

Saimaan ammattikorkeakoulu  
Tekniikka Lappeenranta  
Tietotekniikka  
Ohjelmistotekniikka

Keijo Leskelä

## **Yrityspeli**

Opinnäytetyö 2011

## **Tiivistelmä**

Keijo Leskelä

Yrityspeli, 40 sivua

Saimaan ammattikorkeakoulu

Tekniikka Lappeenranta

Tietotekniikka

Ohjelmistotekniikka

Opinnäytetyö 2011

Ohjaajat: Lehtori Martti Ylä-Jussila, Saimaan ammattikorkeakoulu, projektipäällikkö Asko Kilpeläinen, Saimaan ammattikorkeakoulu

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa Saimaan ammattikorkeakoulun sosiaali- ja terveystieteiden opiskelijoille Yrityspeli. Yrityspelin tavoitteena on kannustaa ja rohkaista opiskelijoita perustamaan oman alan yritys sekä opettaa yrittäjyyden perusteita.

Työ aloitettiin tutkimalla mahdollisia toteutustekniikoita, opiskelemalla peliohjelmoinnin teoriaa sekä laatimalla toiminnallisen määrittelyn perusteella alustava arkkitehtuuri pelille. Pääasiallisesti tekniikaksi valittiin ASP.NET ja ohjelmointikieleksi C#. Tietokantana toimi MySQL-tietokanta ja tietokantakyselyt tehtiin LINQ-tekniikalla. Peliä suunniteltiin ja toteutettiin pieninä kokonaisuuksina. Kehitysympäristönä toimi Microsoft Visual Studio 2010.

Työn tuloksena saatiin kehitettyä Yrityspelin ensimmäinen demoversio. Tämä muodostui toimivan arkkitehtuurin pohjalle rakennetusta pelin ytimestä eli pelimoottorista sekä pelin keskeisimmistä toiminnoista.

Asiasanat: peliohjelmointi, ASP.NET, C#, LINQ

## **Abstract**

Keijo Leskelä

Enterprise game, 40 Pages

Saimaa University of Applied Sciences

Technology Lappeenranta

Degree Programme in Information Technology

Software Engineering

Bachelor's Thesis 2011

Instructors: Mr. Martti Ylä-Jussila, Lecturer, Mr. Asko Kilpeläinen, Project manager

The purpose of this thesis was to design and develop an Enterprise game for students. The thesis is commissioned by the faculty of Health Care and Social Services of Saimaa University of Applied Sciences. The objective of the game was to encourage students to establish themselves a company and to teach them the basics of entrepreneurship.

The thesis was started with the study about theory of game design and available technologies. After having designed the preliminary architecture, ASP.NET was chosen as a main technology for the game. C# was used as the main programming language and MySQL database was queried with LINQ. Microsoft Visual Studio 2010 was the development environment.

As a result of this project, the architecture, the game engine and the most important functions of the game were designed and implemented. These form the first demo version of the Enterprise game.

Keywords: game programming, ASP.NET, C#, LINQ

## Termit

ADO.NET	ADO.NET on tietokantayhteyden ja tietokannan käsittelyn mahdollistava .NET-ympäristöön kuuluva komponenttikirjasto
Agentti	Agentit ovat peliin kuuluvia hahmoja
AJAX	Asynchronous JavaScript and XML eli AJAX on nimitys eri tekniikoiden yhdistelmälle, joka mahdollistaa käyttäjystävällisemmät Internet-sovellukset
ASP.NET	ASP.NET on Microsoftin kehittämä alusta Internet-sovellusten kehittämiseen
CIL	Common Intermediate Language on laitteistoriippumaton ns. välikieli
CLR	Common Language Runtime muodostaa CIL:sta laitteistokohtaisen koodin suoritettaessa sovellusta
CSS	Cascading Style Sheets on Internet-sivujen ulkoasun määrittämiseen käytettävä tyyliohje
DirectX	DirectX on ohjelmointirajapinta Windows ympäristöön
Entity Framework	Entity Framework on joukko ADO.NET:n sisältyviä tekniikoita, jotka helpottavat tietolähteen käsittelyä
Entity Data Model	Entity Data Model on tietolähteestä luotava malli
HTML	Hypertext Markup Language on kuvauskieli jota käytetään erityisesti Internet-sivujen kuvaamiseen
HTTP	Hypertext Transfer Protocol on Internetin tiedonsiirto-protokolla
IEnumerable<T>	IEnumerable<T> on rajapinta, joka mahdollistaa muutujakokoelman iteratiivisen käsittelyn

LINQ	Language Integrated Query on Microsoftin kehittämä menetelmä tietokanta- ym. kyselyjen tekemiseen .NET-ympäristössä
MVC	MVC on ohjelmistoarkkitehtuurityyli, jossa ohjelmiston osat erotetaan malliin (Model), näkymään (View) ja ohjaimen (Controller). Malli hoitaa tiedon käsittelyn, näkymä esittää tiedon käyttäjälle ja ohjain ottaa vastaan käyttäjän syötteet ja muokkaa näkymää ja mallia niiden mukaan
.NET	.NET on Microsoftin kehittämä laaja ohjelmistokehys
OpenGL	OpenGL eli Open Graphics Library on grafiikkarajapinta
Pelimaailma	Pelimaailma on pelin tapahtumaympäristö
SQL	Structured Query Language on standardoitu kieli tietokantakyselyjä varten
UML	Unified Modelling Language on graafinen ohjelmistojen mallinnuskieli
XML	Extensible Markup Language on merkintäkieli, jonka avulla tiedon merkitys saadaan kuvattua tiedon sekaan

## Sisältö

Termit.....	4
1 Johdanto .....	7
2 Asiakas .....	8
2.1 Saimaan ammattikorkeakoulu .....	8
2.2 Sosiaali- ja terveysalan yksikkö.....	9
2.3 Asiakkaan ongelma ja tavoite.....	9
3 Peliohjelmointi.....	10
3.1 Pelit .....	10
3.2 Pelien suunnittelu .....	13
3.3 Pelien arkkitehtuuri.....	14
3.4 Rajapinnat .....	15
3.5 Kehitysohjelmistot ja muut työkalut .....	16
3.6 Käyttöliittymäsuunnittelu.....	16
3.7 Verkkopelit.....	16
3.8 Testaus.....	17
4 Työssä käytettävät tekniikat ja menetelmät .....	18
4.1 Ohjelmiston kehittämistyön mallit .....	18
4.2 Toiminnallinen määrittely.....	19
4.3 Ohjelmiston suunnittelu .....	19
4.4 Kehitysympäristö .....	19
4.5 ASP.NET .....	20
4.6 ADO.NET .....	22
4.7 LINQ.....	23
4.8 MySQL .....	23
4.9 Uniform Server .....	23
4.10 Muita ohjelmia ja tekniikoita.....	24
5 Kehitysprojektin vaiheet.....	26
5.1 Projektin organisointi ja suunnittelu .....	26
5.2 Määrittely.....	27
5.3 Suunnittelu ja toteutus.....	27
5.4 Testaus.....	28
5.5 Ongelmia ja niiden ratkaisuja .....	28
6 Yrityspeli .....	30
6.1 Rekisteröityminen ja kirjautuminen.....	30
6.2 Päävalikko .....	31
6.3 Intro .....	32
6.4 Varsinainen peli.....	33
7 Yhteenveto ja pohdinta .....	35
Kuvat.....	36
Taulukot.....	36
Lähteet.....	37

## 1 Johdanto

Suomen väestö ikääntyy ja sosiaali- ja terveydenhuoltoalalle tarvitaan yhä enemmän osaajia. Julkinen terveydenhuolto ei pysty yksin tähän tarpeeseen vastaamaan, vaan alalle tarvitaan yksityisiä yrittäjiä. Suomalainen nuoriso ei kuitenkaan ole kovin innokasta ryhtymään yrittäjiksi. Harva terveydenhuoltoalan opiskelija edes harkitsee yrittäjäksi ryhtymistä, kun työpaikkoja on tarjolla jopa liikaa. Koska eläkeläisten määrä lisääntyy suhteessa työssäkävään väestöön, kuntien ja valtion mahdollisuudet ylläpitää riittävää terveydenhuoltojärjestelmää loppuvat eikä työpaikkoja ole tarjolla läheskään kaikille halukkaille. Kun hoitoa tarvitsevia ihmisiä kuitenkin riittää, tämä tarjoaa yrittäjäksi uskaltautuville hyvän mahdollisuuden työllistyä sekä ansaita keskimääräistä enemmän. Lisäksi saa vielä vapauden päättää omista työajoistaan. Perustamalla yrityksen nyt ennen suurten ikäluokkien eläköitymistä voi saavuttaa etulyöntiaseman kilpailussa asiakkaista.

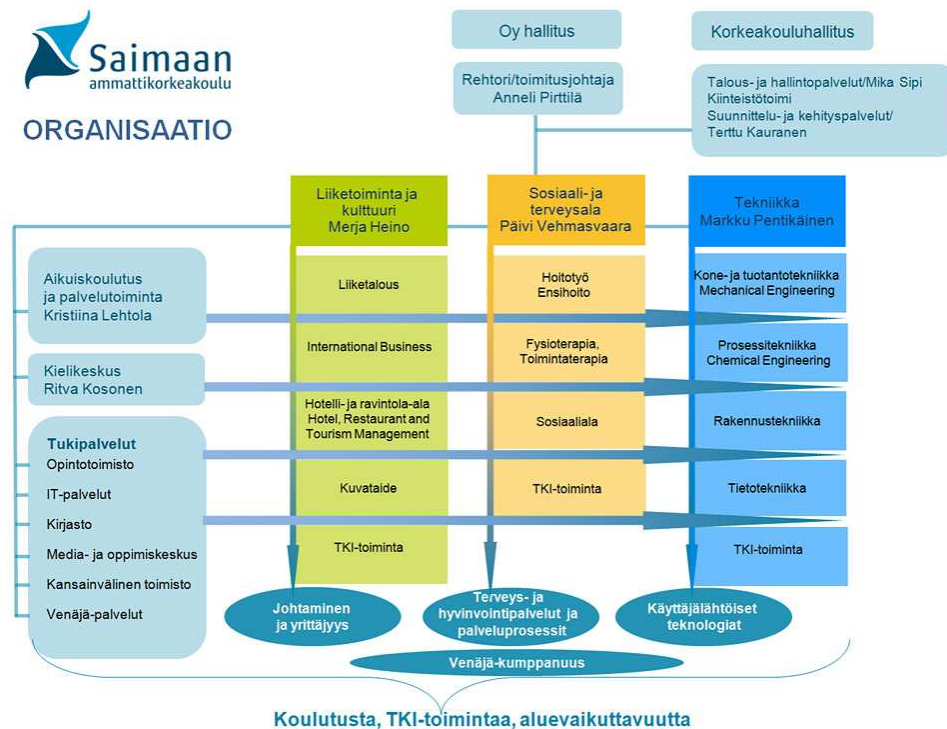
Nykyisin monet nuoret ovat kiinnostuneita tietokonepeleistä ja suurin osa käyttää Internetiä päivittäin. Mikäli Internet, pelit ja yrittäjyys saataisiin niputettua kiinnostavaksi kokonaisuudeksi, voitaisiin opiskelijoita saada kiinnostumaan, rohkaistumaan ja innostumaan yrittäjyydestä. Tämän lopputyön tavoitteena on suunnitella ja toteuttaa yrityspeli, joka kannustaa ja motivoi sosiaali- ja terveystien alan opiskelijoita yrittäjyyteen sekä opettaa yrityksen perustamisen ja pyörittämisen perusteita. Työssä käsitellään peliohjelmointia ja -suunnittelua yleisesti, yrityspelin toteuttamiseen tarvittavia menetelmiä ja tekniikoita. Lisäksi esitellään Yrityspeli-projektia ja sen tuloksia.

## 2 Asiakas

Yrityspelin tilaajana on Saimaan ammattikorkeakoulun sosiaali- ja terveysalan yksikkö. Seuraavassa on lyhyt esittely sekä Saimaan ammattikorkeakoulusta että sosiaali- ja terveysalan yksiköstä. Lisäksi kuvataan asiakkaan tarpeet yrityspelin käyttämisessä.

### 2.1 Saimaan ammattikorkeakoulu

Saimaan ammattikorkeakoulu on noin 3000 opiskelijan ammattikorkeakoulu Lappeenrannassa ja Imatralla. Henkilökuntaa on noin 300. Ammattikorkeakoulun organisaatiokaavio on esitetty kuvassa 2.1. Opetusta järjestetään viidellä toimialalla. Toimialoja ovat tekniikka, sosiaali- ja terveysala, liiketalous, hotelli- ja ravintola-ala sekä kuvataide. Ammattikorkeakoulututkintoon valmistavia koulutusohjelmia on 18, ja ylempään ammattikorkeakoulututkintoon voi valmistua neljässä koulutusohjelmassa. Saimaan ammattikorkeakoulun uusi Lappeenrannan Skinnarilaan syksyllä 2011 valmistunut koulurakennus on esitetty kuvassa 2.2. (Saimaan ammattikorkeakoulu 2011b.)



