

Matti Nalkki

**SILVERLIGHT SOVELLUS:
JAPANIN KIELEN ALKEISIIN TUTUSTUMINEN**

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Luonnontieteiden koulutusala
Tradenomi
Syksy 2009



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Luonnontieteiden koulutusala	Koulutusohjelma Tradenomi
Tekijä(t) Jarno Matti Tapio Nalkki	
Työn nimi SILVERLIGHT SOVELLUS: JAPANIN KIELEN ALKEISIIN TUTUSTUMINEN	
Vaihtoehtoiset ammattiopinnot Ohjelmistosuunnittelu	Ohjaaja(t) Veli-Pekka Piirainen Toimeksiantaja Kajaanin ammattikorkeakoulu
Aika 4.12.2009	Sivumäärä ja liitteet 40 + 11
<p>Opinnäytetyön aihe on Silverlight web-sovellus. Tarkoituksena on tutkia mikä Silverlight on ja kuinka sitä käytetään. Apuna käytetään Expression Studiota, jonka tarkoitus on enimmäkseen tukea Silverlight:a. Expression Studio koostuu useasta ohjelmasta, jotka esitellään tässä työssä. Ohjelmointityöhön liittyy japanin kieli, joten japanin kielestä on oma kohtansa teoriaosuudessa. Ohjelmointityönä on tehty web-sovellus, joka tutustuttaa käyttäjän japanin kielen alkeisiin. Alkuun ongelmat koskivat Silverlight:a ja Expression Studio:a, joiden tuntemus oli heikkoa. Japanin kielestä piti keksiä kaikki harjoitukset ja niiden toimivuutta piti testata, samoin kuin sivuston toimivuutta.</p> <p>Työ aloitettiin tutkimalla ja opettelemalla Silverlight:n käyttöä, samalla kun ohjelmointityö oli vielä alkuvaiheessa. Kun Silverlight oli saatu teknillisesti sisäistettyä, voitiin siitä kirjoittaa teoriaa. Ohjelmointityöstä tehtiin lopullinen suunnitelma, kun Silverlight:n käyttö oli hallussa ja selvitettiin mitä Expression Studion ohjelmia käytetään. Toimivan käyttöliittymän valmistuttua, tehtiin japanin kielen harjoitukset, jotka sitten lisättiin sovellukseen. Viimeiseksi muokattiin sovelluksen ulkonäköä.</p> <p>Ohjelman tarkoituksen kannalta, työ onnistui hyvin ja testaajien mielestä se ajaa tarkoituksensa. Sivusto toimii kuten pitääkin ja se saavuttaa tarkoituksensa. Suurimmat moitteet kohdistuvat ulkonäköön, joka vaatii enemmän taiteellista näkemystä. Vaikka sivuston tarkoitus on tutustuttaa käyttäjä japanin kielen alkeisiin, voidaan sitä käyttää hiragana ja katakana merkkien opetteluun.</p>	
Kieli	Suomi
Asiasanat	Silverlight, Expression, Blend
Säilytyspaikka	<input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Business	Degree Programme Business information technology
Author(s) Jarno Matti Tapio Nalkki	
Title SILVERLIGHT APPLICATION: INTRODUCTION TO BASICS OF JAPANESE LANGUAGE	
Optional Professional Studies Programming	Instructor(s) Veli-Pekka Piirainen
	Commissioned by Kajaani University of applied sciences
Date 4.12.2009	Total Number of Pages and Appendices 40 + 11
<p>This thesis deals with the Silverlight web application. The purpose was to research what Silverlight is and how it is used. Expression Studio was used to help develop the application. Expression Studio has several programs which are introduced in this thesis. The programming work concerns a web application that introduces the Japanese language to the user. The theory part has its own section about the Japanese language. Silverlight and Expression Studio proved problematic, since the author's knowledge about them was not good at first. Practice texts and assignments for the Japanese language had to be created from the start, and in addition, their practical usability as well as the functionality of the web-application had to be tested.</p> <p>The work was started by studying the Silverlight application. When Silverlight was familiar to a certain extent, the theory could be written, as well as programming started. When the necessary Expression Studio programs were decided, the final plan for the application was made. When the user interface was completed and operational, the finalized Japanese language exercises were included in the application. Finally, the appearance of the application was modified.</p> <p>The application was a success since the objective of the work was reached. The testers were pleased and said that they learned the basics of Japanese that the application was supposed to teach. The biggest complaints concerned the appearance, in their opinion it would need a more artistic touch. Even though the aim was to introduce the user to the Japanese language, it can also be used to study the hiragana and katakana letters.</p>	
Language of Thesis	Finnish
Keywords	Silverlight, Expression, Blend
Deposited at	<input checked="" type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input type="checkbox"/> Library of Kajaani University of Applied Sciences

SYMBOLILUETTELO

Affiksi

Sanan runkoon liitettävä osa (Sanakirjani.fi 2009).

Ajax (Asynchronous JavaScript and XML)

Ajax on joukko web-sovelluskehityksen tekniikoita (Garret 2005).

ASP.NET

Microsoftin kehittämä ohjelmointikieli, joka on tarkoitettu dynaamisten web-sivujen luomiseen. (RustyBlock 2008).

C# (C sharp)

Olio-orientoitunut .NET ohjelmointikieli (C# Station 2009).

Directshow

Multimedia rajapinta, joka suorittaa monia eri operaatioita mediatiedostoissa ja suoratoistoissa (stream).

Flash

Adobe Flash on multimediaesitysten kehitysympäristö (Adobe 2009).

H.264/MPEG4

Digitaalisen videon pakkausstandardi, joka pakkaa tiedoston pienempään formaattiin sen varastoimista ja välittämistä varten (Vcodex 2009).

IIS Media Service

http-pohjainen mediatoimintaratkaisu (CMS WiRE 2009).

IronRuby ja IronPython (Ruby ja Python)

Dynaamisia ohjelmointikieliä, jotka suorittavat toimintoja ajon aikana (ScottGu 2007).

Isolaattikieli

Isolaattikieliä ei voida sijoittaa mihinkään kielikuntaan, eli ne muodostavat oman kielikuntansa (Silja 2009).

JavaScript

Komentosarjakieli, joka on tarkoitettu web-ympäristöön käytettäväksi ja sen avulla saadaan web-sivuille dynaamista toiminnallisuutta (Sun Microsystems 2009).

RIA (Rich Internet Application)

Internet-sovellus joka muistuttaa työpöytäsovellusta (Small Business Computing.com 2009).

User Control

Pohja johon käyttöliittymä kootaan. Tämän päälle lisätään esim. nappeja ja tekstikenttiä. (msdn 2009.)

VC-1

Video Codec 1 on videonpakkausalgoritmi ja se toimii mm. Blu-rayn standardina (Loomis, ym. 2007).

Visual Basic

Microsoftin kehittämä ohjelmointikieli, joka perustuu Basic-kieleen (Small Business Computing.com 2009).

WMA (Windows Media Audio)

Microsoftin kehittämä digitaalisen äänitiedoston formaatti. (Small Business Computing.com 2009).

WMV (Windows Media Video)

Microsoftin kehittämä kompressoitun videotiedoston formaatti (Loomis, ym. 2007).

WPF

Windows Presentation Foundation, muodostaa Windows Vistan graafisen rajapinnan. Toimii käyttöliittymän pohjana ja ohjelmoitavana rajapintana (WindowsClient.NET 2009).

WYSIWYG

Lyhenne sanoista ”what you see is what you get”. WYSIWYG on editori tai ohjelma jolla nähdään miltä tekeillä oleva käyttöliittymä tai dokumentti tulee näyttämään. (Whatis?com.)

XAML (eXtensible Markup Language)

XML-pohjainen kieli, johon käyttöliittymä kirjoitetaan. Tämän kanssa ohjelmointikielenä voi käyttää joko Visual Basic- tai C#-ohjelmointikieltä. (Moroney 2008, 7.)

ALKUSANAT

Aihe tuli mieleen alun perin siitä, että opiskelin japanin kieltä itsenäisesti ja halusin kehitellä tavan jolla oppia japanin kielen hiragana ja katakana merkit. Ohjelman toteutus ja muu sisältö puuttui, mutta sain pian neuvon tehdä sivuston joka opettaa japanin kieltä, käyttäen Microsoft Silverlight:a ja Microsoft Expression Studion ohjelmia. Minulla ei ollut aiempaa kokemusta Silverlight:sta ja en ollut edes kuullut Expression Studio:sta, joten opettelini samalla niiden käyttöä. Itse sivuston päätin tehdä enimmäkseen käyttäen Visual Studiota ja tietyt osat tein Expression Blend:n avulla. Alkuun eniten aikaa vei ohjelmoinnin sisäistäminen, jonka paremmin opittuani, aloitin koko työn pari kertaa alusta. Hyvin pian sisällön kehittämisenstä tuli eniten aikaa vievä työ. Minulla ei ollut millä kirjoittaa japanin kielen merkkejä, joten jouduin sitä varten tekemään virtuaalinäppäimistön. Alun perin ajattelin että sivuston avulla käyttäjä voi opetella japanin kielen alkeet, mutta se pian ilmeni liian suureksi työksi ja vähensin työmäärää siten että sivusto toimii tutustumisvälineenä japanin kielen alkeisiin.

SISÄLLYS

1 JOHDANTO	1
2 MICROSOFT SILVERLIGHT JA EXPRESSION STUDIO	3
2.1 Silverlight ja Flash	6
2.2 Silverlight ja XAML	8
2.3 Media Element	11
2.4 Microsoft Expression Studio	12
2.5 Microsoft Expression Blend	12
2.6 Microsoft Expression Web	13
2.7 Microsoft Expression Design	14
2.8 Microsoft Expression Encoder – VC-1	15
3 JAPANIN KIELI	16
4 TUTORIAALI	20
5 OHJELMOINTITYÖ JA SEN TOTEUTUS	31
5.1 GUI, graafinen käyttöliittymä	31
5.2 Toteutustekniikka	35
5.3 Luokat ja metodit	35
5.4 Testaus	37
5.5 Käyttäjien kokemus	37
6 ANALYYSI JA JOHTOPÄÄTÖKSET	38
7 YHTEENVETO	39
LÄHTEET	40
LIITTEET	

1 JOHDANTO

Japanin kulttuurin suosio on ollut kasvussa Suomen nuorten keskuudessa. Japanilaiset animaatiot ja sarjakuvat ovat saavuttaneet suuren suosion ja se on herättänyt mielenkiintoa Japania kohtaan. Ihmiset ovat kaivanneet kouluihin japanin kielen kursseja, mutta opettajien saatavuus on ollut liian vähäistä. Opinnäytetyöksi päätettiin tehdä sovellus, jolla voi opetella itsenäisesti japanin kieltä. Toimeksiantaja on Kajaanin ammattikorkeakoulu.

