>

Microsoft Developer Network – Visual Studio. 2013. Luettu 26.5.2013.
<http://msdn.microsoft.com/en-us/vstudio/cc136611.aspx>

Mono -projekti. 2012. Luettu 5.6.2013.
<http://www.mono-project.com>

W3Schools JavaScript tutorials. 2013. Luettu 13.6.2013.
<http://www.w3schools.com/js/>