Kuva 2.1 Saimaan ammattikorkeakoulun organisaatiokaavio (Saimaan ammattikorkeakoulu 2011a)





Kuva 2.2 Saimaan ammattikorkeakoulun uudisrakennus

Saimaan ammattikorkeakoulu tarjoaa hyvät mahdollisuudet kansainvälistymiseen. Opiskelijoilla on mahdollisuus lähteä opiskelijavaihtoon ulkomaille tai opiskella englanninkielisissä opetusohjelmissa. (Saimaan ammattikorkeakoulu 2011b.)

## **2.2 Sosiaali- ja terveystieteiden yksikkö**

Saimaan ammattikorkeakoulussa voi opiskella sosiaali- ja terveystieteitä viidessä koulutusohjelmassa: ensihoito, fysioterapia, hoitotyö, sosiaaliala ja toimintaterapia. Sosiaali- ja terveystieteiltä valmistuneet voivat toimia myös itsenäisinä yrittäjinä. (Saimaan ammattikorkeakoulu 2011c.)

## **2.3 Asiakkaan ongelma ja tavoite**

Sosiaali- ja terveystieteiden opiskelijoilla on yksi yrittäjyyteen liittyvä kurssi, jonka laajuus vaihtelee koulutusohjelman mukaan. Sen puitteissa opiskelijoille tulisi kuitenkin opettaa perustiedot yrittäjyydestä ja siihen liittyvistä asioista. Opiskelijoita olisi myös saatava rohkaistua ryhtymään yrittäjiksi, sillä alan yrittäjätarve kasvaa tulevaisuudessa väestön ikääntyessä. Asiakkaan tärkeimpiä tavoitteita yrityspelin käyttämisessä onkin juuri opiskelijoiden rohkaus ja motivoiminen yrittämiseen ja sen opiskeluun. Pelin tulee myös opettaa perusasiat yrittämisestä ja siihen liittyvistä asioista.

### **3 Peliohjelmointi**

Pelejä on monenlaisia ja monenlaisissa eri ympäristöissä pelattavia. Peleille on olemassa erilaisia määritelmiä, mutta yleisesti voidaan sanoa, että ne ovat toimintaa, jolla on jonkinlaiset säännöt ja rajoitteet. Puhuttaessa peliohjelmoinnista peleillä tarkoitetaan digitaalisia tietokone- ja pelikonsolipelejä. Peliohjelmointi on siis pelien ohjelmointia, mutta se sisältää myös paljon muuta pelkän ohjelmoinnin lisäksi. Pelituotantoon kuuluvia osa-alueita ovat mm. ideoiminen, suunnittelun eri vaiheet, 3d-mallinnus, grafiikan luominen ja musiikki. (Kellomäki 2011a.)

#### **3.1 Pelit**

Pelien keskeisiä osia ovat pelaajat, säännöt, pelimaailma ja agentit. Pelaajat voivat olla joko oikeita ihmisiä tai tietokoneen ohjaamia hahmoja, jotka osallistuvat peliin. Sääntöjen avulla rajataan se, mitä pelaaja voi tehdä pelissä. Pelissä on usein myös jokin tavoite, johon pyritään. Tällaisia voivat olla esimerkiksi pisteiden kerääminen, vastustajan tuhoaminen tai jonkin ongelman ratkaisu. Pelimaailma on pelin tapahtumaympäristö. Yleensä pelimaailma esitetään pelaajalle graafisena käyttöliittymänä. Agentit ovat pelin hahmoja. Myös pelaajalla on usein oma agenttinsa. Pelejä voidaan luokitella niiden ominaisuuksien perusteella useisiin eri ryhmiin. Aina peliä ei kuitenkaan voida luokitella selvästi pelkästään yhteen kategoriaan. (Kellomäki 2011a.)

Toimintapelit ovat nopeitempisiä pelejä, joissa pelaajalta vaaditaan hyvää käden ja silmän koordinaatiota. Lisäksi pelaajan on osattava taktikoida, oltava nopea ja tarkka. Toimintapelejä ovat mm. erilaiset taistelupelit, tasohyppelyt ja ammuskelupelit (esimerkki kuvassa 3.1). Seikkailupelit puolestaan muodostuvat jonkin tarinan ympärille. Niissä pelaaja saa tarinan edetessä ratkaistavakseen erilaisia ongelmia. Pelaajan on ratkaistava ongelma, ennen kuin hän pääsee etenemään tarinassa. Jotkut pelit ovat toimintaseikkailuja, jotka yhdistävät toiminta- ja seikkailupelien ominaisuuksia. (Manninen 2007, 19 - 20.)



Kuva 3.1 Esimerkki toimintapelistä, The Great Giana Sisters (C64s 2011)

Roolipeleissä ohjataan hahmoa tai usean hahmon ryhmää. Hahmoja ohjataan suorittamaan tehtäviä ja hahmojen ominaisuuksia pyritään kehittämään. Strategiapelit (kuva 3.2) ovat hieman samantapaisia, mutta tehtävien ratkomisen sijaan niissä pyritään päihittämään vastustaja. Strategiapelit sisältävät enemmän loogista ajattelua, päätösten tekoa, suunnittelua ja taktikointia kuin roolipelit. Ongelmapeleissä ratkotaan erityyppisiä ongelmia ja niiden pelaaminen vaatii analyyttistä ajattelua. (Manninen 2007, 20 - 21.)



Kuva 3.2 Esimerkki strategiapelistä, Civilization III (Civ3 2003)

Urheilu- ja kilpailupeleissä on otettu mallia olemassa olevista urheilu- ja kilpailulajeista. Niissä pyritään pelin aiheena olevan lajin mahdollisimman tarkkaan mallintamiseen ja sääntöjen noudattamiseen. Tällaisia pelejä ovat mm. jääkiekkopelit, autopelit ja esimerkiksi TV-ohjelmiin perustuvat tietovisailut. Esimerkki urheilupelistä on esitetty kuvassa 3.3. (Manninen 2007, 22.)



Kuva 3.3 Esimerkki urheilupelistä, NHL 12 (EA Sports 2011)

Opetuspelit (kuva 3.4) ovat tarkoitettu opettamaan pelaajalle asioita tekemisen kautta. Opetettava aihe on yleensä rajattu tietyn aihealueen ympärille. Ajatuksena on, että pelin kautta oppiminen on sekä mielenkiintoisempaa että tehokkaampaa. (Manninen 2007, 22.)



Kuva 3.4 Esimerkki opetuspelistä, Matikkavisa (CognAid 2011)

Opetuspelien suunnittelu eroaa selvästi muiden pelityyppien suunnittelusta. Tyypillisiä opetuspelejä ovat mm. erilaiset matematiikkapelit, sanapelit ja liikennesääntöjen harjoitteluun tarkoitettut pelit. Pelityyppejä on paljon muitakin, mutta edellä on esitetty niistä muutamia yleisimpiä. (Manninen 2007, 22.)

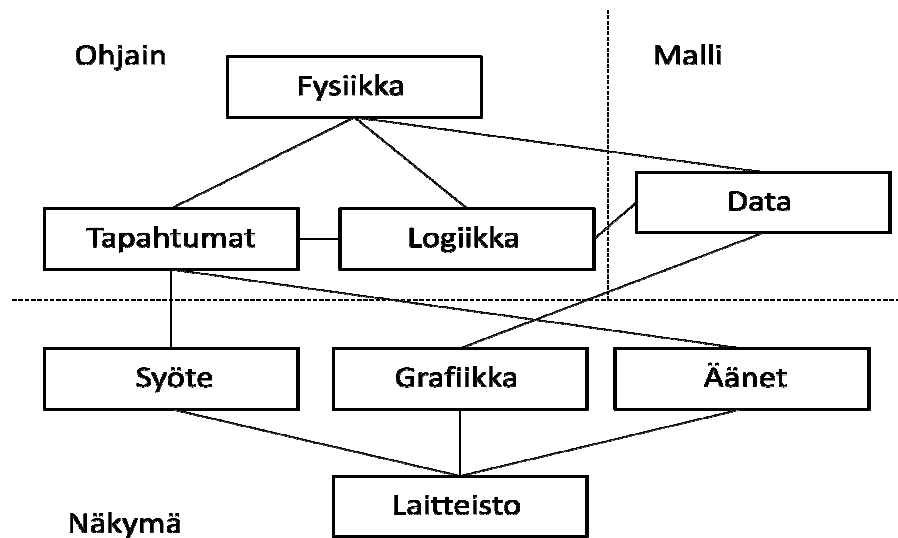
### 3.2 Pelien suunnittelu

Pelisuunnittelun tavoitteena on luoda runko pelin toteutukselle. Suunnittelun aloittamiseksi tarvitaan peli-idea, jonka pohjalle peli suunnitellaan. Pelisuunnittelussa määritellään pelin ominaisuudet ja säännöt, joiden mukaan peli etenee. Lisäksi suunnitteluun kuuluvat mm. pelin objektien luominen, pelimekaniikan suunnittelu ja pelin tasapainotus. Pelin objekteilla tarkoitetaan pelin hahmoja, pelimaailmaa, tavaroita sekä muita resursseja, joita peli sisältää. Pelimekaniikka

on pelin ja pelaajan välistä vuorovaikutusta, joka luodaan erilaisten sääntöjoukkojen perusteella. Pelisuunnittelun keskeisimpiä tavoitteita ovat toteutusvaiheen ongelmien ennaltaehkäisy ja hyvän pelikokemuksen muodostaminen. Käytännössä pelisuunnittelu on osa iteratiivista pelituotantoprosessia ja suunnitelma muuttaa muotoaan prosessin edetessä. (Manninen 2007, 28 - 32.)

### 3.3 Pelien arkkitehtuuri

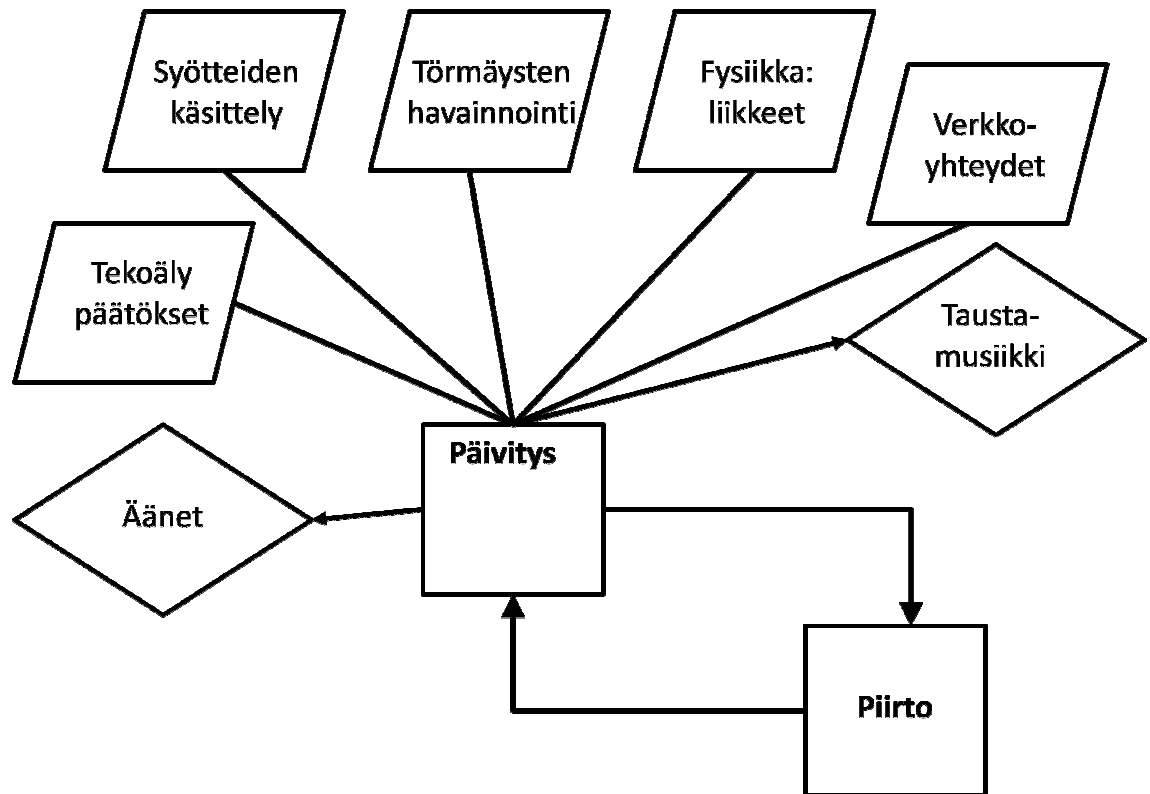
Tyypillinen peliarkkitehtuuri muodostuu kuvan 3.5 mukaisesta MVC-mallista (Model View Controller) tai sen muunnelmasta. Peliarkkitehtuurissa malli (Model) sisältää pelimaailman tilannetta kuvaavat tietorakenteet, funktiot tilanteen muuttamiseen ja siihen liittyviin kyselyihin vastaamiseen. Näkymä (View) esittää mallilta saamansa tiedot pelaajalle sopivassa muodossa, esimerkiksi grafiikan ja äänen avulla. Näkymä ottaa vastaan syötteitä ja muodostaa niistä tapahtumia ohjaimelle (Controller). Ohjain vastaanottaa syötteitä näkymiltä ja tekee malliin syötteiden mukaisia päivityksiä. (Kellomäki 2010a.)