Alkuun sovelluksen oli tarkoitus opettaa japanin kielen alkeet, mutta se on liian laaja alue, joten sovelluksesta tehtiin japanin kielen tutustumisväline. Toimeksiantajan toiveesta, ohjelmointityönä tehdään Silverlight web-sovellus, jossa käytettiin apuna Expression Studio ohjelmistoa. Silverlight ja Expression Studion ohjelmistot ovat uusia ja niihin liittyvää opinnäytetyötä ei toimeksiantajalle oltu vielä tehty. Tavoitteena oli tutkia Silverlight:a ja Expression Studion ohjelmia ja kehittää Silverlight web-sovellus.

Tässä työssä esitellään Silverlight ja Expression Studio ohjelmistot. Silverlight:sta on esittelyä ja perustiedon lisäksi, kerrottu sen versioista ja XAML:stä. Expression Studio:sta kerrotaan sen jokaisesta ohjelmasta erikseen. Teoriaosuudessa on oma osuutensa japanin kielestä, joka on oleellinen osa opinnäytetyötä. Käytännön osio sisältää ohjelmaan liittyvän sisällön lisäksi Silverlight tutoriaalin, joka seuraa ohjelmointityön tyyliä. Koko työstä on tehty analyysi, joka sisältää omia mietteitä ja arviointia sekä palautetta testaaajilta.

Tavoitteena opinnäytetyössä oli tutustua Silverlight:iin ja Expression Studioon ja opetella niiden käyttöä. Molemmat ovat suhteellisen uusia, joten tieto niistä oli vähäistä. Opinnäytetyö piti aloittaa tutustumalla ja opettelemalla edellä mainittuja ohjelmistoja. Tämä tapahtui samaan aikaan, kun ohjelmointityötä kehitettiin. Kun Silverlight oli saatu sisäistettyä, saattoi web-sovelluksesta tehdä lopullisen suunnitelman. Ohjelman tavoitteena on opettaa japanin kielen alkeita, joten siitä on pyritty tekemään suhteellisen nopea käyttää, helppolukuinen ja miellyttävän näköinen.

Silverlight:n käyttöä opeteltiin käymällä läpi tutoriaali oppimateriaaleja sekä harjoittelemalla. Ohjelmointi tehtiin uusimmalla Silverlight 3 versiolla ja Expression Blend:llä. Japanin kielestä kerrotaan hieman sen perusteita, mutta ei perehdytä syvällisesti kielioppiin. Ohjelmointityö keskittyy enimmäkseen opettamaan käyttäjälle japanin kielen hiragana ja katakana merkkejä. Ohjelmasta kerrotaan sen käyttöliittymästä, toteutustekniikasta, toiminnoista, testauksesta ja

kuinka hyvin työ onnistui. Testaus osioon on myös kerätty käyttäjiltä saamaa palautetta ja mitä muutoksia heidän ehdotuksistaan tehtiin.

2 MICROSOFT SILVERLIGHT JA EXPRESSION STUDIO

Microsoft Silverlight on Adoben Flash-multimediatekniikan kilpailija. Silverlight toimii selaimessa plug-in osana, eli liitännäisenä. Silverlight renderöi XAML:än, joka määrittelee sovelluksen ulkonäön ja toimii myös ohjelmointirajapintana. Silverlight:n avulla kehittäjä ja suunnittelijat voivat tehdä yhteistyötä, rakentaessaan Internet-sovelluksia, jotka ovat laadultaan samaa tasoa kuin työpöytäsovellukset (Moroney 2008, xiv.). Silverlight:n liitännäinen on alustariippumaton ja selainriippumaton. Liitännäisen lisäksi, sen tarkoitus on tarjota laadukkaampaa ja tehokkaampaa mediakokemusta ja parempia Internet-sovelluksia (Moroney 2008, 3.). Silverlight käyttää apunaan Expression Studion tarjoamia tekniikoita, jotka keskittyvät mm. grafiikkaan, videoiden suoratoistoon ja kuvien selailuun. Sovelluksia voidaan luoda käyttäen eri ohjelmankehitysympäristöjä, kuten Visual Studio, Expression Studion ohjelmat tai käyttäen näiden yhdistelmää. Visual Studiossa voidaan ohjelmoida Visual Basic- tai C#-ohjelmointikielellä. Silverlight on siis kehittäjiä varten, jotka haluavat luoda rikkaita Internet-sovelluksia ja multimediasisältöä. RIA (Rich Internet Application) tarkoittaa rikkaita Internet-sovellusta, joka muistuttaa toimivuudeltaan ja ominaisuuksiltaan työpöytäsovelluksia. (Microsoft Silverlight 2009.)

Web-sovelluksien kehittyessä tarvittiin uutta tekniikkaa tukemaan niitä. Ajax on viimeisin ratkaisu, jolla voidaan näiden sovelluksien vaatimukset täyttää tehokkaasti ja sujuvasti. Alkuun käytettiin JavaScript:a Ajax:n hallintaan. JavaScript koodi on vaikeasti ylläpidettävää ja sovellukset näyttävät erilaisilta selaimesta riippuen tai ne eivät saata toimia joissakin selaimissa lainkaan. ASP.NET:in myötä Ajax:n hallinnasta on tullut helpompaa. Silverlight luotiin hyödyntämään ASP.NET:ä ja sillä tehtyjen sovelluksien on tarkoitus toimia ja olla samanlaisia eri selaimissa. (Sai Stuff to Developers 2007.)

Version 2 myötä, Silverlight tukee dynaamisia kieliä (Dynamic Languages). Nämä ovat ohjelmointikieliä, joiden toiminnot suoritetaan ajon aikana. Silverlight tukee kolmea suosituinta dynaamista kieltä: Ruby, Python ja JavaScript. Jokainen vaatii oman opettelunsa ja ne poikkeavat hyvin paljon C#:sta ja Visual Basic:sta. (Moroney 2008, 313.)

NBC:n sivu (<http://www.nbcolympics.com/>) on tehty Silverlight tekniikalla. Kyseinen sivu sisältää monia erilaisia toimintoja joiden avulla voi saada hyvän käsityksen Silverlight:n mahdollisuuksista. Sivulla huomattavimpia Silverlight:n toimintoja ovat kuvat ja videot, joita voi-

daan selata liukuvalla ikkunalla. Sivuilta löytyy myös monia interaktiivisia toimintoja. Microsoftin kehittäjä sivulta (<http://blogs.msdn.com/tims/archive/2007/07/07/from-a-to-z-50-silverlight-applications.aspx>) löytyy listattuna useita Silverlight sovelluksia, joista näkee mitä kaikkea Silverlight:lla voidaan tehdä. Blue Rose Games sivuilla (<http://www.blurosegames.com/brg/default.aspx>) on mm. Silverlight:lla tehtyjä pelejä sekä ohjeet pelin tekemiseen Silverlight:lla.

Internet-sisällön kehittyessä, käyttäjät ovat alkaneet vaatia tehokkaampia sovelluksia. Silverlight:n avulla voidaan luoda kauniita ja helppokäyttöisiä sivustoja, sovelluksia ja multimediaa, jotka muistuttavat työpöytäsovelluksia. Vaikka sivusto onkin luotu Silverlight:lla, sen käyttö ei takaa nopeutta tai välttämättä hidasta toimintaa. Sivuston nopeus riippuu sen sisällön raskaudesta ja ohjelmoinnista. Silverlight:lla tehdään myös sovelluksia, jotka voidaan asentaa omalle koneelle. Näitä sovelluksia kutsutaan ”Out of Browser” nimellä. Koneelle asennettuja sovelluksia voidaan lisäksi päivittää automaattisesti tai manuaalisesti. Sovellus voi myös tarkistaa onko verkkoyhteys luotu, päivittää itseään ja omata pääsyn eristettyyn tallennustilaan (Isolated Storage). Näitä sovelluksia voidaan avata pikakuvakkeella ja ne toimivat selaimen ulkopuolella. Lisäksi sovellus on turvallinen ympäristö, joka on helppo asentaa ja se päivittää itsensä automaattisesti luotuaan verkkoyhteyden. (Microsoft Silverlight 2009.)

Silverlight tarjoaa mahdollisuuden rakentaa 3D grafiikoita ja animaatioita. 3D malleja voidaan pyörittää ja skaalata ilman että tarvitsee kirjoittaa riviäkään koodia. Pixel Shader Effects mahdollistaa kuvan sumenemisen ja varjon luomisen. Pixel Shader Effect:llä on mahdollista kirjoittaa omia efektejä, jotka voidaan liittää mihin tahansa graafiseen sisältöön. Bitmap Caching parantaa renderöintiä kuvaa suurennettaessa. Animointi muistuttaa jo Flashin tasoa ja Silverlight:n viimeisin lisäys animointiin on matemaattiset funktiot. (Microsoft Silverlight 2009.)

Silverlight 1, kehitettiin nimen Windows Presentation Foundation/Everywhere (WPF/E) alla. Ensimmäinen versio julkaistiin 4.9.2007. Silverlight 1 koostui seuraavista komponenteista:

- Input, kerää tiedot laitteista, kuten näppäin ja hiiri.
- UI core, hallitsee bitmap kuvien, vektorigrafiikkojen, tekstien ja animaatioiden renderöintiä.

- Media, on MP3, WMA standardien, WMV7, WMV8 ja WMV9/VC-1 suoratoistoon (stream).
- XAML, sallii käyttöliittymän kehittämisen XAML kielellä. (Microsoft Corporation 2009.)

Alkuvaiheessa Silverlight oli saatavilla vain Windows ja Mac käyttöjärjestelmille ja Linux versio oli kehitteillä. Silverlight 1 on saatavilla mm. selaimille Internet Explorer, Mozilla Firefox ja Opera. (Loponen 2007.)

Silverlight 2 ilmestyi 14.10.2008. Merkittävin uudistus oli .NET tuen lisääminen. Muita uudistuksia olivat mm.:

- .NET Framework tuki monipuolisella perusluokkakirjastolla.
- Sisäänrakennetut kontrollit, mm. DataGrid, ListBox, Slider ja ScrollViewer.
- Sivupohjan muokkaaminen. Sovelluksen ulkonäön ja tuntuman muokkauksesta tuli helpompaa.
- Deep zoom, jonka avulla voidaan tarkentaa kuvia.
- Tukee useampia ohjelmointikieliä, kuten C#, Visual Basic, JavaScript, IronPython ja IronRuby.
- Tuki useille alustoille ja selaimille. Tähän kuuluvat Mac, Linux ja Windows, selaimilla Firefox, Safari ja Internet Explorer. (Microsoft Corporation 2009.)

Silverlight:iin oli lisätty JavaScriptiä ja CSS:ää hyödyntäviä toimintoja. Silverlight 2 on saatavilla mobiililaitteisiin, Macille ja Linuxille. (Lehto 2008.)