Kuva 3.5 Peliarkkitehtuurin MVC-malli (Kellomäki 2010a)

Pelin rungoksi riittää alustus, pääsilmut ja lopetus -osiot. Alustuksessa voidaan esimerkiksi tarkastaa resurssien (muisti, levytila, prosessorin nopeus yms.) riittävyys, ladata pelissä tarvittavia elementtejä ja alustaa tiettyjä resursseja. Pääsilmutuksen peruserä on, että syötteen tullessa pelimaailma päivitetään. Usein pelimaailma päivitetään myös tietyn ajan kuluttua. Käytännössä pääsilmutus on paljon monimutkaisempi ja sen toteutuksessa tulee ottaa hu-

mioon erilaisia pelin toimivuuteen ja pelattavuuteen liittyviä asioita. Pääsilman rakenne on esitetty kuvassa 3.6. (Kellomäki 2010a.)



Kuva 3.6 Pääsilman rakenne (Florida Institute of Technology 2009)

Lopetuksessa vapautetaan resurssit ja suljetaan pelisovellus. Lopetus voi tapahtua joko tarkoituksellisesti tai esimerkiksi virheen seurauksena. (Kellomäki 2010a.)

### 3.4 Rajapinnat

Pelien tekemiseen liittyy erilaisia rajapintoja. Tällaisia ovat mm. grafiikka-, ääni- ja laitteistorajapinnat. Lisäksi useissa peleissä on tärkeässä osassa fysiikka ja fysiikkamoottori. Peliohjelmointiin on olemassa runsaasti valmiita rajapintoja, jotka ovat käytännössä valmiita funktioita sisältäviä kirjastoja. Niiden tarkoitus on helpottaa ja nopeuttaa pelin tekemistä. (Kellomäki 2010b.)

Grafiikkarajapintojen avulla toteutetaan laitteiston (esimerkiksi näytönohjain) kanssa tapahtuva graafisiin ominaisuuksiin liittyvä kommunikointi ja toimenpiteet. Tällä tavoin ohjelmoijan ei tarvitse tietää laiteläheisestä ohjelmoinnista vaan riittää, että hänellä on käytössään rajapinnan tarjoamat funktiot. Yleisim-

min käytetyt grafiikkarajapinnat ovat DirectX ja OpenGL eli Open Graphics Library. Äänirajapinnat toimivat samaan tapaan (näytönohjaimen sijaan äänikortti), ja äänen lisääminen peliin on ohjelmoijalle helppoa. Laitteistorajapinnat puolestaan on tarkoitettu oheislaitteiden, kuten hiiren, näppäimistön ja peliohjainten hallintaan. Fysiikkamoottoria hyödynnetään pelin objektien liikkeiden, törmäysten ja muiden vastaavien ominaisuuksien simuloinnissa. (Kellomäki 2010b.)

### **3.5 Kehitysohjelmistot ja muut työkalut**

Pelien kehitykseen on tarjolla runsaasti ohjelmistoja ja apuvälineitä. Yleensä kehitysohjelmisto sisältää normaalin ohjelmointiympäristön lisäksi jonkinlaisen pelimoottorin. Pelimoottori on kokoelma kirjastoja (rajapintoja), jotka on tarkoitettu erityisesti pelien tekoon. Pelimoottorit voivat olla vain yhtä peliä varten suunniteltuja tai yleiskäyttöisiä. Lisäksi on olemassa erityisesti tiettyyn pelikehityksen osa-alueeseen tarkoitettuja ohjelmia. Näitä ovat mm. grafiikan tekoon tai äänen käsittelyyn tarkoitettut ohjelmat. (Kellomäki 2011b.)

### **3.6 Käyttöliittymäsuunnittelu**

Käyttöliittymä on pelin kiinnostavuuden ja pelattavuuden kannalta yksi tärkeimpiä tekijöitä. Käyttöliittymäsuunnittelussa tulee ottaa huomioon kohdelaitteisto. Konsolipelit vaativat erilaisen käyttöliittymän kuin PC-pelit mm. erilaisten syöttölaitteistojen (ohjaimet vs. näppäimistö ja hiiri) vuoksi. Käyttöliittymä ei saisi olla liian hieno, ettei se vie huomiota itse pelaamiselta. Sen tulisi olla yksinkertainen, helppokäyttöinen ja yhtenäinen koko pelin ajan. Käyttöliittymän elementit tulisi sijoittaa siten, että ne muodostavat tasapainoisen kokonaisuuden. Parhaimmillaan pelaaja ei juuri huomaa käyttöliittymän olemassaoloa vaan pystyy eläytymään peliin täysin. (Ye 2000.)

### **3.7 Verkkopelit**

Verkkopelit ovat pelejä, joita pelataan Internetin tai muun tietokoneiden (tai muiden laitteiden) välisen verkon välityksellä. Ne voivat olla yksin- tai moninpelisiä sekä kuulua lähes mihin tahansa lajityyppiin. Osa verkkopeleistä vaatii pelin asentamista tietokoneelle ja osaa voi pelata suoraan Internetissä. Verkkopelien suunnittelussa on joitakin huomioonotettavia asioita. Yksi tärkeimmistä on tieto-



turva, sillä Internetissä on vaara joutua erilaisten hyökkäysten kohteeksi. Heikko tietoturva voi mahdollistaa huijaamisen pelissä ja siten viedä peli-ilon muilta pelaajilta. Muita tärkeitä asioita verkkopelien suunnittelussa ovat mm. fyysisten resurssien asettamien rajojen huomioiminen, reaaliaikaisuus ja loogisen rakenteen valitseminen. Loogisen rakenteen perusratkaisuvaihtoehdot ovat asiakaspalvelin- tai vertaisverkkorakenne. (Kellomäki, 2010d)

### **3.8 Testaus**

Normaalin ohjelmistotestauksen menetelmät toimivat myös peliohjelmoinnissa. Peliohjelma ei ole yhtä kriittinen kuin jokin toinen ohjelmatyyppi voi olla, joten pelitestauksessa painotukset ovat hieman erilaisia. Yleisimpiä testityyppejä ovat toiminnallisuus-, rasitus- ja suorituskykytestaus sekä pelattavuus-, käytettävyys- ja konfiguraatiotestaus. Toiminnallisuustestauksessa testataan, että määrittelyistä ja muusta dokumentaatiosta saatavat tavoitteet täyttyvät. Koska pelit vaativat usein paljon tehoa käytettävältä laitteistolta, kannattaa peleille suorittaa rasitus- ja suorituskykytestausta. Näissä peliä testataan olosuhteissa, joissa resurssit (esimerkiksi keskusmuisti) ovat vähissä. (Kellomäki 2010c.)

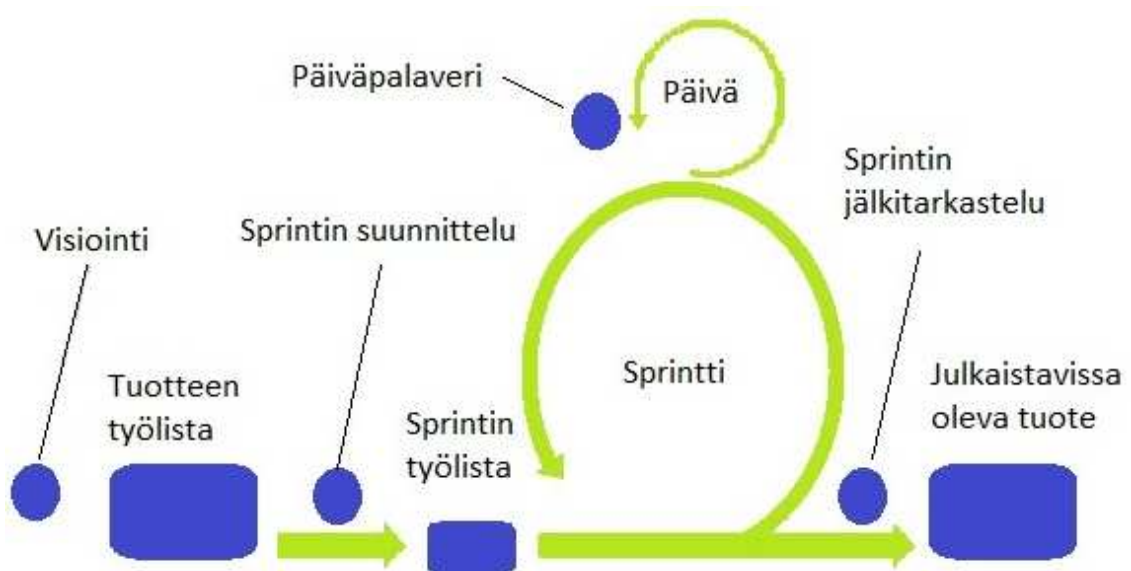
Pelattavuustestauksessa varmistetaan, että pelin tulevat käyttäjät pitävät pelistä. Testauksen tekevät yleensä ulkopuoliset testaajat eli tässä tapauksessa pelaajat. Pelaajat saavat pelin pelattavakseen ja lopuksi heitä haastatellaan pelikokemusten selvittämiseksi. Käytettävyystestaus on hyvin samantapaista kuin pelattavuustestaus, mutta siinä pelaajille ei anneta minkäänlaisia ohjeita. Tarkoituksena on selvittää, kuinka pelaaja pystyy toimimaan pelissä. Konfiguraatiotestauksella tutkitaan pelin toimivuutta erilaisilla laitteistoyhdistelmillä. (Kellomäki 2010c.)

## 4 Työssä käytettävät tekniikat ja menetelmät

Tässä luvussa esitellään työssä käytetyt tekniikat ja menetelmät. Lisäksi kerrotaan työssä käytetyistä työkaluista sekä ohjelmistotuotannon teoriasta yleisesti. Kaikkein yleisimpiä tietotekniikan menetelmiä ja käytäntöjä ei käsitellä.

### 4.1 Ohjelmiston kehittämistyön mallit

Ohjelmiston kehittämistyö voidaan jakaa eri tavoin vaiheisiin, jolloin muodostuu vaihejakomalli. Vaiheita voivat olla esimerkiksi esitutkimus, määrittely, suunnittelu, toteutus, testaus, käyttöönotto ja ylläpito. Tyypillisimmin käytetty vaihejakomalli on vesiputousmalli, jossa edellä mainitut vaiheet suoritetaan vaihe kerrallaan esitutkimuksesta alkaen. Edelliseen vaiheeseen palaaminen on hankalaa. Pahimmassa tapauksessa myöhäisessä vaiheessa havaittu ongelma pakottaa aloittamaan koko projektin alusta. Tämän vuoksi on kehitetty muita vaihejakomalleja, joista monet tosin pohjautuvat vesiputousmalliin. Niillä on pyritty kuvaamaan ohjelmistokehitysprosessin iteratiivisuutta vesiputousmallia paremmin. Muita vaihejakomalleja ovat mm. protoilu, RUP (Rational Unified Process), spiraalimalli sekä erilaiset ketterät menetelmät (kuva 4.1). (Haikala & Märijärvi 2004, 42 - 46; Hiltunen 2005.)



Kuva 4.1 Esimerkki ketterän menetelmän periaatteesta (Ketterät käytännöt 2011)

Ketterissä menetelmissä ei panosteta dokumentointiin yhtä paljoa kuin muissa menetelmissä. Kehitys jaetaan lyhyisiin iteraatioihin, joiden aikana käydään periaatteessa koko ohjelmiston kehitysprosessi läpi. Ratkaisut ovat joustavia, virheet havaitaan aikaisessa vaiheessa ja muutosten ja korjausten tekeminen sujuu nopeasti. (Haikala & Märijärvi 2004, 47 - 48.)

#### **4.2 Toiminnallinen määrittely**

Toiminnallisessa määrittelyssä kuvataan järjestelmälle (ohjelmistolle) asetetut vaatimukset sekä vaatimukset toteuttava järjestelmä. Toiminnallisen määrittelyn tuloksena syntyvä dokumentti sisältää yleiskuvauksen järjestelmästä; mitä ollaan tekemässä, kenelle ja miksi. Järjestelmästä kuvataan toiminnot ja ominaisuudet, jotka sen tulee toteuttaa, tietokanta ja sen sisältö, sekä erilaiset toteutukseen liittyvät rajoitukset. Lisäksi kuvataan mahdolliset ulkoiset liittymät. Usein dokumentointiin käytetään apuna UML-mallinnuskieltä (Unified Modelling Language) ja sen eri kaavioita. (Haikala & Märijärvi 2004, 78 - 81.)

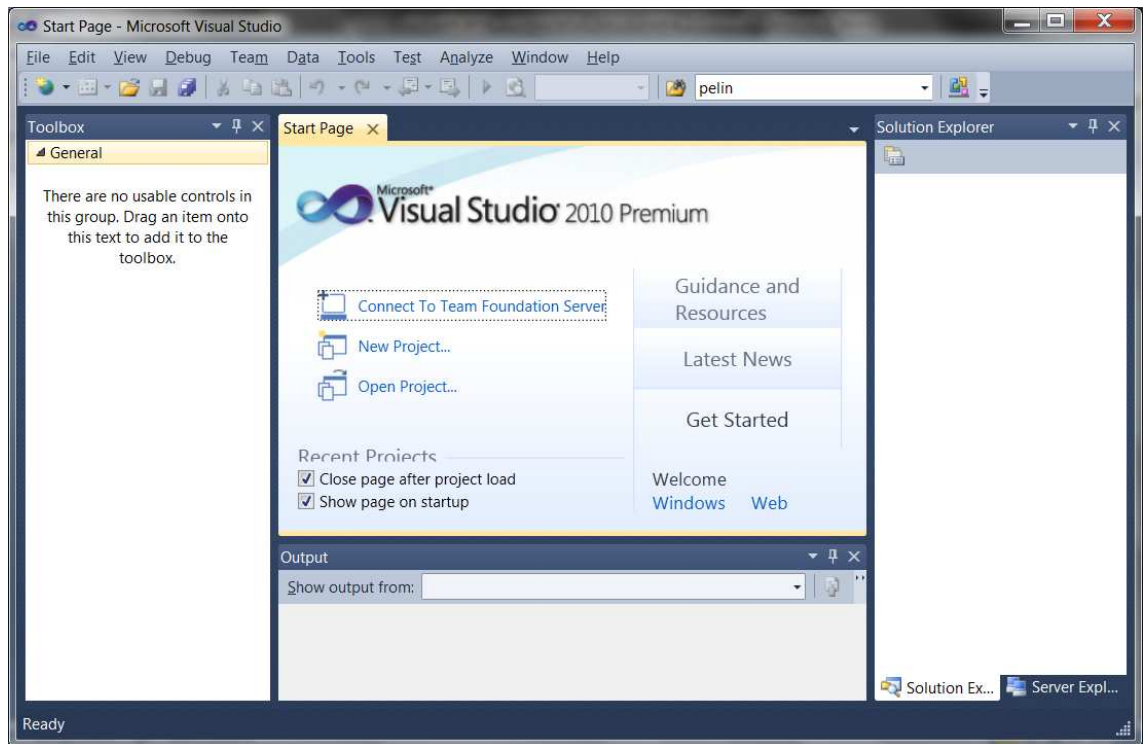
#### **4.3 Ohjelmiston suunnittelu**

Ohjelmiston suunnittelun tavoitteena on tuottaa määrittelyyn perustuva tekninen kuvaus järjestelmästä. Suunnittelu voidaan jakaa arkkitehtuurisuunnitteluun ja moduulisuunnitteluun. Arkkitehtuurisuunnittelussa järjestelmää kuvataan korkealla abstraktiotasolla. Tarkoitus on jakaa järjestelmä osiin ja määritellä rajapinnat. Moduulisuunnittelussa määritellään näiden osien (moduulien) tarkempi sisältö ja toiminta. Ohjelmiston suunnitteluvaiheen aikana syntyviä dokumentteja voivat olla esimerkiksi arkkitehtuurisuunnitelma tai tekninen määrittely. (Haikala & Märijärvi 2004, 81 - 83.)

#### **4.4 Kehitysympäristö**

Ohjelmiston kehitysympäristö on tärkeä tekijä ohjelmistoprojektin onnistuneessa toteutuksessa. Kehitysympäristö voi koostua yhdestä tai useammasta ohjelmasta. Mikäli kehitysympäristön ohjelmat on yhdistetty yhdeksi ohjelmistoksi, voidaan puhua integroidusta ohjelmointiympäristöstä (IDE, Integrated Development Environment). IDE:n vähimmäisvaatimukset ovat mahdollisuus kirjoittaa koodia sekä kääntää se kohdealustalle sopivaan muotoon. Usein IDE sisältää

kuitenkin runsaasti muita työkaluja ja välineitä ohjelmointia helpottamaan. Eri-  
laisia kehitysympäristöjä on tarjolla runsaasti. Yksi tällainen on Microsoftin Vi-  
sual Studio 2010 (kuva 4.2).



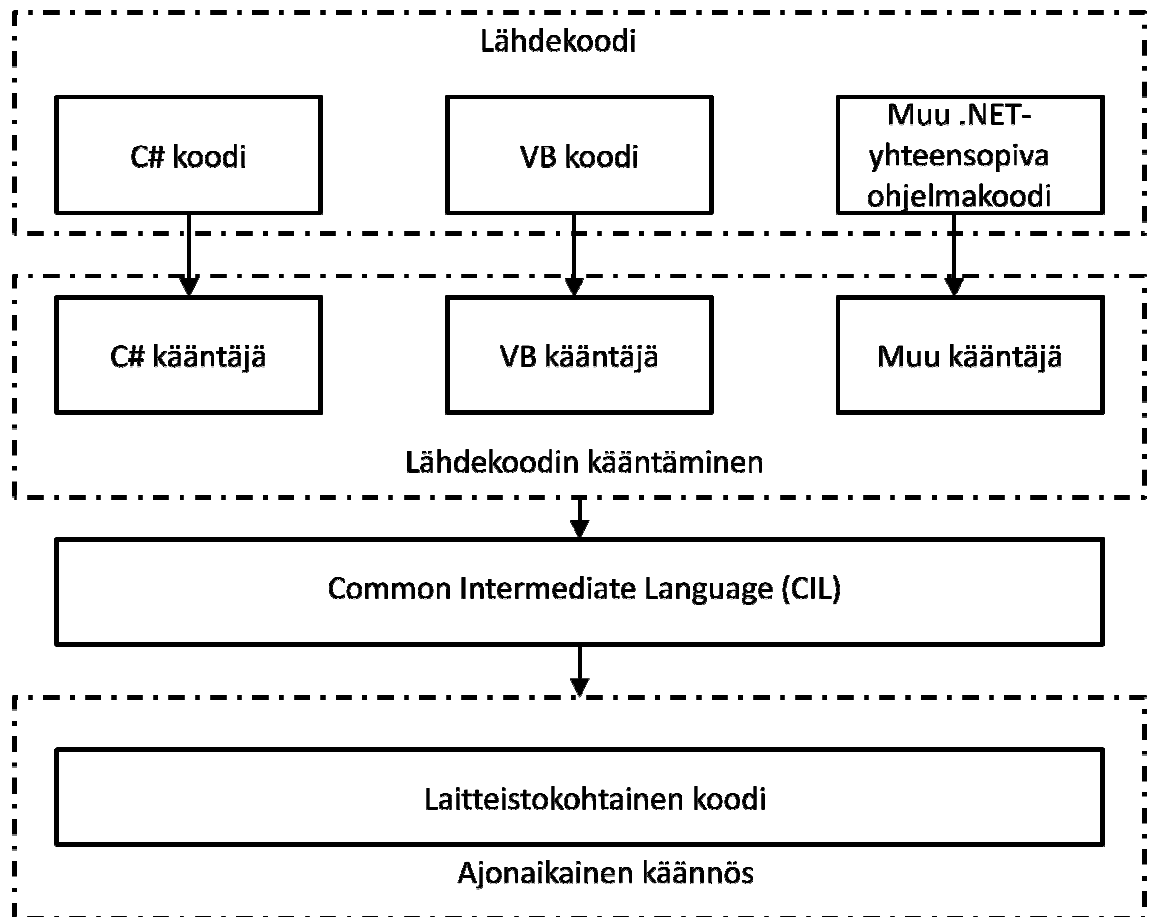
Kuva 4.2 Microsoft Visual Studio 2010 Premium

Visual Studio 2010 (VS2010) ohjelmistosta on useita eri versioita. Tässä käsitel-  
lään pääasiassa Premium-versiota, mutta perusominaisuudet ovat kaikilla ver-  
sioilla samat. VS2010 sisältää kehitysympäristön sekä runsaasti erilaisia työka-  
luja. Se on tarkoitettu älypuhelin-, Internet- ja PC-ohjelmistokehitykseen. Tietty-  
jen lisäosien avulla sitä voidaan käyttää myös konsolipelien kehityksessä. Pre-  
mium-versiolla on mahdollista ohjelmoinnin lisäksi mm. suorittaa automatisoitua  
testausta ja erilaisia analyysejä sekä luoda tietokantoja ja realistista testausda-  
taa. Se sisältää myös tuen ketterien menetelmien käyttöön. (Microsoft 2010a.)

#### 4.5 ASP.NET

ASP.NET on Microsoftin kehittämä alusta Internet-sovellusten kehittämiseen.  
Se hyödyntää .NET-kehystä ja mahdollistaa siten kaikkien .NET-  
ominaisuuksien hyödyntämisen Web-sovelluksissa. Koska .NET-teknologia  
hyödyntää CLR:a (Common Language Runtime), voidaan sen ohjelmointiin  
käyttää mitä tahansa CLR-yhteensopivaa ohjelmointikieltä (esim. C#, Visual

Basic). Ohjelmakoodi käännetään ensin CIL-kielelle (Common Intermediate Language). CIL on laitteistoriippumaton kieli, josta sovellusta suoritettaessa muodostetaan CLR:n avulla laitteistokohtainen ohjelmakoodi. CLR:n perusperiaate on esitetty kuvassa 4.3. (Microsoft 2010b; DotNetSpark 2011a, 2011b.)



Kuva 4.3 CLR:n perusperiaate (Source ASP.Net 2010)

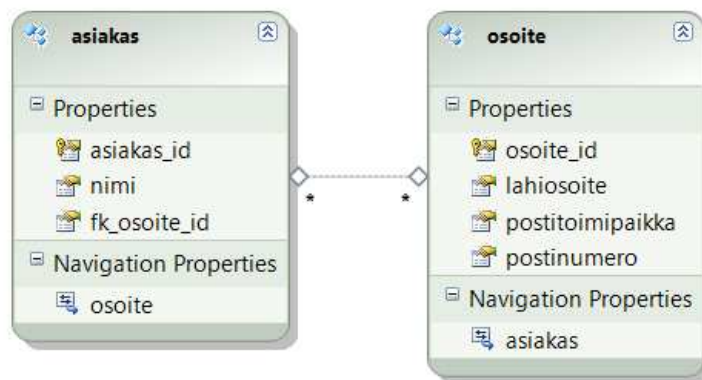
Kuten yleensä Internetissä, myös ASP.NET käyttää tiedonsiirrossa HTTP-protokollaa (Hypertext Transfer Protocol). HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), ja Javascript sekä muu staattinen teksti voidaan siirtää palvelimelta asiakkaalle sellaisenaan. Sen sijaan ASP.NET:n dynaamiset kontrollit on muunnettava sivua luotaessa HTML:ksi, jotta ne voidaan esittää käyttäjälle. Tästä huolehtii ASP.NET-ajoympäristö (runtime), joka suorittaa myös sivulle lisätyt ohjelmakoodit. Koodit voivat olla tehtynä millä tahansa .NET yhteensopivalla ohjelmointikielellä. Ne voivat olla joko HTML:n seassa tai erillisessä tiedostossa ja ne suoritetaan automaattisesti tai käyttäjän toimenpiteen seurauksena. (Spaanjars 2010.)

Webforms on yksi tapa toteuttaa ASP.NET-sivuja. Sivujen toteutus webforms:n avulla on hyvin lähellä tavallisten Windows-sovellusten tekemistä. Webforms sisältää runsaasti valmiita komponentteja, joita on helppo lisätä sivulle. Komponentteja voi myös tehdä itse tai hankkia ns. kolmannen osapuolen komponentteja. Webforms esittää tiedot käyttäjälle, mutta sovelluksen logiikka toteutuu palvelinpuolen koodissa. (StartVBDotNet 2011.)

#### 4.6 ADO.NET

ADO.NET tarjoaa pääsyn tietokantaan tai muuhun tietovarastoon. Sen avulla voidaan muodostaa yhteys tietokantaan sekä hakea ja muokata siellä sijaitsevia tietoja. ADO.NET jakaa tietokantayhteyden muodostamisen ja tiedonmuokkauksen erillisiin komponentteihin, joita on mahdollista käyttää sekä yhdessä että erikseen. Tämä on toteutettu ADO.NET:n sisältämällä .NET-kehysten kirjastoilla (data provider), joiden avulla muodostetaan tietokantayhteys, suoritetaan kysyksiä ja vastaanotetaan tietoja. (Microsoft 2010c.)

Entity Framework (EF) on joukko ADO.NET:n sisältyviä teknologioita. EF:n avulla tietokannasta voidaan luoda eräänlainen oliomalli, joka helpottaa tiedon käsittelyä. Tätä mallia kutsutaan entiteettimalliksi (Entity Data Model). Entiteettimalli on esitetty kuvassa 4.4.



Kuva 4.4 Entiteettimalli

Toinen vaihtoehto on tehdä ensin malli ja luoda tästä mallista tietokanta. EF:n sisältämät menetelmät tiedon käsittelyyn ovat LINQ (ks. kappale 4.7), Entity

SQL sekä edellisten ominaisuuksia yhdistävä Query builder methods. (Microsoft 2010d.)