Viimeisin versio, Silverlight 3 julkaistiin 10.7.2009. Versiossa 3 tuli mahdolliseksi asentaa sovellukset koneelle, jolloin niitä saattoi käyttää ilman selainta. Teräväpiirtomedian toisto on yksi merkittävimmistä lisäyksistä, sillä tämä sallii sellaisten tiedostojen kuin H.264 video, AAC audio ja MPEG-4 sisällön toiston. IIS Media Service on ilmainen selaintuote, jolla Silverlight:n suoratoisto (stream) toteutetaan http:ssä. Silverlight 3 toi myös mukanaan yli 50 uudistusta ja yli 100 uutta kontrollia. (ScottGU 2009, Microsoft Silverlight 2009, Jukka 2009.).

2.1 Silverlight ja Flash

Version 1 ilmestyessä Silverlight:lla oli vaikeuksia Flash:a vastaan, koska Flash:lla oli jo suuri määrä kannattajia ja se oli teknillisesti huomattavasti edellä Silverlight:a. Silverlight pysyi ki- sassa mukana, koska se tarjosi kehittäjille ja suunnittelijoille Flash:ssa kaivattua sisältöä, esi- merkiksi hakukoneen. Animoinnissa Flash käyttää frame:ja (kehys/raami), eli jos halutaan liikuttaa objekti paikasta A paikkaan B kolmessa sekunnissa, lasketaan kuinka monta frame:a kolmeen sekuntiin luodaan ja kaikille frame:ille lasketaan matriisit. Silverlight käyttää WPF animaatio mallia, joka on aikaan perustuva. Siinä asetetaan aloitus- ja päättymismääritykset, eikä käyttäjän tarvitse laskea objektien matriiseja, sillä Silverlight laskee ne automaattisesti. Flash käyttää kompressoitua formaattia ja sen tekstit ja kuvat ovat sulautettu videoon, joten sen koko on suhteellisen pieni. Silverlight taas käyttää XAML:ää kielenään, jota ei ole komp- ressoitu, joten se on kooltaan suurempi. Koodaukseen Flash käyttää ActionScript:ä, joka si- sältää laajan luokkakirjaston ja kykenee käyttämään muita kieliä ja Framework:ja. Silverlight käyttää myös useita kieliä, joista suosituimmat ovat C#, Visual Basic ja JavaScript. Silverlight tukee myös .NET Framework:ja. Flash tukee useita videoformaatteja, kun taas Silverlight tukee vain VC – 1, WMV ja WMA formaatteja. Flash:lla on laajat käyttömahdollisuudet ih- misille joilla on kuulemis- tai näkemisvaikeuksia tai jotka käyttävät mielellään pikanäppäimiä. Flash:n avulla käyttäjä voi katsella videoita ja hallita niiden kulkua näppäimistöllä. Silverlight 3 on ensimmäinen selain plug-in (liitännäinen), joka tarjoaa kaikki järjestelmän värit, helpot- taen heikkonäköisten toimintaa. Flash tukee seuraavia käyttöjärjestelmiä:

- Windows XP/Vista/2000
- Windows Server 2003/2008
- Mac OS 10.1/10.5 (PowerPC)
- Mac OS 10.1/10.5 (Intel)
- Linux 5
- openSUSE 11
- Ubuntu 7.10

Silverlight tukee käyttöjärjestelmiä:

- Windows Vista/XP/2000
- Windows Server 2003/2008

- Windows Mobile 6
- Mac OS 10.1/10.5 (PowerPC)
- Mac OS 10.1/10.5 (Intel)

Flash tukee kaikkia kuvaformaatteja, kun taas Silverlight tukee vain PNG ja JPG formaatteja (Muhammed 2009). Seuraava taulukko on suora lainaus Muhammed Usama Alamin arvostelusta (taulukko 1.):

Features	Flash	Silverlight
Animation		better
File size	better	
Scripting		better
Video/Audio		better
Sound processing	better	
Accessibility	better	
Platform compatibility	better	
Text representation/SEO		better
Supported image formats	better	
Socket programming	better	
Webcam support	better	
Deployment	better	
Windows application	better	
Media streaming		better

Taulukko 1. Vertailu Flash:sta ja Silverlight:sta (Muhammed 2009).

Nämä tiedot eivät enää täysin pitää paikkaansa, sillä Silverlight:ssa on tapahtunut paljon muutoksia tiuhaan tahtiin. Edellä oleva vertailu on tehty jonkin aikaa Silverlight 3 julkaisun jälkeen, mutta nyt vuoden 2009 lopulla monet vertailun kohteet eivät enää pidä paikkaansa. On vaikeaa arvioida juuri nyt näitä kahta keskenään, sillä näinä päivinä Silverlight:n 4 versioita on ilmestymässä beta-versio, jonka on sanottu tuovan paljon uudistuksia. Flash on tois-taiseksi voiton puolella ja syystäkin, sillä Flash:llä on ollut enemmän aikaa kehittyä.

2.2 Silverlight ja XAML

XAML on XML:ään perustuva kieli, joka määrittelee sovelluksen ilmeen. Kohteet, joita XAML käsittelee, ovat UI eli käyttöliittymä, grafiikka, media, kontrollit ja animaatiot. Jokaisella XAML dokumentilla on juurielementti, esim. Canvas tai Grid, johon käyttöliittymä tulee. Oletuksena XAML:ssä on käytössä tietyt nimiavaruudet (namespace) joita Silverlight tarvitsee toimiakseen (Moroney 2008, 7 – 8.). Alla oleva esimerkki sisältää tarpeelliset nimiavaruudet (lihavoituna) ja juurielementtinä toimii Grid.

```
<UserControl x:Class="ProjektinNimi.UserControlNimi"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" >
  <Grid x:Name="LayoutRoot" Background="White">

      Sisältö

  </Grid>
</UserControl>
```

Yllä olevaan “SISÄLTÖ” kohtaan rakennetaan käyttöliittymä. Elementin sisällä olevia elementtejä kutsutaan nimellä Children, joita ovat esim. napit ja tekstikentät. (Moroney 2008, 7 – 8.)

Silverlight XAML tukee erilaisia muotoja, joita voidaan käyttää monimutkaisempien objektien rakentamiseen. Joitain perusmuotoja ovat mm:

- Rectangle on suorakulmaisiin muotoihin
- Ellipse on ellipsin ja ympyrän tekoon
- Line on suora viiva
- Polygon on monikulmaisia muotoja varten
- Polyline piirtää useita viivasegmenttejä
- Path piirtää epälineaarisesti (Moroney 2008, 9.)

Objekteja voidaan muokata mm. seuraavilla toiminnoilla:

- RotationTransform pyörittää elementtiä tietyn asteen verran
- ScaleTransform venyttää tai kutistaa objektia
- SkewTransform laajentaa objektia määrättyyn suuntaan tietyn verran

- `TranslateTransform` liikuttaa objektia annetun vektorin mukaisesti
- `MatrixTransform` käytetään matemaattiseen muokkaukseen, joka voi yhdistää kaikki mainitut (Moroney 2008, 10.)

Alla olevassa esimerkissä venytetään `Stack Panel` kontrollia positiiviseen X suuntaan, eli tässä tapauksessa oikealle. Tällöin `Stack Panel` ja sen sisältö näyttää olevan kallellaan.

```
<StackPanel.RenderTransform>
    <TransformGroup>
        <SkewTransform AngleX="50"/>
    </TransformGroup>
</StackPanel.RenderTransform>
```

Elementin koko määritellään `Height` ja `Width` arvoilla. `Stretch` määrittelee kuinka elementtiä venytetään, esimerkiksi kuva. `Stretch` on arvoltaan `none`, `uniform`, `uniform to fill` tai `fill`. `None` ei venytä kuvaa lainkaan, vaan kuva säilyttää oletusarvonsa. `Uniform` pitää kuvakoon oletusarvon mukaisena tai annetun `Height` ja `Width` arvon kokoisena. `UniformToFill` venyttää kuvan annetun rajojen mukaisesti, säilyttäen oman kuvasuhteensa. `Fill` venyttää tai pienentää kuvaa täyttämään koko alueen, vaikka oma kuvasuhde ei säily. (Moroney 2008, 250 – 251.)

XAML:ssä tehdään ulkonäköön liittyvät säädökset. Kaikilla `NIMI.xaml` dokumenteilla on `NIMI.xaml.cs` dokumentti, jonne tarpeellinen koodi tehdään, esim. mitä tehdään kun nappia painetaan. XAML:ssä näkyy myös design pohja, josta näkee miltä työ näyttää. Kyseiseen pohjaan ei voi tehdä suoraan mitään, vaan se päivittyy sitä mukaan kun XAML:ään kirjoittaa. Designer pohjassa ei myöskään nähdä miten `NIMI.xaml.cs`:ssä oleva koodi vaikuttaa ulkonäköön.

XAML:ssä elementtien pohjana käytetään `Canvas`, `Grid` tai `Stack Panel` kontrollia. Jokainen sisältää oman tyylinsä, kuinka sisältö siihen asetellaan. `Canvas` pohjassa asetelu on esimerkiksi:

```
<Button Canvas.Top="10" Canvas.Left="10"/>
```

Esimerkin mukaan, nappi asetetaan `Canvas` pohjan vasemmasta ylänurkasta, eli 0 pisteestä, 10 pistettä oikealle ja 10 alas. `Grid`:ssä määritellään montako pysty- ja vaakariviä se sisältää ja sisältöön määritellään mihin ruutuun se kuuluu, esimerkiksi:

```
<Button Grid.Row="1" Grid.Column="0"/>
```

Stack Panel järjestää sisällön automaattisesti vierekkäin tai allekkain, riippuen kumpi tyyli sille on määrätty. Kaikille elementeille on vielä olemassa Vertical- ja Horizontal Alignment, jotka määrittelevät kuinka kyseinen elementti asetellaan alueen sisään. XAML kontrollit sisältävät monia ominaisuuksia, joista yleisimmät ovat:

- Font family, font size, font source, font stretch, font style, font weight ja foreground, jotka vaikuttavat fonttiin, muuttamalla väriä, tyyliä, kokoa, jne.
- Text ja Content, joihin lisätään kontrollissa näkyvä teksti.
- Edellä mainitut asetteluun tarkoitetut Canvas.Top ja Grid.Row ominaisuudet.
- Background, johon määritellään mahdollinen väri, väriyhdistelmä tai kuva.
- Height ja width, määrittelee pituuden ja leveyden.
- Opacity, määrittelee läpinäkyvyyden, normaali 1.0 - näkymätön 0.0.

Väriin liittyviä ominaisuuksia ovat esimerkiksi foreground, solid color brush ja linear gradient brush. Foreground määrittelee fontin värin, solid color brush täyttää alueen jollakin tietyllä värillä, linear gradient brush on väriyhdistelmä. Image brush asettaa kuvan alueelle. Kuvan Stretch ominaisuuteen määritellään kuinka se näkyy. Fill täyttää koko alueen, uniform käyttää annettuja mitta-arvoja, uniform to fill venyy koko alueelle, säilyttäen alkuperäisen kuvasuhteen ja none säilyttää kuvan oletuskoon. Silverlight sisältää monia muitakin kontroleja, kuin vain edellä mainitut. Näitä ovat mm. text block, text box, button, radio button, check box ja item list. Text block on tekstikenttä joka on vain lukemista varten, text box:iin voi kirjoittaa, button on nappi, radio button:t ovat joukko vaihtoehtoja, joista käyttäjä voi valita vain yhden kerrallaan, check box on ”rasti ruutuun” kontrolli ja item list on lista, jossa voi olla melkein mitä vain, kuten myytäviä tuotteita. (Moroney 2008, 73 – 75, 250 – 251.)