#### **4.7 LINQ**

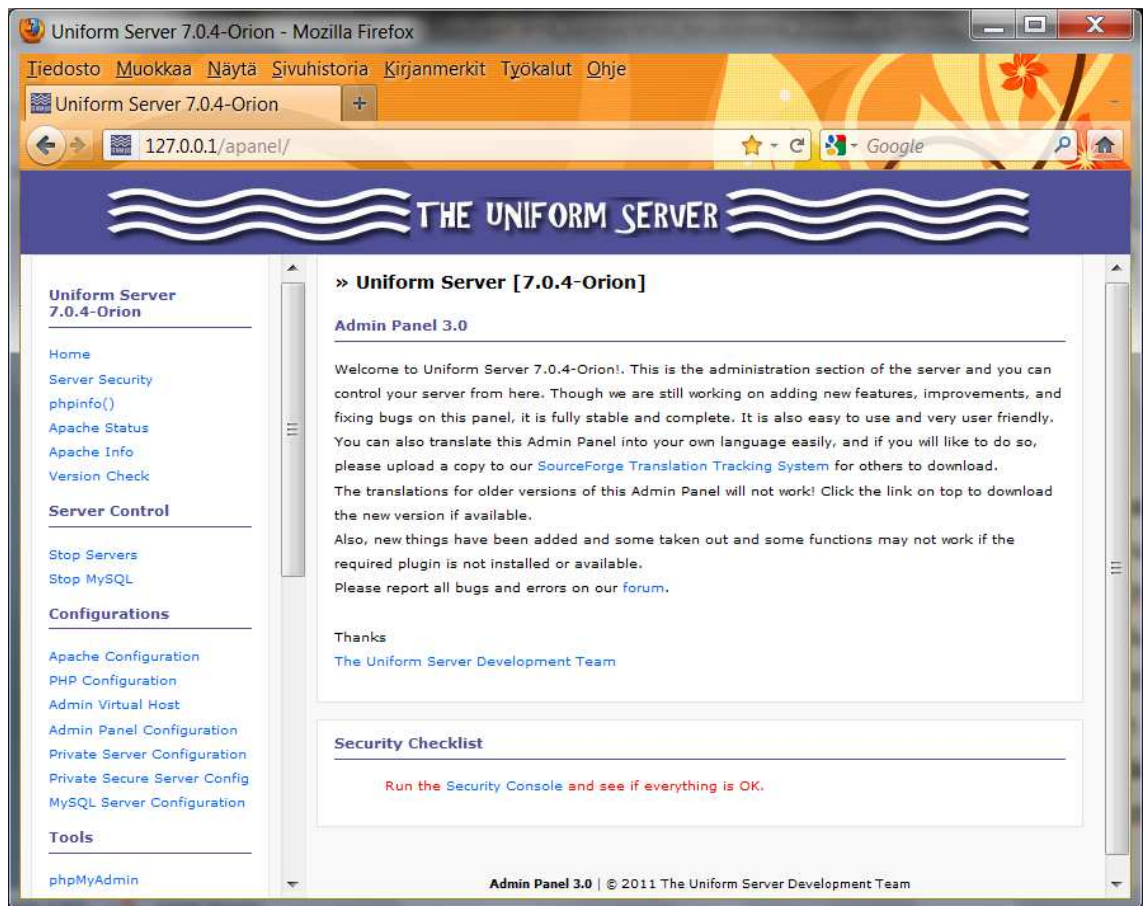
LINQ eli Language Integrated Query on Microsoftin kehittämä kyselykieli .NET ympäristöön. Se mahdollistaa tietokanta- ym. kyselyiden tekemisen ohjelmoijan käyttämällä ohjelmointikielellä (esim. C# tai Visual Basic). Se sisältää valmiit metodit mm. hakujen suodatukseen. Kyselyjä voidaan tehdä mihin tahansa IEnumerable<T>-rajapinnan toteuttavaan tietolähteeseen. LINQ myös mahdollistaa kolmansien osapuolien tekemät laajennukset ja muutokset sen perustoimintoihin. (Box & Hejlsberg 2007.)

#### **4.8 MySQL**

MySQL on nykyisin Oraclen omistama avoimen lähdekoodin SQL-tietokanta ja sen hallintajärjestelmä. MySQL noudattaa pääosin SQL-standardia (Structured Query Language), mutta joidenkin laajennusten osalta standardista on poikettu. Se toimii mm. Windows ja Linux -ympäristöissä ja sisältää ohjelmointirajapinnat useille ohjelmointikielille. .NET-ohjelmointia varten MySQL:n on saatavilla Connector/NET-ajuri. Se mahdollistaa .NET-ympäristön tietokannan käsittelyominaisuuksien (EF, LINQ jne.) hyödyntämisen MySQL-tietokannoissa ADO.NET:n avulla. Työkaluja MySQL:n käyttämiseen ovat mm. PhpMyAdmin ja MySQL Workbench, joiden avulla voidaan esimerkiksi luoda tietokantoja ja hallita käyttöoikeuksia. (MySQL 2011.)

#### **4.9 Uniform Server**

Uniform Server on kevyt ja pienikokoinen WAMP eli Windows, Apache, MySQL ja PHP -palvelin. Se on siirrettävä eli sitä voidaan käyttää esimerkiksi USB-tikulta. Uniform Server sisältää palvelimen lisäksi mm. PhpMyAdmin-sovelluksen sekä muita palvelimen ja tietokannan hallintaan tarvittavia sovelluksia. Parhaiten Uniform Server soveltuu testauskäyttöön, mutta sitä voidaan käyttää myös varsinaisena palvelimena. Uniform Serverin käyttöliittymä on esitetty kuvassa 4.5. (Uniform Server 2011.)



Kuva 4.5 Uniform Server -palvelimen käyttöliittymä

#### 4.10 Muita ohjelmia ja tekniikoita

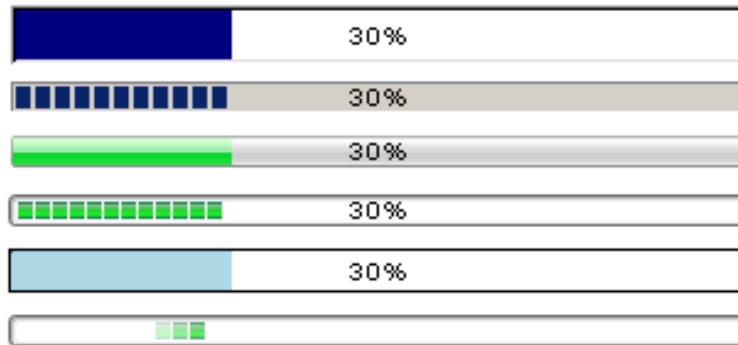
DayPilot Lite on avoimen lähdekoodin versio Annpoint-yhtiön kehittämästä DayPilot-kontrollikokoelmasta ASP.NET-ympäristöön. Se sisältää kalenteri (Calendar) ja aikataulutus (Scheduler) ASP.NET-kontrollit. DayPilot Calendar Lite on esitetty kuvassa 4.6. (DayPilot 2011a.)

	11/13/2011	11/14/2011	11/15/2011	11/16/2011	11/17/2011	11/18/2011	11/19/2011
9 AM							
10 AM							
11 AM						Event 6 (11:00 AM - 1:00 PM)	
12 PM							
1 PM							
2 PM							
3 PM						Event 1 (3:50 PM - 3:55 PM)	
4 PM						Event 2 (4:00 PM - 4:15 PM)	
5 PM						Event 3 (4:15 PM - 6:45 PM) Sales Dept. Meeting	

Kuva 4.6 DayPilot Calendar Lite (DayPilot 2011b)



EO Progress Bar on ilmainen ASP.NET-komponentti, jolla voidaan seurata jonkin parametrin tai muun vastaavan tilaa tai esimerkiksi latausprosessin edistymistä. Sen on kehittänyt Essential Objects Inc. EO Progress Bar eri versioineen on esitetty kuvassa 4.7. (Essential Objects 2011.)



Kuva 4.7 EO Progress Bar:n eri esitystapoja (Essential Objects 2011)

AJAX (Asynchronous JavaScript and XML) on tapa hyödyntää jo olemassa olevia tekniikoita. Sitä käytetään silloin, kun ei haluta päivittää koko sivua, vaan ainoastaan osa siitä. (W3Schools 2011a.) CSS eli Cascading Style Sheets on tyyliohje, jonka avulla voidaan muotoilla Internet-sivujen ulkoasua halutun näköiseksi. Se määrittelee, miten HTML-elementit esitetään sivulla. (W3Schools 2011b.)

## 5 Kehitysprojektin vaiheet

Tässä luvussa kerrotaan Yrityspelin kehityksen vaiheista. Aluksi käydään läpi projektin organisointi ja siihen liittyvät asiat. Seuraavana kerrotaan määrittelystä, suunnittelusta ja toteutuksesta sekä testauksesta. Lopuksi esitellään työn edetessä esille tulleita ongelmia ja niiden ratkaisuja.

### 5.1 Projektin organisointi ja suunnittelu

Yrityspeli- ja projektin organisointi alkoi 2010 syksyllä Janne Ariluodon aloittaessa toiminnallisen määrittelyn Asko Kilpeläisen ja Martti Ylä-Jussilan ohjauksessa. Keväällä 2011 liityin projektiin suunnittelemaan ja toteuttamaan Yrityspeliä. Myös Ville Lehto liittyi projektiin asiantuntijaksi. Yrityspeli- ja projektin tehtävät ja vastuut jakautuivat taulukon 5.1 mukaisesti.

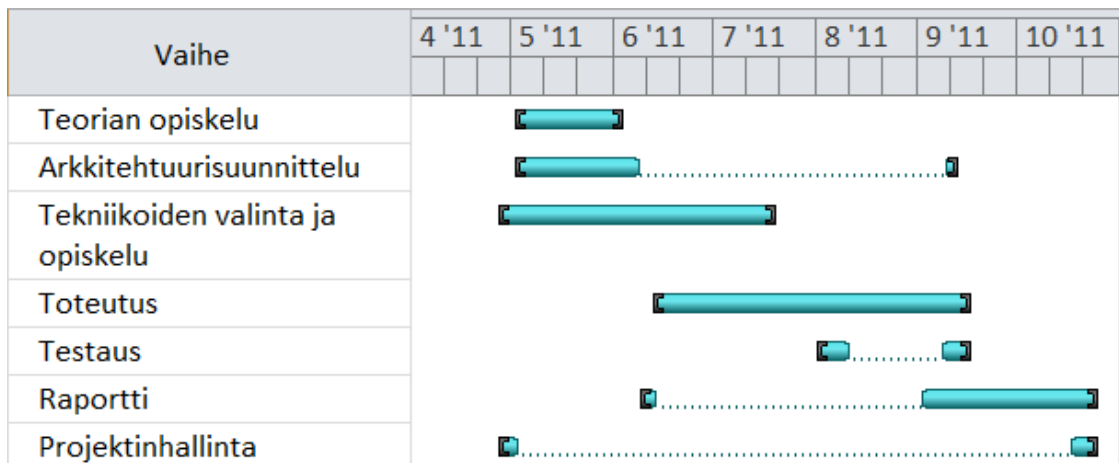
Taulukko 5.1 Yrityspelin organisaatio

Yrityspeli - Organisaatio		
Nimi	Tehtävä	Vastuut
Martti Ylä-Jussila	Opinnäytetyön ohjaaja, konsultti	Projektin ohjaus ja konsultointi
Asko Kilpeläinen	Opinnäytetyön ohjaaja, asiakkaan edustaja	Asiakkaan tavoitteiden esittäminen
Ville Lehto	Asiantuntija, asiakkaan edustaja	Yrittämisen asiantuntijan näkökulman esille tuominen
Janne Ariluoto	Projektipäällikkö, määrittelijä, graafinen suunnittelija	Kokonaisuus, toiminnallinen määrittely, pelin ulkoasu
Keijo Leskelä	Ohjelmistosuunnittelija	Suunnittelu, arkkitehtuuri, tekniikat, toteutus

Omalta osaltani projekti käynnistyi projektisuunnitelman laatimisella. Tähän sisältyi aikataulun laatiminen sekä työmääräarvion tekeminen. Alustava työmääräarvio työn eri vaiheista ja aikataulusta on esitetty taulukossa 5.2. Kuvassa 5.1 on esitetty Ganttin kaavio Yrityspelin toteutuneista vaiheista ja niiden ajoittumisesta.

Taulukko 5.2 Alustava työmääräarvio ja aikataulu

Vaihe	Arvioitu kesto [h]	Valmis
Tekniikoiden opiskelu	100	15.6.2011
Määrittelyn viimeistely	30	31.5.2011
Arkkitehtuurin suunnittelu	50	30.6.2011
Toteutus	100	15.10.2011
Testaus	50	25.10.2011
Koulutus ja käyttöönotto	40	1.11.2011
Raportti	100	2.11.2011



Kuva 5.1 Ganttin kaavio yrityspelin työvaiheista

Projektin etenemistä seurattiin viikkopalavereissa, joihin asiakkaan edustaja osallistui tarvittaessa. Lisäksi pidettiin suunnittelupalavereja. Pääosin työskentely oli itsenäistä. Projektin edetessä huomasimme, että toteutus vaatii huomattavasti arvioitua enemmän työtä. Tähän vaikuttivat mm. määrittelyn keskeneräisyys, tekijöiden vähäinen kokemus vastaavanlaisista projekteista sekä pelissä vaadittu reaali maailman mallintamisen laajuus.

## 5.2 Määrittely

Toiminnallisesta määrittelystä vastasi pääosin Janne Ariluoto. Määrittely eteni pikkuhiljaa suunnittelun ja toteutuksen rinnalla. Vuoden 2011 syksyyn mennessä se ehti muuttua varsin radikaalisti kevään versiosta.

## 5.3 Suunnittelu ja toteutus

Aloitin suunnittelun tutkimalla mahdollisia toteutustekniikoita ja opiskelemalla peliohjelmoinnin teoriaa. Tutkittuani asiaa päätin valita toteutuksen pohjaksi ASP.NET-alustan. Vaihtoehtoja olisi ollut runsaasti, mutta halusin opiskella uuden, laajasti käytetyn tekniikan, josta voisi olla hyötyä myös tulevaisuudessa. Lisäksi ASP.NET:n ominaisuudet ja sen kehitysyökalut vaikuttivat varsin monipuolisilta.

Seuraavaksi suunnittelin pelille alustavan arkkitehtuurin, jota tarkensin myöhemmin soveltumaan paremmin valittuihin tekniikoihin. Alustavan arkkitehtuurisuunnitelman jälkeen aloin testata sen toimivuutta toteuttamalla pelin ytimen ja

alun toimintoja. Samalla suunnittelin ja toteutin tietokantaa pieninä, tarpeen mukaisina kokonaisuuksina. Ydin osoittautui toimivaksi lähes sellaisenaan, vaikka joitain pieniä muutoksia jouduinkin tekemään. Muutoinkin pelin ytimestä muodostui joustava, sillä muutosten ja lisäysten tekeminen siihen onnistuu suhteellisen helposti. Tapahtumien ja sivujen suunnittelussa pyrin yleiskäyttöisyyteen, jotta samoja asioita ei tarvitsisi tehdä moneen kertaan.

Toteutus siis eteni siten, että peliä suunniteltiin pieni osa ja sitten toteutettiin se. Toteutuksen pääpaino oli toiminnoissa ja toiminnallisuudessa, ulkoasun suunnittelisi ja viimeistelisi myöhemmin Janne Ariluoto. Vaikka pelistä oli tarkoitus tulla yksinkertainen, oli mallinnettavia osia niin paljon, ettei kaikkea ollut mahdollista saada valmiiksi tämän opinnäytetyön puitteissa. Loppuvaiheessa projektiin alettiin hakea toista toteuttajaa jatkamaan se valmiiksi.

#### **5.4 Testaus**

Koska peli ei ehtinyt valmistua projektin aikana, ei testaustakaan päästy tekemään kuin lähinnä moduulitestauksen tasolla. Moduulitestauksen ohella tein joitakin testejä selvittääkseni järjestelmän käyttäytymistä usealla yhtäaikaishalla käyttäjällä.

#### **5.5 Ongelmia ja niiden ratkaisuja**

Projektin edetessä vastaan tuli monenlaisia ongelmia, jotka täytyi ratkaista. Koska aiempaa kokemusta käytetyistä tekniikoista ja menetelmistä ei ollut, aiheutui pelkästään tästä syystä lukuisia virhe- ja ongelmatilanteita. Onneksi myös muilla oli ollut samankaltaisia ongelmia ja Internetissä oli runsaasti tietoa niiden ratkaisemiseen. Internetin oppaat ja ym. ohjeistukset olivat kuitenkin pääasiassa pelkkiä Microsoftin tuotteita käyttäville, joten esimerkiksi tietokantayhteyksiin liittyvissä ongelmissa vaadittiin hieman enemmän soveltamista.

Pelin kannalta merkittävimpiä ongelmia oli se, miten yksittäiset pelit ja niihin liittyvä tieto saadaan pelaaja- ja pelikohtaisiksi. Ratkaisuna päädyttiin käyttämään tietokantataulua, johon tuli pelikohtaiset perustiedot. Tämän taulun pääavainta käytettiin sitten muiden pelikohtaisten taulujen viiteavaimena. Lisäksi se sisälsi viiteavaimen pelaajaan, jotta voidaan tunnistaa pelaajan pelit.

Yksi ongelma oli pelaajan tekemä toimenpide, joka aiheutti tiedon tallentamisen tietokantaan. Jos pelaaja tämän jälkeen ei tallenna peliä ja jatkaa aiemmin tallennetusta kohdasta, säilyy kantaan tallennettu tieto. Tällöin pelaajan on mahdollista "huijata" eli esimerkiksi pelata sama asiakaskäynti useaan kertaan ja saada joka kerta maksu tilille. Ongelma ratkaistiin lisäämällä esimerkiksi tilipahtumien yhteyteen ns. pakotettu tallennus. Eli tallennettaessa tilille tallennetaan myös pelitilanne automaattisesti. Pelitilanne tallennetaan myös siirryttäessä seuraavaan päivään, jotta päivän lopuksi generoitavia varauksia ei generoida uudelleen.

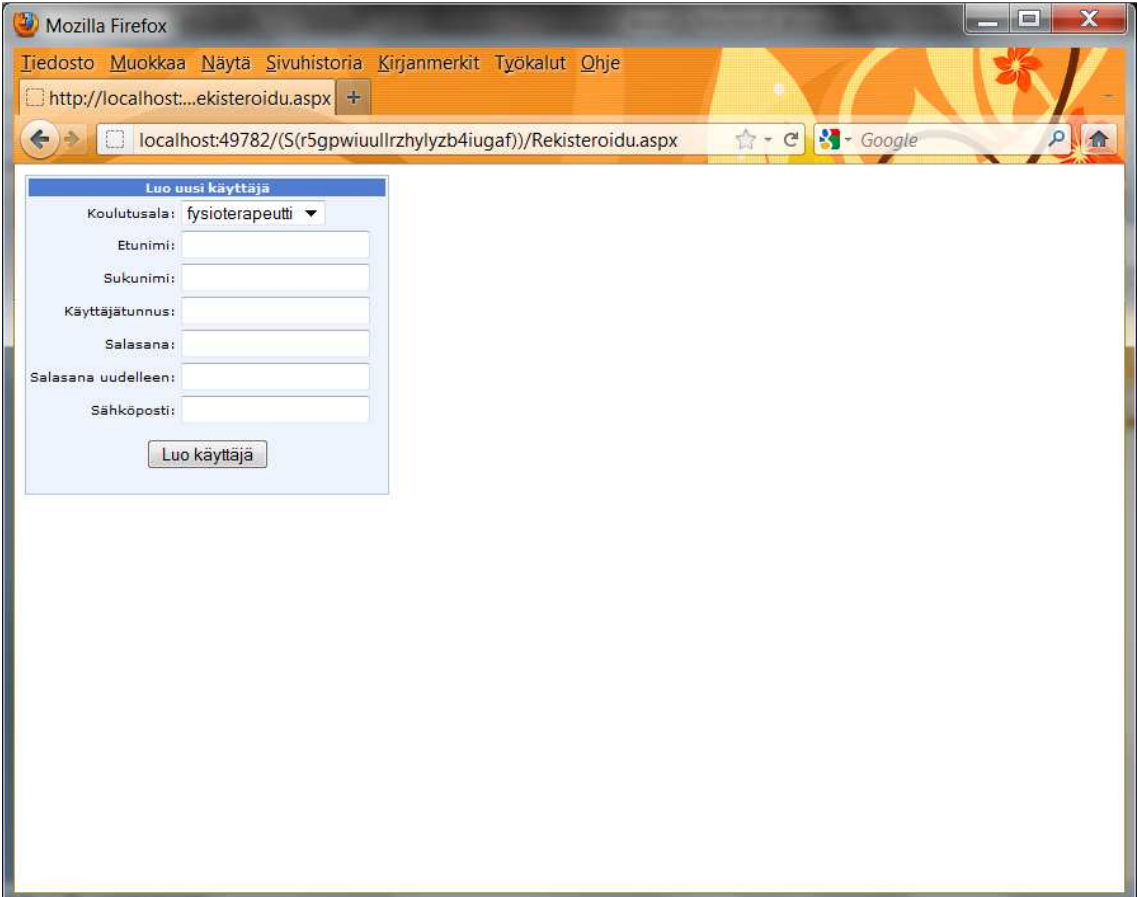
Staattisten muuttujien kanssa tuli ongelmia, sillä ASP.NET:ssä kaikki käyttäjien toimenpiteet kohdistuvat samaan muuttujaan ja muokkaavat sitä ristiin. Ongelma ratkaistiin käyttämällä istuntoja (session) staattisten muuttujien sijaan. Myös globaalit muuttujat toimivat eri tavalla kuin perinteisissä työpöytäsovelluksissa. Tallennettaessa oliota istuntoon on luokan oltava sarjallistuva (serializable). Istuntoon tallennettuja muuttujia ei voi käyttää suoraan LINQ-kyselyissä, vaan ne on ensin tallennettava paikallisiin muuttujiin.

## 6 Yrityspeli

Yrityspelissä pelaaja pääsee kokeilemaan yrittäjyyttä ilman taloudellista riskiä. Tässä luvussa esitellään yrityspelin toteutettuja osia. Ulkoasua ei vielä ole suunniteltu eikä toteutettu. Myös toiminnallisuus on vielä kesken (toivottavasti jatkaja löytyy).

### 6.1 Rekisteröityminen ja kirjautuminen

Ennen pelin aloittamista pelaajan on rekisteröidyttävä ja kirjaututtava järjestelmään. Rekisteröityminen tapahtuu täyttämällä kuvan 6.1 mukaiset tiedot ja painamalla Luo käyttäjä -painiketta.

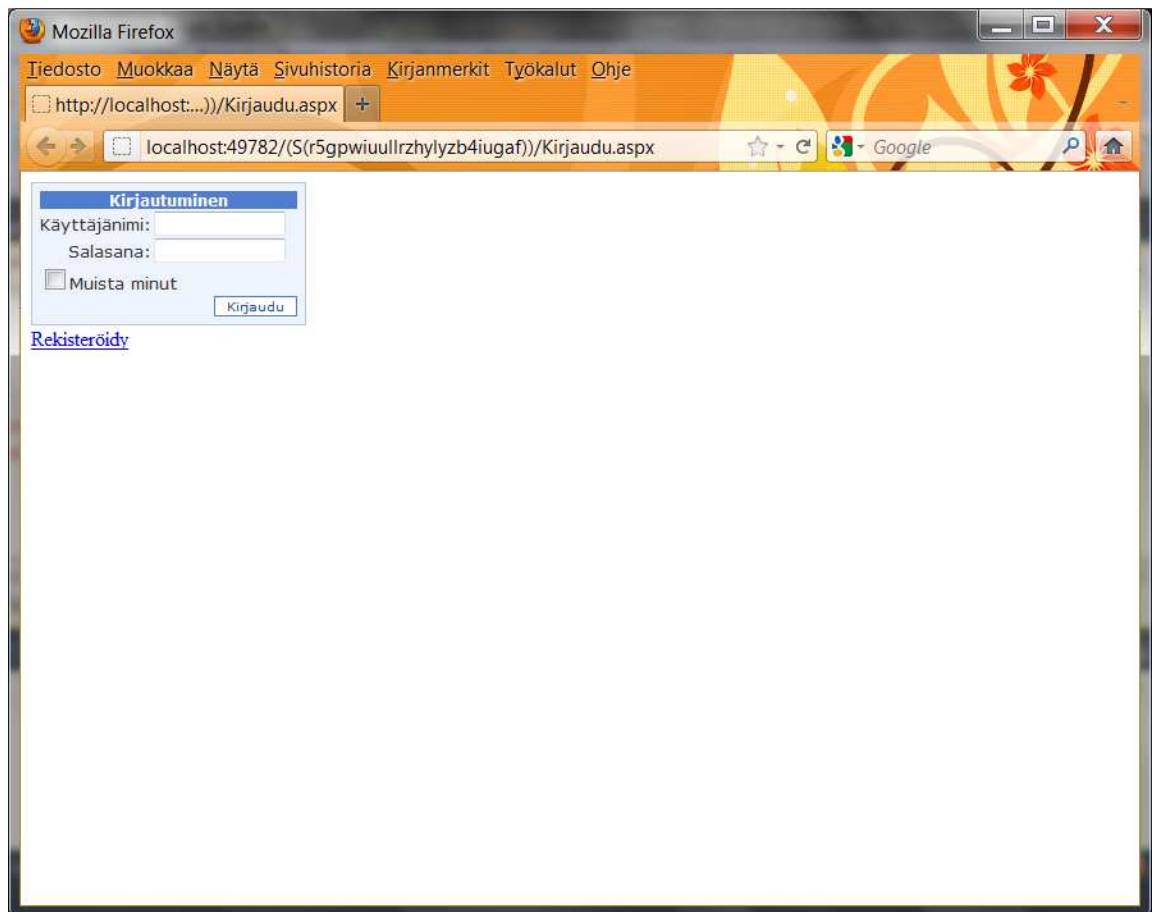


The image shows a screenshot of a Mozilla Firefox browser window. The address bar displays the URL `localhost:49782/(S(r5gpwiuullrzhylyzb4iugaf))/Rekisteroidu.aspx`. The main content area contains a registration form with the following fields and a button:

- Koulutusala:** fysioterapeutti (dropdown menu)
- Etunimi:** text input field
- Sukunimi:** text input field
- Käyttäjätunnus:** text input field
- Salasana:** text input field
- Salasana uudelleen:** text input field
- Sähköposti:** text input field
- Luo käyttäjä** button

Kuva 6.1 Yrityspeliin rekisteröityminen

Tämän jälkeen pelaaja voi kirjautua Yrityspeliin kuvassa 6.2 esitetyn kirjautumissivun kautta.