Style:n (tyyli) avulla kaikille kontrolleille voidaan antaa jonkinlainen tyyli. Style voidaan kirjoittaa projektin App.xaml dokumenttiin tai suoraan sitä tyyliä tarvitsevaan dokumenttiin. Style on tehokas apu, jos haluaa sovelluksen kykenevän muuttamaan ulkonäköään lennosta, esim. napit voivat muuttua neliöstä palloiksi, fontin kokoa voidaan muuttaa tai taustakuva voidaan vaihtaa. Jos haluaa saada tietyn tyylin kaikille sivuille, kannattaa se silloin kirjoittaa App.xaml dokumenttiin. Jos tyyli on vain käytössä yhdellä sivulla, voi sen kirjoittaa joko ky-

seisen sivun XAML dokumenttiin tai projektin App.xaml dokumenttiin. Tyyli annetaan kontrollille kirjoittamalla sen Style ominaisuuteen haluttu tyyli, esimerkiksi:

```
<TextBlock Style="{StaticResources TextBlockStyle1}">
```

Yllä mainittu esimerkki hakee kyseisen tyylin App.xaml:stä ja antaa sen TextBlock:lle. Esimerkkinä mitä ohjelmointityössä on käytetty:

```
<Style x:Key="TextBlockStyle1" TargetType="TextBlock">
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontWeight" Value="Bold"/>
    <Setter Property="FontFamily" Value="Comic Sans MS"/>
</Style>

<Style x:Key="StyleBackground2" TargetType="Grid">
    <Setter Property="Background" Value="#FFF4AE0F"/>
</Style>
```

Ensimmäinen Style muuttaa Text Block:n tyyliä, antamalla sille uuden tekstikoon, muuttamalla tekstin paksummaksi ja vaihtamalla fonttia. Toinen muuttaa Grid kontrollin pohjaväriä.

2.3 Media Element

MediaElement:n avulla voidaan rakentaa ja muokata sovelluksessa olevia ääniä ja videoita. MediaElement tukee videoformaatteja WMV1, WMV2 WMV3, WMVA ja WMVC1 sekä audioformaatteja WMA7, WMA8, WMA9 ja MP3. Jotta video saadaan näkyviin, kirjoitetaan MediaElement kontrollin Source ominaisuuteen videon URL osoite. Näytön kokoa voidaan muuttaa samaan tyyliin kuin kaikkia muitakin elementtejä XAML:ssä, eli määrittämällä Height ja Width arvon. (Moroney 2008, 247 – 251.)

```
<MediaElement Source="video.wmv" Height="100" Width="100"/>
```

Videon päälle voidaan kirjoittaa tekstiä, muokkaamalla Z järjestystä, eli syvyysastetta tai kirjoittamalla XAML:ään, MediaElement:n jälkeen, mitä haluaa videon päällä näkyvän. Tähän tyyliin voidaan luoda videon toistoon tarvittavat napit, eli Play, Stop ja Pause. Tähän voi käyttää niitä kontrolleja mitä parhaakseen näkee. Napeilla on Click tapahtuma, joka käynnistyy kun painetaan hiiren vasemmalla painikkeella ja tarpeellinen koodi tehdään C#:lla tai Vi-

sual Basic:lla. Jos käyttää tekstikontrolleja napin sijaan, käytetään tapahtumaa `MouseDown`, eli hiiren vasen painike on alhaalla. (Moroney 2008, 255 – 247.)

`MediaElement:n` `CurrentState` (nykyinen tila) arvoon kuuluu `Buffering`, `Closed`, `Error`, `Opening`, `Pause`, `Playing` ja `Stopped`. `Buffering` tarkoittaa että video lataa vielä ja video on taukotilassa. `Closed` tarkoittaa että video on suljettu. `Error` tulee, kun latauksessa tai videon toistamisessa on ilmennyt ongelma. `Opening` tarkoittaa, että video on löytynyt ja lataus on alka-
massa. `Paused`, `Playing` ja `Stopped` tarkoittavat, että video on taukotilassa, toistossa tai pysäytettyä. (Moroney 2008, 260.)

2.4 Microsoft Expression Studio

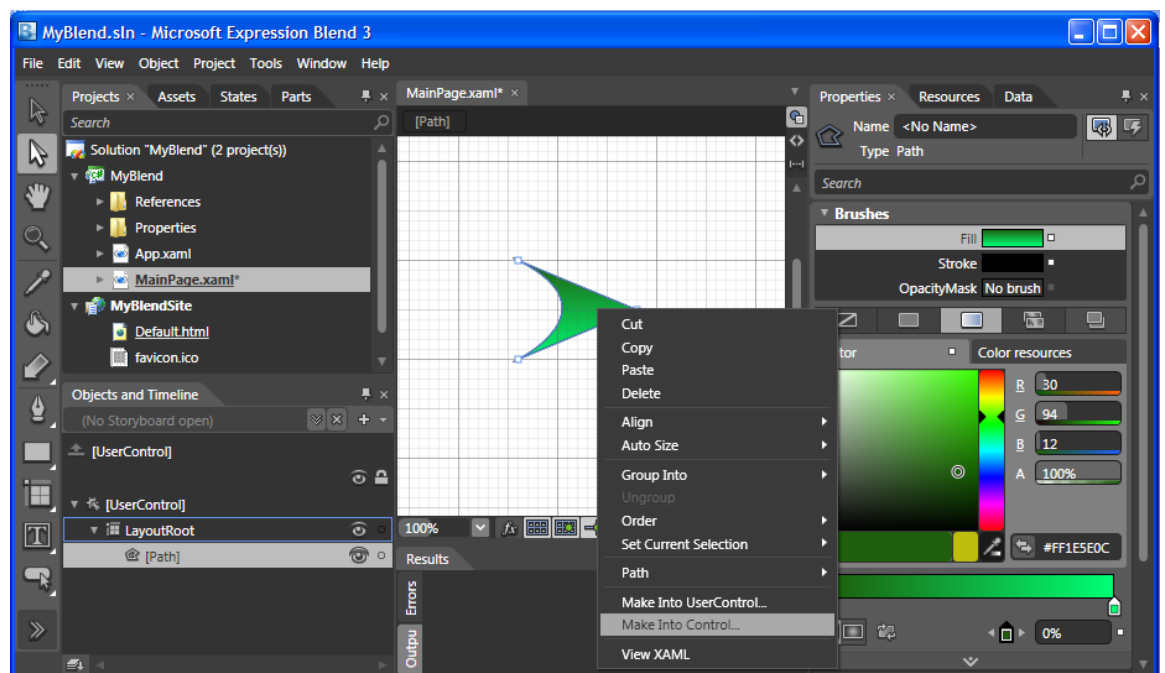
Microsoft Expression Studion ohjelmat on Visual Studion kaltaisia ohjelmankehitysympäristöjä, joilla luodaan web- ja Windows-asiakasohjelmasovelluksia sekä multimediasisältöä. Expression Studio:sta löytyvät ohjelmat `Blend`, `Web`, `Design` ja `Encoder – VC-1`. `Blend` on tarkoitettu Silverlight sovelluksien rakentamiseen graafisella kehityspöydällä. `Web` on web-sivujen kehitystä varten ja se on suunniteltu korvaamaan Microsoft FrontPage. `Design` on web ja XAML grafiikkojen luomiseen ja muokkaamiseen tarkoitettu työkalu. `Encoderin` avulla digitaaliset videot saadaan Silverlight web-videoiksi. Expression Studioon kuului myös Expression Media - Digital Design työkalu, jolla voidaan luoda kuvasto/luettelo kuville ja videoille, mutta Expression Media ei enää kuulu Expression Studio 3 ohjelmistoihin. Expression:n työkalut pystyvät yhteistyöhön Visual Studion kanssa, jossa tarpeellinen ohjelmointi voidaan tehdä Visual Basic- tai C#-ohjelmointikielellä.(xprblog 2007, Microsoft Expression 2009.)

2.5 Microsoft Expression Blend

Expression Blend on tarkoitettu tukemaan Silverlight:a. Blend:ssä on kaksi mahdollisuutta projektille, Silverlight 1 Site ja Silverlight 2 Application. Silverlight 1 Site-projekti luo Silverlight JavaScript projektin, joka sisältää oletus XAML dokumentin. Tässä projektissa on HTML sivu, johon on lisätty tarpeelliset Silverlight objektit. Silverlight 2 Application on niitä

projekteja varten, jossa halutaan käyttää .NET:ää. Projekti sisältää kaiken tarpeellisen, jotta sitä voidaan ohjelmoida .NET Framework:lla. (Moroney 2008, 14.)

Blend:llä voidaan luoda nopeasti ja tarkasti sivun ulkonäkö. Kaikki sisältö voidaan lisätä ja asetella hiiren avulla Drag-and-Drop-menetelmällä. Tarpeelliset koodaukset voidaan tehdä joko Blend:ssä tai Visual Studiolla. Blend:llä on mahdollista rakentaa omia räätälöityjä kontroleja esim. piirtäen Path muodon ja tehden siitä nappi kontrollin (kuvio 1.). Tarkemmat grafiikat tehdään Expression Designer:lla ja videot Expression Encoder:lla, jotka otetaan käyttöön Blend:n avulla.



Kuvio 1. Blend:n kehitysympäristö, jossa ollaan luomassa nappikontrollia.

2.6 Microsoft Expression Web

Expression Web on suunnattu web-kehityksen ja -suunnittelun ammattilaisille. Expression Web tarjoaa web-kehitystyökalun, joka sisältää HTML yhteensopivuuden, WYSIWYG ja tekstieditori tuen, CSS:än ja useiden sivupohjien tuen. Expression Web tukee ASP.NET:ää ja ASP.NET Ajax:a, joten se sopii hyvin työkaluksi rakentaessa sivuja ASP.NET:llä. (Kyrnin, 2009.)