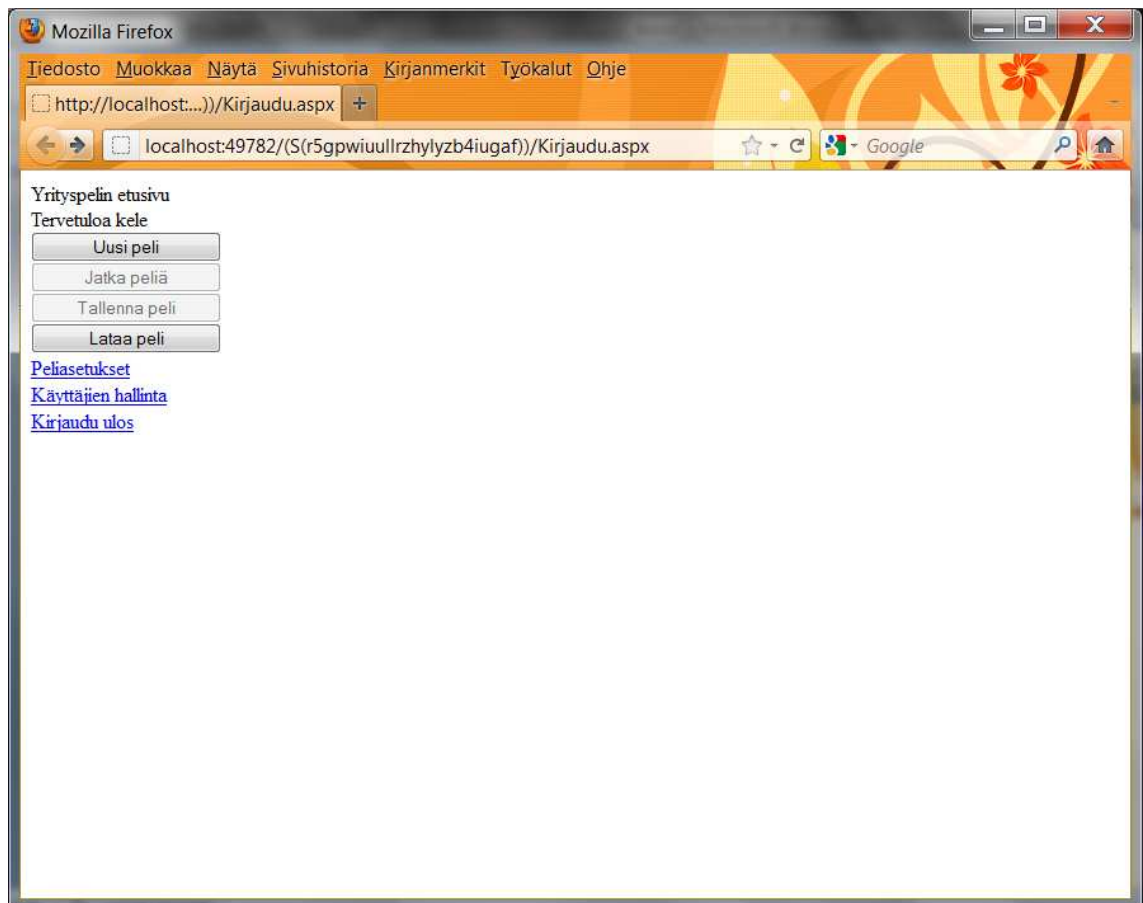


Kuva 6.2 Kirjautuminen

Kirjaututtuaan pelaaja näkee päävalikon ja käyttöoikeuksiensa mukaiset painikkeet.

## 6.2 Päävalikko

Päävalikosta (kuva 6.3) pelaaja voi aloittaa uuden pelin, jatkaa menossa olevaa peliä, tallentaa pelin ja ladata tallennetun pelin. Osa painikkeista tulee käyttöön vasta, kun tietyt ehdot täyttyvät. Esimerkiksi peliä ei voi ladata, ennen kuin peli on tallennettu.



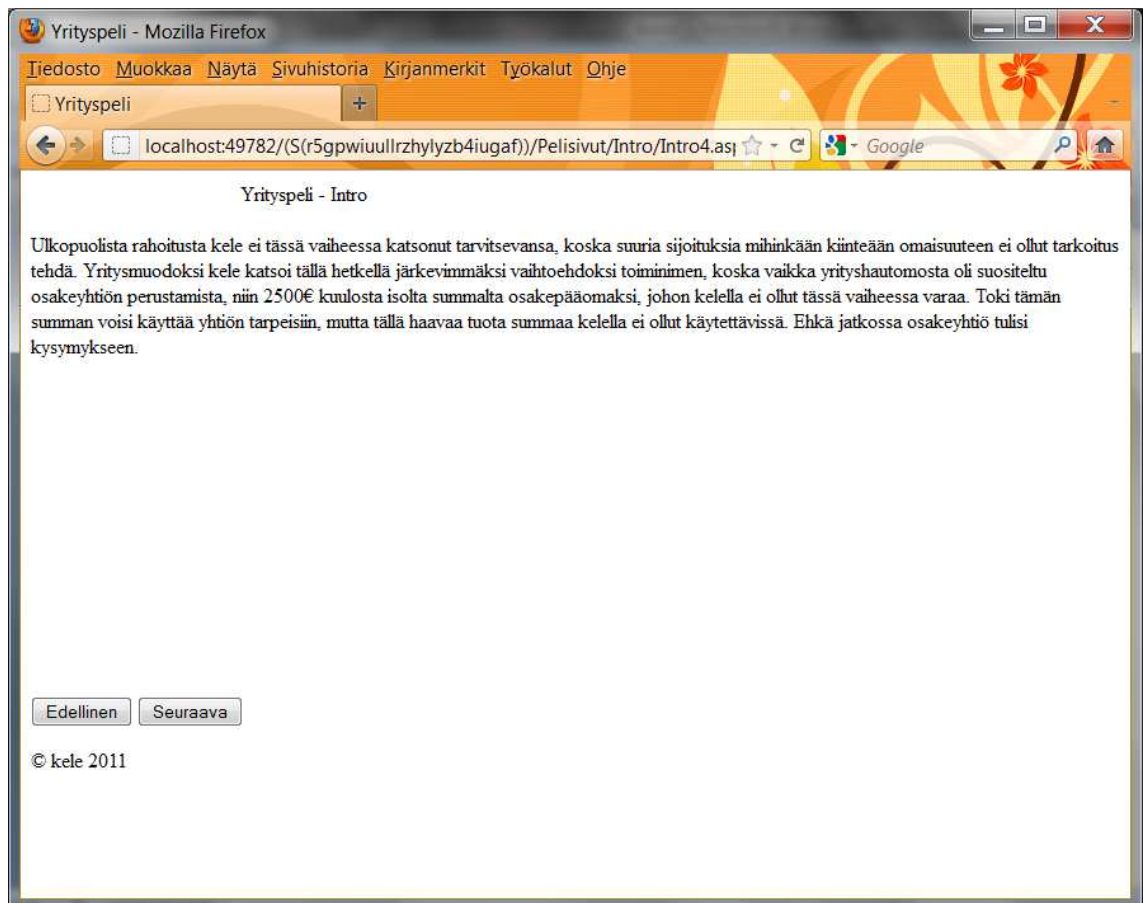
Kuva 6.3 Päävalikko

Lataa peli -painike avaa uuden sivun, jossa kyseisen pelaajan tallennetut pelit näytetään. Pelejä voi olla tallennettuna maksimissaan neljä. Kun yritetään tallentaa viidettä peliä, poistetaan vanhin peli. Opettaja-oikeudet omaavan käyttäjän on mahdollista hallita pelin asetuksia ja muita käyttäjiä. Asetuksia hallitaan erillisellä sivulla.

### 6.3 Intro

Introssa johdatellaan pelaaja peliin ja perustetaan yritys ohjatusti. Pelaaja saa tässä vaiheessa päättää yrityksensä nimen ja opettajan määrittämien asetusten mukaisesti joitakin muita tietoja. Introssa kerrotaan pelaajalle myös taustatietoa. Intro on esitetty kuvassa 6.4.



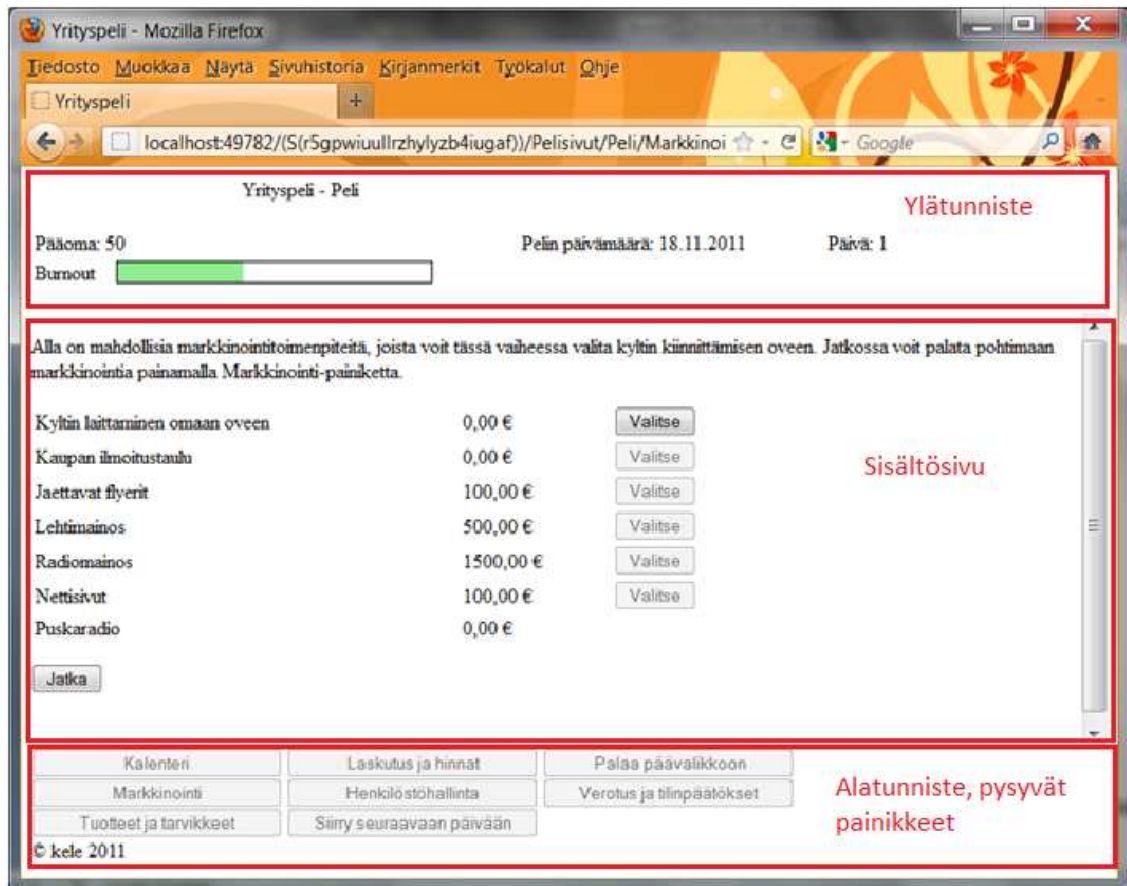


Kuva 6.4 Intro

Intro koostuu useista sivuista ja siinä edetään painamalla Seuraava-painiketta. Pelaajan on myös mahdollista palata aiemmille sivuille. Kun intro on käyty läpi, alkaa varsinainen peli.

## 6.4 Varsinainen peli

Itse peli muodostuu pelaajalle koko ajan näytettävistä painikkeista pelin alatunnisteissa sekä vaihtuvasta sisältösivusta (kuva 6.5). Lisäksi ylätunnisteissa näytetään pelaajan pääoma, pelin päivämäärä, burnout-taso sekä pelissä menossa oleva päivä.



Kuva 6.5 Yrityspelin käyttöliittymä

Pelaaja pelaa pelissä päiväkohtaisia tapahtumia ja hoitaa lisäksi muita yrittäjyyteen liittyviä tehtäviä. Tehtäviä ovat mm. markkinointi, laskutus, verojen tilitys ja palveluiden hinnoittelu. Näitä tehtäviä pelaaja pääsee tekemään alatunnisteen painikkeista, jotka avaavat näytölle uuden sisältösivun. Päiväkohtaisia tapahtumia pelaaja suorittaa sisältöosiin automaattisesti avautuvilla sivuilla ja painikkeilla. Pelin ensimmäisinä päivinä pelaajalle näytetään ohjeita ja vinkkejä, jotta tämä oppisi käyttämään peliä. Myöhemmin näitä vihjeitä ei enää näytetä.

## 7 Yhteenveto ja pohdinta

Yrityspeli-projekti osoittautui haastavaksi ja vaativaksi. Projektin kuluessa suunniteltiin ja toteutettiin Yrityspelin ensimmäinen demoversio. Aivan kaikkia haluttuja ominaisuuksia ei ehditty toteuttaa. Peliin saatiin kuitenkin suunniteltua toimiva arkkitehtuuri, toteutettua pelin ydin sekä suuri osa keskeisimmistä toiminnoista. Valitut tekniikat ja menetelmät osoittautuivat toimiviksi ja tarkoitukseen sopiviksi.

Tässä työssä opiskelin ja esittelin myös peliohjelmoinnin teoriaa. Peliohjelmointi on mielenkiintoinen tietotekniikan osa-alue, vaikka peliohjelmoinnin perusteoria ei täysin olekaan sovellettavissa Yrityspelin kaltaisten opetuspelien tekemiseen. Kiinnostukseni peliohjelmointiin kuitenkin heräsi ja tulevaisuudessa luultavasti teen jonkinlaista peliprojektia ainakin harrastusmielessä.

Opiskeltavaa aiheeseen liittyen riittää vaikka kuinka paljon. Erityisesti ASP.NET:iin liittyvää materiaalia on saatavilla hyvin paljon. Jotta pystyisi valitsemaan parhaan mahdollisen toteutustavan, olisi tähänkin työhön käytetty aika mennyt pelkkään opiskeluun. Projekti on ollut kuitenkin opettavainen, ja seuraavaa projektia on tältä pohjalta huomattavasti helpompi lähteä suunnittelemaan ja toteuttamaan.

Parannettavaa projektissa jäi varmasti monelle osa-alueelle. Esimerkiksi suunnittelu vähäisellä kokemuksella ja käytettäessä uusia tekniikoita ei ollut helppoa. Monissa asioissa oli kokeilemalla etsittävä toimivaa ratkaisua. Yhteistyötä ja keskustelua muun projektiryhmän kesken olisi voinut olla enemmän, jolloin projektiin olisi saatu laajempaa näkökulmaa ja asiat olisivat voineet edetä sujuvammin. Joissakin kohdissa olisi voinut tyytyä yksinkertaisempaan toteutukseen ja yksityiskohtien viimeistelyn jättää myöhemmälle. Virheistä kuitenkin oppii ja myöhemmin oikeiden ratkaisujen löytäminen on helpompaa.

## Kuvat

Kuva 2.1 Saimaan ammattikorkeakoulun organisaatiokaavio (Saimaan ammattikorkeakoulu 2011a).....	8
Kuva 2.2 Saimaan ammattikorkeakoulun uudisrakennus .....	9
Kuva 3.1 Esimerkki toimintapelistä, The Great Giana Sisters (C64s 2011) .....	11
Kuva 3.2 Esimerkki strategiapelistä, Civilization III (Civ3 2003).....	12
Kuva 3.3 Esimerkki urheilupelistä, NHL 12 (EA Sports 2011).....	12
Opetuspelit (kuva 3.4) ovat tarkoitettu opettamaan pelaajalle asioita tekemisen kautta. Opetettava aihe on yleensä rajattu tietyn aihealueen ympärille. Ajatuksena on, että pelin kautta oppiminen on sekä mielenkiintoisempaa että tehokkaampaa. (Manninen 2007, 22.) .....	13
Kuva 3.4 Esimerkki opetuspelistä, Matikkavisa (CognAid 2011).....	13
Kuva 3.5 Peliarkkitehtuurin MVC-malli (Kellomäki 2010a) .....	14
Kuva 3.6 Pääsilmutkan rakenne (Florida Institute of Technology 2009) .....	15
Kuva 4.1 Esimerkki ketterän menetelmän periaatteesta (Ketterät käytännöt 2011).....	18
Kuva 4.2 Microsoft Visual Studio 2010 Premium .....	20
Kuva 4.3 CLR:n peruseriaate (Source ASP.Net 2010) .....	21
Kuva 4.4 Entiteettimalli .....	22
Kuva 4.5 Uniform Server -palvelimen käyttöliittymä.....	24
Kuva 4.6 DayPilot Calendar Lite (DayPilot 2011b).....	24
Kuva 4.7 EO Progress Bar:n eri esitystapoja (Essential Objects 2011) .....	25
Kuva 5.1 Ganttin kaavio yrityspelin työvaiheista .....	27
Kuva 6.1 Yrityspeliin rekisteröityminen .....	30
Kuva 6.2 Kirjautuminen.....	31
Kuva 6.3 Päävalikko .....	32
Kuva 6.4 Intro.....	33
Kuva 6.5 Yrityspelin käyttöliittymä.....	34

## Taulukot

Taulukko 5.1 Yrityspelin organisaatio .....	26
Taulukko 5.2 Alustava työmääräarvio ja aikataulu .....	26

## Lähteet

Box, D. & Hejlsberg, A. 2007. LINQ: .NET Language Integrated Query. Microsoft. <http://msdn.microsoft.com/en-us/library/bb308959.aspx> Luettu 26.9.2011

C64s, 2011. Giana Sisters. [http://c64s.com/game/1397/giana\\_sisters/](http://c64s.com/game/1397/giana_sisters/) Luettu 18.11.2011

Civ3, 2003. Civilization III: Conquests Screenshots. <http://www.civ3.com/images/screenshots/conquests/sengoku2.jpg> Luettu 18.11.2011

CognAid, 2011. Matikkavisa 1. [http://www.opetusohjelmat.fi/kauppa/product\\_details.php?p=7](http://www.opetusohjelmat.fi/kauppa/product_details.php?p=7) Luettu 18.11.2011

DayPilot, 2011a. DayPilot Lite. <http://www.daypilot.org/daypilot-lite.html> Luettu 27.9.2011

DayPilot, 2011b. DayPilot Calendar Lite. <http://www.daypilot.org/calendar-lite.html> Luettu 18.11.2011

DotNetSpark, 2011a. MSIL or IL (Microsoft Intermediate Language). <http://www.dotnetspark.com/tutorial/6-40-msil-il-microsoft-intermediate-language.aspx> Luettu 20.9.2011

DotNetSpark, 2011b. CLR (Common Language Runtime). <http://www.dotnetspark.com/tutorial/6-38-clr-common-language-runtime.aspx> Luettu 20.9.2011

EA Sports, 2011. NHL 12. <http://www.ea.com/fi/nhl-12> Luettu 18.11.2011

Essential Objects, 2011. Free Progress Bar for ASP.NET. <http://www.essentialobjects.com/Products/EOWeb/ProgressBar.aspx> Luettu 18.11.2011

Florida Institute of Technology, 2009. Main Game Loop Architecture.  
<https://cs.fit.edu/proxy/proj/SidusBellum/wiki/Main%20Game%20Loop%20Architecture> Luettu 20.11.2011

Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. Helsinki: Talentum.

Hiltunen, M. 2005. Johdatus tietojärjestelmiin, Kehittämistyön vaiheet ja elinkaarimallit. Oulun kauppaoppilaitos.

[http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien\\_kaytto\\_ja\\_kehittamien/johdatus\\_tietojarjestelmiin/kehittamistyon\\_vaiheet\\_ja\\_elikaarimallit/kehittamistyon\\_vaiheet\\_ja\\_elikaarimallit.htm](http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittamien/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elikaarimallit.htm) Luettu 21.9.2011

Kellomäki, T. 2010a. Peliohjelmointi, Luku 4: Pelinrakenne. Tampereen teknillinen yliopisto. [http://www.cs.tut.fi/~peliohj/peliohj2010-04-Pelin\\_rakenne-webversio.pdf](http://www.cs.tut.fi/~peliohj/peliohj2010-04-Pelin_rakenne-webversio.pdf) Luettu 5.9.2011

Kellomäki, T. 2010b. Peliohjelmointi, Luku 5: Rajapinnat. Tampereen teknillinen yliopisto. <http://www.cs.tut.fi/~peliohj/peliohj2010-05-Rajapinnat-webversio.pdf> Luettu 6.9.2011

Kellomäki, T. 2010c. Peliohjelmointi, Luku 10: Testaus ja optimointi. Tampereen teknillinen yliopisto. <http://www.cs.tut.fi/~peliohj/peliohj2010-10-TestausOptimointi-webversio.pdf> Luettu 8.9.2011

Kellomäki, T. 2010d. Peliohjelmointi, Luku 11: Verkkopelit. Tampereen teknillinen yliopisto. <http://www.cs.tut.fi/~peliohj/peliohj2010-11-Verkkopelit-webversio.pdf> Luettu 19.9.2011

Kellomäki, T. 2011a. Peliohjelmointi, Luku 1: Johdanto. Tampereen teknillinen yliopisto. <http://www.cs.tut.fi/~peliohj/peliohj2011-01-Johdanto-webversio.pdf> Luettu 9.9.2011

Kellomäki, T. 2011b. Peliohjelmointi, Luku 2: Peliteollisuus. Tampereen teknillinen yliopisto. <http://www.cs.tut.fi/~peliohj/peliohj2011-02-Peliteollisuus-webversio.pdf> Luettu 7.9.2011

Ketterät käytännöt, 2011. Scrum - mahdollisuuksien taide.

<http://www.ketteratkaytannot.fi/Menetelmat/Scrum/> Luettu 18.11.2011

Manninen, T. 2007. Pelisuunnittelijan käsikirja Ideasta eteenpäin. Kustannus Oy Rajalla.

Microsoft 2010a. Advanced tools for development teams.

<http://go.microsoft.com/?linkid=9747923> Luettu 19.9.2011

Microsoft 2010b. ASP.NET and Visual Web Developer.

<http://msdn.microsoft.com/en-us/library/dd566231.aspx> Luettu 20.9.2011

Microsoft 2010c. ADO.NET overview. <http://msdn.microsoft.com/en-us/library/h43ks021%28v=VS.100%29.aspx>

Luettu 23.9.2011

Microsoft 2010d. Entity Framework Overview. <http://msdn.microsoft.com/en-us/library/bb399567.aspx>

Luettu 26.9.2011

MySQL, 2011. What is MySQL. <http://dev.mysql.com/doc/refman/5.6/en/what-is-mysql.html>

Luettu 26.9.2011

Saimaan ammattikorkeakoulu, 2011a. Organisaatiokaavio.

[http://www.saimia.fi/fi-FI/images/saimaan\\_amk\\_organisaatio.jpg](http://www.saimia.fi/fi-FI/images/saimaan_amk_organisaatio.jpg) Luettu

18.11.2011

Saimaan ammattikorkeakoulu, 2011b. Saimaan amk lyhyesti.

<http://www.saimia.fi/fi-FI/tietoja-saimaan-amk/saimaan-amk-lyhyesti> Luettu

31.5.2011

Saimaan ammattikorkeakoulu, 2011c. Sosiaali- ja terveysala.

<http://www.saimia.fi/fi-FI/koulutustarjonta/sosiaali-ja-terveysala> Luettu 31.5.2011

Source ASP.Net, 2010. What is Common Language Runtime (CLR)?

[http://www.sourceasp.net/post16/What-is-Common-Language-Runtime-](http://www.sourceasp.net/post16/What-is-Common-Language-Runtime-CLR.aspx)

[CLR.aspx](http://www.sourceasp.net/post16/What-is-Common-Language-Runtime-CLR.aspx) Luettu 20.11.2011

Spaanjars, I. 2010. Beginning ASP.NET 4 in C# and VB. Wiley Publishing, Inc.  
<http://msdn.microsoft.com/en-us/library/gg615484%28v=VS.88%29.aspx> Luettu  
22.9.2011

StartVBDotNet, 2011. Web Forms.  
<http://www.startvbdotnet.com/aspsite/forms/default.aspx> Luettu 22.9.2011

Uniform Server, 2011. The Uniform Server. <http://www.uniformserver.com/> Luettu  
27.9.2011

W3Schools, 2011a. AJAX Tutorial <http://www.w3schools.com/ajax/> Luettu  
27.9.2011

W3Schools, 2011b. [http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp) Luettu  
27.9.2011

Ye, Z. 2000. Designing User Interfaces for Games. Carnegie Mellon University.  
[serenobs.tistory.com/attachment/dk0.pdf](http://serenobs.tistory.com/attachment/dk0.pdf) Luettu 9.9.2011