Huomattavimmat ominaisuudet Expression Webissä ovat:

- **Koodieditori**, joka sisältää IntelliSense ja värikoodituen HTML, CSS, JavaScript ja PHP:lle. Mahdollisista virheistä tai puutteista koodissa ilmoitetaan reaaliajassa.
- **Graafinen suunnittelupinta**, mahdollistaa mm. koon ja sijainnin muokkaamisen hiirellä.
- **Gaafisen diagnoosin** avulla kehittäjä näkee muutokset eri selaimilla reaaliajassa, ilman että ohjelma pitää käynnistää aina erikseen.
- **Laaja tuki ohjelmointi ja palvelin teknologioille**, joita ovat Microsoft Visual Studio ja Expression Studio lisäksi PHP, HTML/XHTML, XML/XSLT, CSS, JavaScript, ASP.NET, ASP.NET Ajax, Silverlight, Flash, Windows Media ja Photoshop.
- **Interaktiiviset sivut**. Sivulle voidaan lisätä Silverlight, Flash, Deep Zoom ja Windows mediatiedostoja.
- **Omat grafiikat**. Expression Design:n avulla voidaan luoda ja muokata graafista sisältöä.
- **Silverlight videoiden suoratoisto**. Expression Web sisältää kopion Expression Encoderista, jonka avulla voidaan toistaa WMV, VC-1 ja H.264 videoita.
- **Joustava julkaiseminen**. Julkaisun mukana Expression Web välittää useita tiedostoja kerralla ja nopeasti. Expression Web:llä voi hallita useita sivustoja. (Microsoft Expression, 2009.)

2.7 Microsoft Expression Design

Expression Design:lla voidaan tehdä graafisia malleja sovellukseen käytettäväksi. Sillä voidaan yhdistää vektori-pohjaisia ja bitmap kuvia. Expression Design tukee monien graafisten tiedostojen lukua, kuten:

- Adobe Illustrator – PDF Compatible (*.ai)
- Adobe Photoshop (*.psd)

- Graphical Interchange Format (.gif)
- Portable Network Graphics format (.png)
- Bitmaps (.bmp, .dib, .rle)
- JPEG formats (.jpeg, .jpg, .jpe, .jfif, .exif)
- Windows Media Photos (.wdp, .hdp)
- Tagged Image File Format (.tiff, .tif)
- Icons (-ico) (Moroney 2008, 11 – 12)

Expression Design kirjoittaa seuraavia:

- XAML Silverlight Canvas
- XAML WPF Resource Dictionary
- XAML WPF Canvas
- Adobe Illustrator (.ai)
- Portable Document Format (.pdf)
- Adobe Photoshop (.psd)
- Tagged Image File Format (.tiff, .tif)
- JPEG formats (.jpeg, .jpg)
- Windows Bitmap (.bmp)
- Portable Graphics Network format (.png)
- Graphical Interface Format (.gif)
- Windows Media Photos (.wdp) (Moroney 2008, 12.)

2.8 Microsoft Expression Encoder – VC-1

Expression Encoder käyttää Silverlight:a videoiden koodaamiseen, muokkaamiseen ja julkaisemiseen. Kaikki tiedostot, joille Directshow filter on saatavilla ja se on koneelle asennettu, pystytään importoimaan Encoderin avulla. Mediatiedostot muutetaan WMV:ksi tai WMA:ksi käyttäen profiilia, joka on optimoitu tiedoston välittäväälle asiakasohjelmalle. Encoderilla voi myös luoda metadataa videoihin. Metadataa ovat esim. Internet videoiden päällä olevat pienet laatikot jotka, sisältävät tekstiä. (Moroney 2008, 19.)

3 JAPANIN KIELI

Japanin kielen opetteleminen aloitetaan Japanissa jo lastentarhassa, jossa heille opetetaan oman nimen kirjoittaminen, käyttäen japanin kielen hiragana merkkejä. Japanin kieli koostuu hiraganoista, katakanoista ja kiinalaisista merkeistä eli kanjeista. Hiraganoilla kirjoitetaan suurin osa japaninkielisistä sanoista ja ne ovatkin ensimmäinen opeteltava asia japanin kielestä. Katakanoilla kirjoitetaan vierasperäiset nimet ja sanat sekä tietyt japanilaisperäiset sanat, kuten jotkin eläin- ja kasvilajit sekä ääntä tai tapaa kuvaavat adverbit. Kanjeja käytetään ilmaimaan kaikki käsitteelliset asiat. Japanin hallitus on määritellyt 1945 yleisessä käytössä olevaa kanji merkkiä, jotka jokaisen japanilaisen tulee osata. Näistä merkeistä 1006 opetellaan alasteella ja 939 yläasteella ja lukiossa. Numerot kirjoitetaan kanjeilla tai arabialaisilla numeroilla. Arabialaiset numerot ovat huomattavasti yleistyneet ja vain suuret luvut, kuten esim. 千万 (10 000 000) tai 京 (10 000 000 000 000 000), kirjoitetaan kanjeilla.

Japanissa latinalaista aakkostoa kutsutaan nimellä romaji. Perinteisesti japania kirjoitetaan ylhäältä alas ja oikealta vasemmalle. Nykyään kirjoitetaan paljon vaakasuorassa vasemmalta oikealle, mutta sivuja silti liikutaan eteenpäin oikealta vasemmalle. Lehdet ja kirjat käyttävät enimmäkseen ylhäältä alas lukusuuntaa.

Suomalaiselle japanin ääntäminen on yleensä hyvin helppoa. Molemmilla on kuitenkin muutamia äännteitä, joita toisella ei ole. Japanissa ei ole L- ja R-äännteitä, vaan pikemminkin niiden välimuoto, joka saattaa kuulostaa Suomalaisen korvaan ärrävikaiselta. L-kirjaimen sisältävät nimet korvataan yleensä japanin R-äännteellä. V-äänne korvataan japanin kielessä B-äännteellä.

Japanin kielen kääntäminen tapahtuu käyttämällä jotain tiettyä järjestelmää. Hepburn-järjestelmä on yleisin ja sen on kehittänyt englantia äidinkielenään puhuva amerikkalainen, joten se perustuu englannin kielen ääntämiseen. Suomella ei ole omaa järjestelmää, vaan usein käytetään Hepburn-järjestelmän variaatiota.

Äidinkielenään japania puhuu noin 127 miljoona ihmistä. Tähän lasketaan okinawan kieli ja Ryukyu-saarilla puhutut kielet. Monet tutkijat pitävät japanilaisia kieliä toistensa murteina ja siten japanista on tullut maailman suurin isolaattikieli. Muoto-opiltaan japanin kieli on sa-

mankaltainen kuin suomen kieli, siinä sanan vartaloon liittyy päätteitä ja muita affikseja. (Karppinen, 7 – 9, Silja 2009.)

Japani kieli näyttää esim. tältä:

Japaniksi	Aakkosissa (romaji)	Suomeksi
わたしはサンナです	wa ta shi wa sa n na de su	Minä olen Sanna

Kaikki muu kirjoitettiin hiraganoilla, paitsi ei vierasperäistä nimeä, joka kirjoitettiin katakanal-la. Ääntäminen on hyvin pitkälle samankaltaista kuin suomen kieli, sillä useimmat sanat äännetään samoin kuin luetaan.

watashi	minä	anata	sinä
inu	koira	neko	kissa
hon	kirja	pen	kynä

Olemassa on monia sanoja ja päätteitä, jotka lausutaan erilailla kuin ne kirjoitetaan. Näiden tunteminen vaatii opettelua.

<i>Sana</i>	<i>Suomennos</i>	<i>Muutos</i>	<i>Lausutaan</i>
suki	pitää, tykätä	U:ta ei lausuta	ski
aishiteru	rakastan sinua	I:tä ei lausuta	aishteru
imasu	olla olemassa	U.ta ei lausuta	imas
arimasu	olla olemassa	U.ta ei lausuta	arimas
desu	olla	U.ta ei lausuta (paitsi joissain murteissa)	des

Taulukossa 2. on lueteltuna kaikki hiraganat ja niiden romaji käänнос ja taulukossa 3. on katanakanat.

あ	A	い	I	う	U	え	E	お	O
か	KA	き	KI	く	KU	け	KE	こ	KO
さ	SA	し	SHI	す	SU	せ	SE	そ	SO
た	TA	ち	CHI	つ	TSU	て	TE	と	TO
な	NA	に	NI	ぬ	NU	ね	NE	の	NO
は	HA	ひ	HI	ふ	FU	へ	HE	ほ	HO
ま	MA	み	MI	む	MU	め	ME	も	MO
や	YA			ゆ	YU			よ	YO
ら	RA	り	RI	る	RU	れ	RE	ろ	RO
わ	WA							を	O
ん	N								
が	GA	ぎ	GI	ぐ	GU	げ	GE	ご	GO
ざ	ZA	じ	JI	ず	ZU	ぜ	ZE	ぞ	ZO
だ	DA	ぢ	JI	づ	ZU	で	DE	ど	DO
ば	BA	び	BI	ぶ	BU	べ	BE	ぼ	BO
ぱ	PA	ぴ	PI	ぷ	PU	ぺ	PE	ぽ	PO
きゃ	KYA			きゅ	KYU			きょ	KYO
しゃ	SHA			しゅ	SHU			しょ	SHO
ちゃ	CHA			ちゅ	CHU			ちょ	CHO
にゃ	NYA			にゅ	NYU			にょ	NYO
ひゃ	HYA			ひゅ	HYU			ひょ	HYO
みゃ	MYA			みゅ	MYU			みょ	MYO
りゃ	RYA			りゅ	RYU			りょ	RYO
ぎゃ	GYA			ぎゅ	GYU			ぎょ	GYO

Taulukko 2. Hiraganat ja niiden käänös latinalaisiin aakkosiin.

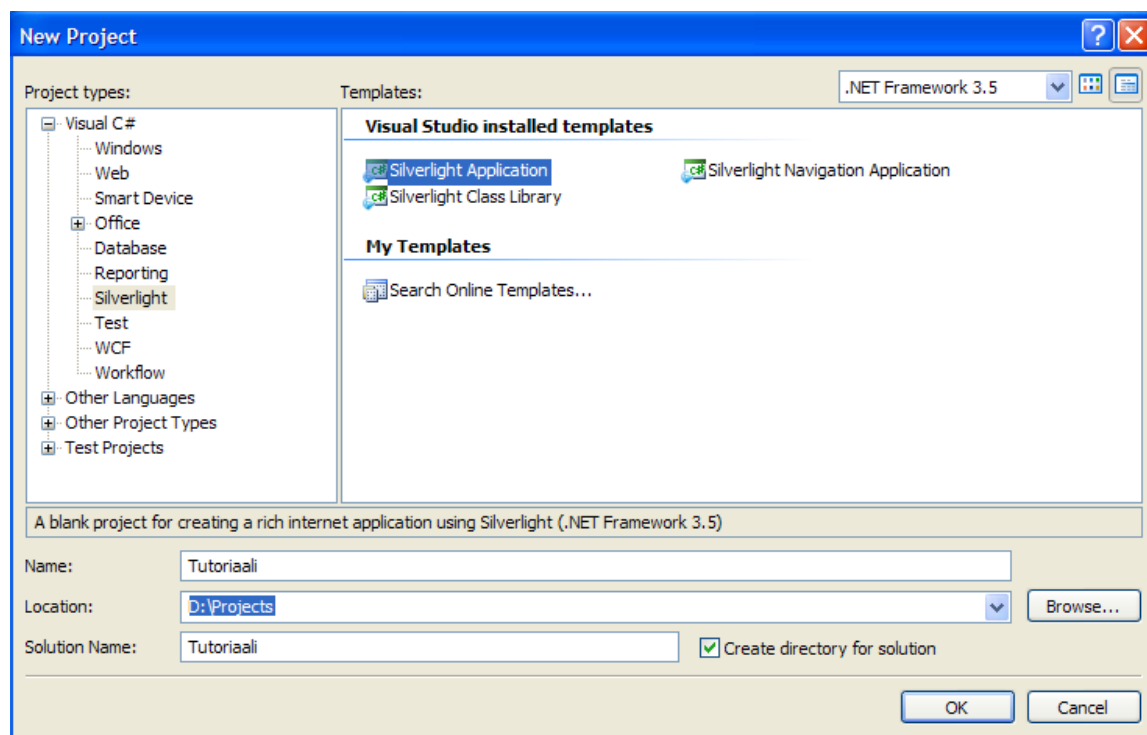
ア	A	イ	I	ウ	U	エ	E	オ	O
カ	KA	キ	KI	ク	KU	ケ	KE	コ	KO
サ	SA	シ	SHI	ス	SU	セ	SE	ソ	SO
タ	TA	チ	CHI	ツ	TSU	テ	TE	ト	TO
ナ	NA	ニ	NI	ヌ	NU	ネ	NE	ノ	NO
ハ	HA	ヒ	HI	フ	FU	ヘ	HE	ホ	HO
マ	MA	ミ	MI	ム	MU	メ	ME	モ	MO
ヤ	YA			ユ	YU			ヨ	YO
ラ	RA	リ	RI	ル	RU	レ	RE	ロ	RO
ワ	WA							ヲ	O
ン	N								
ガ	GA	ギ	GI	グ	GU	ゲ	GE	ゴ	GO
ザ	ZA	ジ	JI	ズ	ZU	ゼ	ZE	ゾ	ZO
ダ	DA	ヂ	Ji	ヅ	ZU	デ	DE	ド	DO
バ	BA	ビ	BI	ブ	BU	ベ	BE	ボ	BO
パ	PA	ピ	PI	プ	PU	ペ	PE	ポ	PO
キャ	KYA			キュ	KYU			キョ	KYO
シャ	SHA			シュ	SHU			ショ	SHO
チャ	CHA			チュ	CHU			チョ	CHO
ニャ	NYA			ニュ	NYU			ニョ	NYO
ヒャ	HYA			ヒュ	HYU			ヒョ	HYO
ミャ	MYA			ミュ	MYU			ミョ	MYO
リャ	RYA			リュ	RYU			リョ	RYO
ギャ	GYA			ギュ	GYU			ギョ	GYO
ジャ	JA			ジュ	JU			ジョ	JO
チャ	JA			チュ	JU			チョ	JO
ビャ	BYA			ビュ	BYU			ビョ	BYO
ピャ	PYA			ピュ	PYU			ピョ	PYO

Taulukko 3. Katakanaat ja niiden käännös latinalaisiin aakkosiin.

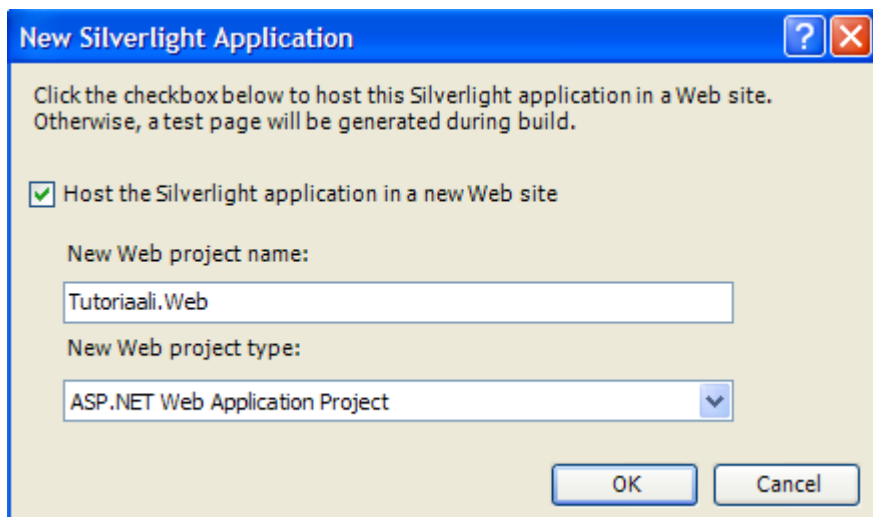
4 TUTORIAALI

Tämän tutoriaalin tarkoitus on opettaa Silverlight:n käyttöä ja sen toteutus on ohjelmointityön kaltainen. Tutoriaalissa tehdään sivusto, jolla on useita sivuja ja linkit niille. Sivun yläosalla on linkki etusivulle ja sivun vasemmassa laidassa on linkit sivulla suunnistamista varten. Jäljelle jääneeseen osaan tulee näkyviin valittu sivu. Käytettävät kontrollit ja ohjelmointi on samaa, kuin mitä ohjelmointityössä on käytetty. Tutoriaalissa on käytetty Visual Studio 2008 ohjelmankehitysympäristöä ja ohjelmointikielenä on C#.

Aloita luomalla uusi Silverlight projekti ja anna sille nimeksi **Tutoriaali** (kuvio 2.). Valitse ASP.NET Web Application Project ja paina OK (kuvio 3.).



Kuvio 2. Luodaan uutta Silverlight projektia ja annetaan sille nimi.



Kuvio 3. Valitaan projektityyppiä.

MainPage.xaml:ssä poista **DesignWidth** ja **DesignHeight** (kuvio 4.), jolloin sivun koko on näytön kokoinen.



Kuvio 4. MainPage.xaml:n oletussisältö.

Kuten kuviossa 4. näkyy, **x:Name** määrittelee nimen kontrollille. **LayoutRoot** tulee aina Grid elementin nimeksi, kun uusi User Control luodaan. Seuraavat kohdat kirjoitetaan **<Grid> </Grid>** merkintöjen väliin (kuvio 4.).

```
<Grid.RowDefinitions>
    <RowDefinition Height="1*" />
    <RowDefinition Height="6*" />
</Grid.RowDefinitions>

<Grid.ColumnDefinitions>
    <ColumnDefinition Width="100" />
    <ColumnDefinition />
</Grid.ColumnDefinitions>
```

Grid.RowDefinition kohtaan määritellään montako pystyriviä Grid, eli tässä tapauksessa **LayoutRoot**, sisältää ja mahdollisesti niiden koko. **ColumnDefinition** kohtaan määritellään vaakarivien määrä ja mahdollinen koko. Row kohdassa **Height = "1*"** tarkoittaa että, ensimmäinen rivi on automaattisesti 1/7 näkyvän sivun korkeudesta ja toinen rivi on 6/7 osaa. Column kohdassa vasemman laidan leveys on 100 pistettä ja loppuosa kuuluu oikeanpuoliselle.

Seuraavaksi luodaan Canvas kontrolli, joka asetetaan sivuston vasempaan ylänurkkaan. Kirjoita **ColumnDefinition**:n jälkeen:

```
<Canvas x:Name="Etusivu"
        Grid.Column="0"
        Grid.Row="0"
        Background="Aqua">

</Canvas>
```

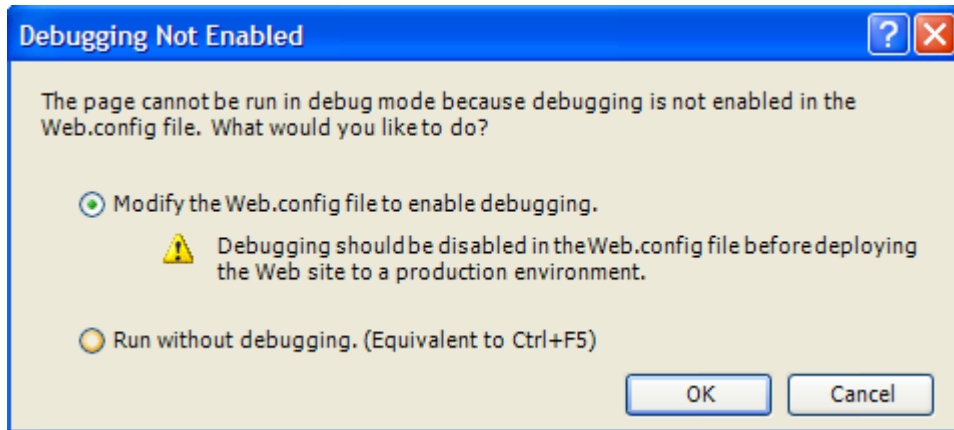
Grid.Column määrittelee kohdan vaakatasossa ja **Grid.Row** pystytasossa. **Background** antaa alueelle tietyn värin. Lisää Canvas kontrollille teksti:

```
<Canvas x:Name="Etusivu"
        Grid.Column="0"
        Grid.Row="0"
        Background="Aqua">

        <TextBlock Text="Etusivulle"
                   FontSize="16"
                   FontFamily="Arial"
                   FontWeight="Bold"/>

</Canvas>
```

Text kohtaan kirjoitetaan teksti, joka halutaan näkyvän. **FontSize** on fontin koko, **FontFamily** on mitä fonttia käytetään ja **FontWeight** on fontin paksuus. Suorita Build (Build -> Build Solution) projektille ja käynnistä sen jälkeen ohjelma (Debug -> Start Debugging). Tämän jälkeen tulee kuviossa 5. näkyvä ilmoitus. Paina OK.



Kuvio 5. Valitaan tehdäänkö sovellukselle debug toiminto, aina kun ohjelma käynnistetään.

Tässä vaiheessa pitää tulla näkyviin valkoinen sivu, jossa on vaaleansininen laatikko vasemmassa ylälaidassa, jonka sisällä on teksti ”**Etusivulle**”. Sulje sivu.

Kirjoita seuraavaksi äskeisen Canvas kohdan jälkeen:

```
<StackPanel Orientation="Vertical"
            Grid.Column="0"
            Grid.Row="1">
</StackPanel>
```

Stack Panel asettelee sisällön automaattisesti allekkain. Tämän sisälle lisätään kaksi nappia seuraavasti:

```
<StackPanel Orientation="Vertical"
            Grid.Column="0"
            Grid.Row="1">
    <Button x:Name="btnEtusivu"
            Content="Etusivu"
            Height="30"/>
    <Button x:Name="btnToinenSivu"
            Content="Toinen SivU"
            Height="30"/>
</StackPanel>
```

Content on napin teksti ja Height määrittelee kontrollin korkeuden. Kirjoita seuraavaksi Stack Panel:n jälkeen:

```
<ScrollViewer x:Name="scrlView"
            Grid.Column="1"
            Grid.Row="1"
            Background="Aquamarine"
```

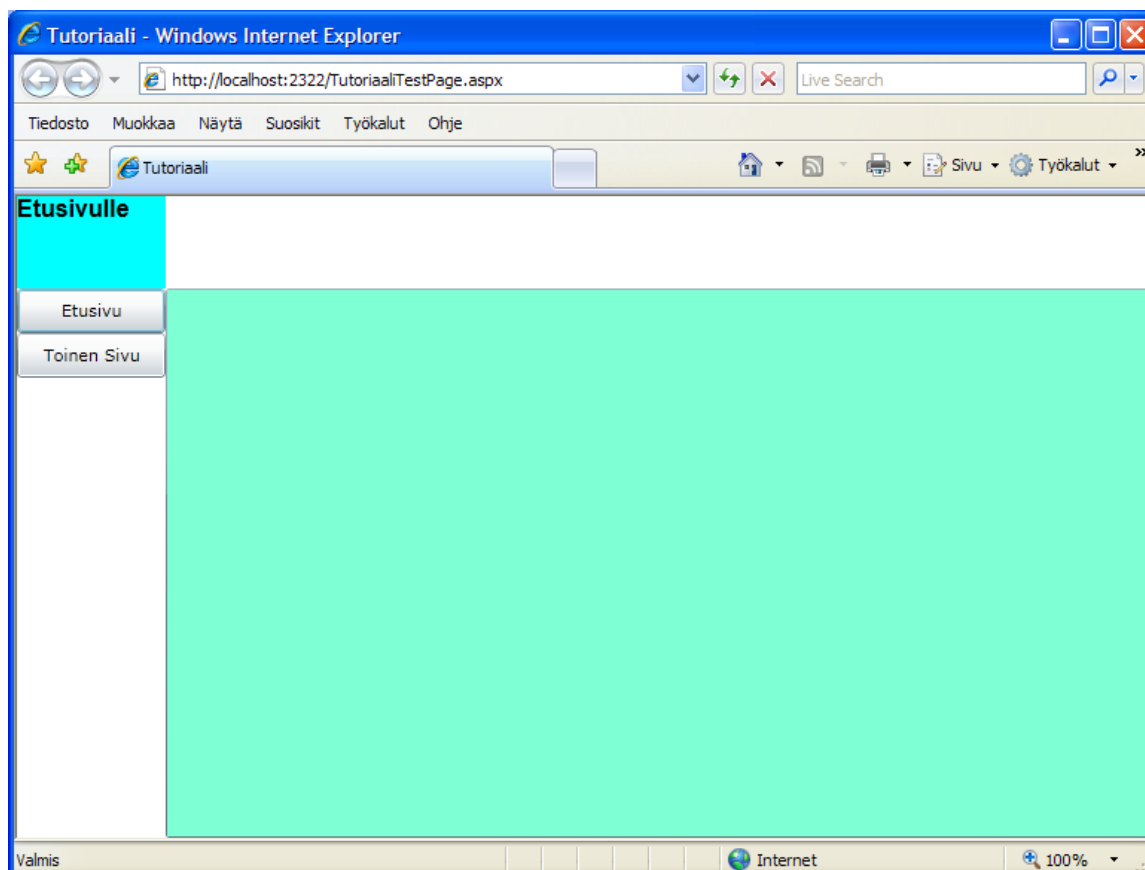
```

VerticalScrollBarVisibility="Auto"
HorizontalScrollBarVisibility="Auto">

</ScrollViewer>

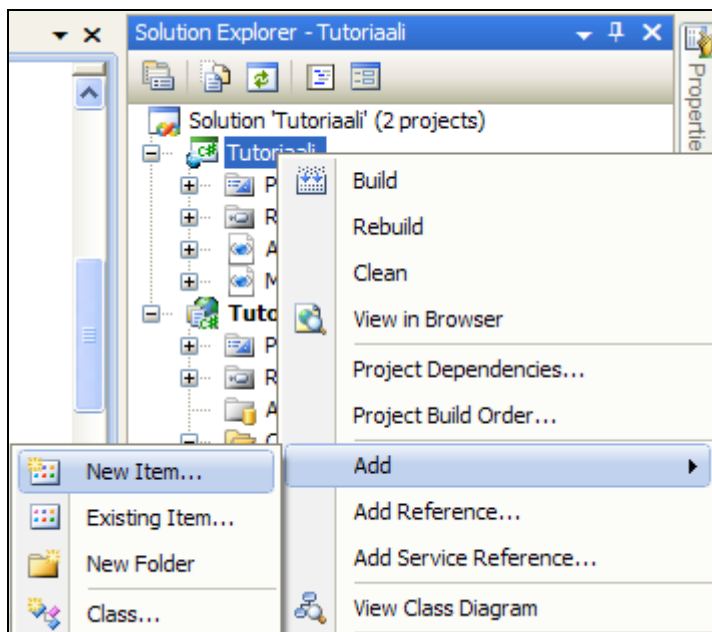
```

Scroll Viewer sisältää vierityspalkin. Tässä vierityspalkkien näkyvyys on asetettu automaattiseksi eli ne eivät näy, jos sisältö mahtuu sivulle. Kokeile jälleen miltä työ näyttää. Sen tulee näyttää tältä (kuvio 6.).

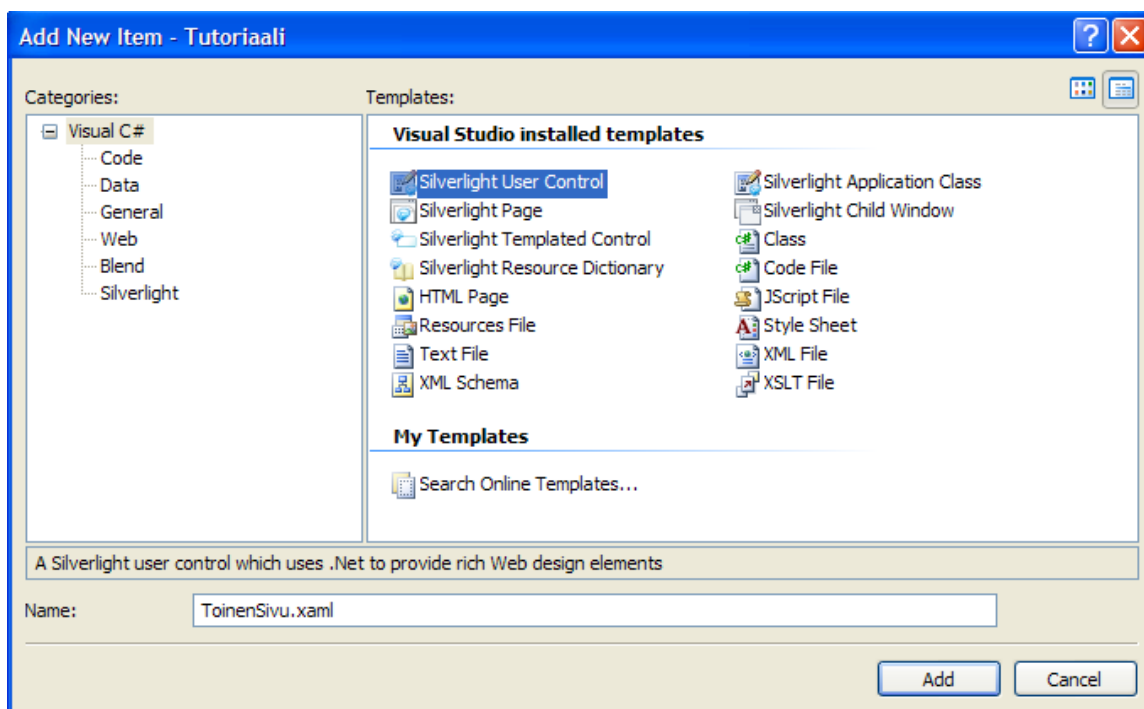


Kuvio 6. Sivuston ulkonäkö.

Seuraavaksi luodaan uusi sivu. Klikkaa hiiren oikealla painikkeella **Tutoriaali:** ja valitse **New Item** (kuvio 7.). Valitse **Silverlight User Control** ja anna sille nimeksi **ToinenSivu.xaml** (kuvio 8.).



Kuvio 7. Lisätään uusi osio projektiin.



Kuvio 8. Valitaan uudeksi osioksi User Control ja annetaan sille nimi.

ToinenSivu.xaml sisältö näyttää oletuksena tältä:

```
<UserControl x:Class="Tutoriaali.ToinenSivu"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="400" Height="300">
```

```

    <Grid x:Name="LayoutRoot" Background="White">

    </Grid>
</UserControl>

```

Poista sivun koko rajoitus (**Width** ja **Height**) ja muuta **LayoutRoot**:n tausta mustaksi (**Black**). Lisää yksi tekstikenttä seuraavasti:

```

<UserControl x:Class="Tutoriaali.ToinenSivu"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Grid x:Name="LayoutRoot" Background="Black">

        <TextBlock FontSize="100"
            Foreground="White">
            <TextBlock.Text>
Terve!
Tämä
on
toinen
sivu!
            </TextBlock.Text>
        </TextBlock>

    </Grid>
</UserControl>

```

Foreground antaa tekstille värin. **TextBlock.Text** avulla voi kirjoittaa selkeämmän näköisesti enemmän tekstiä, kuin kirjoittamalla **Text=""** ” ” lainausmerkkien sisään. Tämän sivun tarkoitus on näyttää, kuinka uusi sivu saadaan näkyviin ja kuinka Scroll Viewer toimii.

Palaa **MainPage.xaml**:ään. Mene napin **btnToinenSivu** luo, kirjoita sille **Click** tapahtuma ja valitse **New Event Handler** (kuvio 9.). Klikkaa oikealla hiiren painikkeella **Click**:n päällä ja valitse **Navigate to Event Handler** (kuvio 10.).

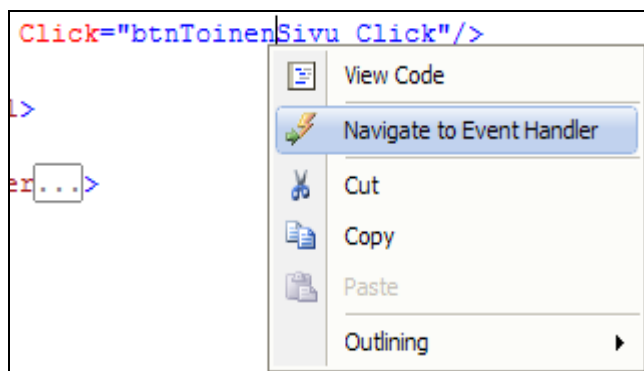


```

<StackPanel Orientation="Vertical"
    Grid.Column="0"
    Grid.Row="1">
    <Button x:Name="btnEtusivu"
        Content="Etusivu"
        Height="30"/>
    <Button x:Name="btnToinenSivu"
        Content="Toinen SivU"
        Height="30"
        Click=""/>
</StackPanel>

```

Kuvio 9. Kirjoitetaan Click tapahtuma.



Kuvio 10. Suunnistetaan napin Click koodiin.

Tarkista että **MainPage.xaml.cs** on auki. Luodaan uusi **ToinenSivu** User Control, joka laitetaan Scroll Viewer:n sisään. **MainPage.xaml.cs**:ssä kirjoita napin koodiin seuraavasti:

```
private void btnToinenSivu_Click(object sender, RoutedEventArgs e)
{
    scr1View.Content = new ToinenSivu();
}
```

Palaa **MainPage.xaml**:ään ja tee **btnEtusivu**:lle **Click** tapahtuma samaan tyyliin. Kirjoita sen koodiin seuraavasti:

```
private void btnEtusivu_Click(object sender, RoutedEventArgs e)
{
    scr1View.Content = null;
}
```

Etusivulla Scroll Viewer on siis tyhjä ja **btnToinenSivu** nappia painamalla Scroll Viewer:lle annetaan sisällöksi **ToinenSivu** User Control. Tämä on vain yksi tapa, jolla sovellukseen voi sisällyttää useampia sivuja. Kyseistä tapaa on käytetty tämän opinnäytetyön ohjelmointityössä.

Seuraavaksi lisätään tapahtuma, jonka avulla vasemmassa ylänurkassa olevaa Canvas-pohjaa painamalla saadaan etusivu näkyviin. Lisätään 3 tapahtumaa, joista 2 annetaan Canvas **Etusivu**:lle (kuvio 11.) ja 1 Grid **LayoutRoot**:lle (kuvio 12.). Tapahtumaa kirjoittaessa, valitse aina **New Event Handler**.

```
<Canvas x:Name="Etusivu"
    Grid.Column="0"
    Grid.Row="0"
    Background="Aqua"
    MouseLeftButtonDown="Etusivu_MouseLeftButtonDown"
    MouseLeftButtonUp="Etusivu_MouseLeftButtonUp">
```

Kuvio 11. Lisätään tapahtumat hiiren vasen painike alas ja ylös.

```
<Grid x:Name="LayoutRoot" MouseLeftButtonUp="LayoutRoot_MouseLeftButtonUp">
```

Kuvio 12. Lisätään LayoutRoot:lle tapahtuma hiiren vasen painike ylös.

Tee seuraavaksi kuvion 13. ympyröidyt lisäykset **MainPage.xaml.cs**:ään.

```
public partial class MainPage : UserControl
{
    private bool _bOk = false;

    public MainPage() {...}

    private void btnToinenSivu_Click(object sender, RoutedEventArgs e) {...}

    private void btnEtusivu_Click(object sender, RoutedEventArgs e) {...}

    private void LayoutRoot_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
    {
        _bOk = false;
    }

    private void Etusivu_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)
    {
        _bOk = true;
    }

    private void Etusivu_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
    {
        if (!_bOk)
        {
            scrollView.Content = null;
        }
    }
}
```

Kuvio 13. Vaaditut lisäykset MainPage.xaml.cs:ään

Tämä tarkistaa, että hiiren vasen painike on ollut pohjassa ja vapautettu Canvas **Etusivu** päällä, jolloin etusivu tulee näkyviin. Jos vapauttaa hiiren muualla tai sattuu painamaan hiirtä

toisessa paikassa ja vapauttaa sen jälkeen hiiren Canvas **Etusivu**-pohjan päällä, ei tapahdu mitään. Jotta tämä toimii täysin oikein, lisää **LayoutRoot**:lle väri.

```
<Grid x:Name="LayoutRoot"
      MouseLeftButtonUp="LayoutRoot_MouseLeftButtonUp" Background="White">
```

Kuviossa 13. näkyvä **LayoutRoot_MouseLeftButtonUp** ei käynnisty kun hiiri vapautetaan **LayoutRoot**:n ”tyhjissä” kohdissa tämän vuoksi sille annetaan väri. Kaikki pohjat näyttävät oletukseltaan valkoisilta, mutta todellisuudessa ne ovat tyhjiä.

Lisää tapahtuma Scroll Viewer:n vierityspalkille, jotta sitä voidaan hallita hiiren vierityspainikkeella. Samoin kuin aiemmat tapahtumat, lisää Scroll Viewer:lle **MouseWheel** tapahtuma ja kirjoita sen koodiin seuraavasti.

```
<ScrollViewer x:Name="scrlView"
              Grid.Column="1"
              Grid.Row="1"
              Background="Aquamarine"
              VerticalScrollBarVisibility="Auto"
              HorizontalScrollBarVisibility="Auto"
              MouseWheel="scrlView_MouseWheel">

</ScrollViewer>
```

```
private void scrlView_MouseWheel(object sender, MouseWheelEventArgs e)
{
    if (e.Delta < 0)
        scrlView.ScrollToVerticalOffset(scrlView.VerticalOffset + 30);
    else
        scrlView.ScrollToVerticalOffset(scrlView.VerticalOffset - 30);
}
```

Tarkista vielä näyttääkö koodi oikealta liitteen 1. avulla. Valmiin sivuston ulkonäön voit tarkistaa liitteen 2. avulla. Tässä tutoriaalissa opetetut asiat ovat käytössä tämän opinnäytetyön ohjelmointityössä, joskin suuremmassa mittakaavassa. Tutoriaalın sivusto on kömpelön näköinen, mutta antaa selkeän käsityksen kuinka Silverlight toimii.

Lisäharjoituksena luo kolmas sivu aiemman ohjeen mukaisesti (kuvio 7.). Anna sille nimeksi **KolmasSivu.xaml**. **KolmasSivu.xaml**:ssä poista jälleen **Width** ja **Height** ja anna **LayoutRoot**:lle jokin väri, esimerkiksi **RosyBrown**. Lisää **LayoutRoot**:n sisään:

```
<Grid.ColumnDefinitions>
    <ColumnDefinition Width="100"/>
    <ColumnDefinition Width="100"/>
    <ColumnDefinition/>
</Grid.ColumnDefinitions>
```

```

<StackPanel Orientation="Vertical"
    Grid.Column="0">
    <RadioButton GroupName="group1"
        Content="Ryhmä 1: A"/>
    <RadioButton GroupName="group1"
        Content="Ryhmä 1: B"/>
    <RadioButton GroupName="group1"
        Content="Ryhmä 1: C"/>
</StackPanel>

<StackPanel Orientation="Vertical"
    Grid.Column="1">
    <RadioButton GroupName="group2"
        Content="Ryhmä 2: A"/>
    <RadioButton GroupName="group2"
        Content="Ryhmä 2: B"/>
    <RadioButton GroupName="group2"
        Content="Ryhmä 2: C"/>
</StackPanel>

```

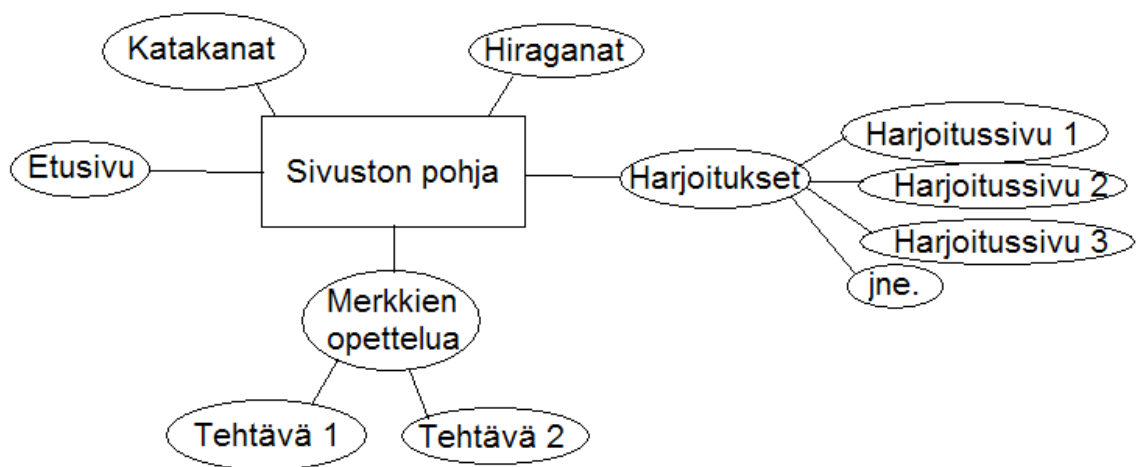
Radio button on niitä tilanteita varten, joissa käyttäjä voi valita vain yhden vaihtoehdon monen joukosta. Radio button:lle pitää antaa **GroupName** arvo, joka määrittelee mihin ryhmään se kuuluu. Viimeiseksi luo nappi, jolla voi suunnistaa **KolmasSivu**:lle. Mene **MainPage.xaml**:ään ja lisää uusi nappi, **btnToinenSivu** jälkeen. Lisää sille **Click** tapahtuma ja suunnista sen koodiin. Kirjoita napin koodiin:

```
scr1View.Content = new KolmasSivu();
```

Liitteessä 3. on **MainPage.xaml** ja **MainPage.xaml.cs** dokumentteihin lisätty lisäharjoituksen koodi ja uuden sivun kuva.

5 OHJELMOINTITYÖ JA SEN TOTEUTUS

Ohjelmointityönä on tehty Silverlight web-sovellus, johon on myös käytetty Expression Blend ohjelmankehitysympäristöä. Itse sovelluksen tarkoitus on esitellä japanin kielen alkeita. Sovelluksella ei opeteta alkeita syvällisesti, vaan pikemminkin käyttäjä voi tutustua sillä japanin kieleen ja samalla saada käsityksen millaista sen opiskelu on. Sovellus keskittyy hiragana ja katakana merkkeihin, joiden opetteleminen on mahdollista sovelluksen avulla. Kuviossa 14. on kaavio sivuston toiminnasta.



Kuvio 14. Sivuston toimivuutta kuvaava kaavio.

5.1 GUI eli graafinen käyttöliittymä

Sivusto sisältää viisi pääsivua:

- **Etusivulla** on selitys sivuston tarkoituksesta ja seloste mitä jokainen sivu itsessään sisältää.
- **Harjoitukset**, sisältää nimensä mukaisesti harjoituksia käyttäjälle, joiden tarkoitus on opettaa hieman alkeita ja tutustuttaa käyttäjä kyseiseen kieleen.
- **Hiraganat** ja **Katakanat**, molemmat sivut sisältävät luettelon nimensä mukaisista merkeistä ja niiden merkityksen aakkosissa.

- **Merkkien opettelua**, sisältää kaksi harjoitusta joiden avulla käyttäjä voi opetella hiraganoja ja katakanoja.

Sivuilla suunnistamiseen on käytettyä ratkaisua, jossa aiemmin käydyt sivut eivät jää muistiin. Sivut ovat User Control:ja jotka näkyvät pääsivulla. Sivuja selatessa vanha sivu poistetaan ja uusi luodaan sen tilalle. Tätä tapaa käyttäen muistiin ei jää mitään ylimääräistä. Sivustossa liikkumiseen käytetään samannäköistä pohjaa jokaisella sivulla. Pohja jota käytetään sivujen tutkimiseen sisältää ylätasoa, jonka kautta voidaan palata etusivulle ja loppuosa on jaettu kahteen, jossa haluttu sivu tulee näkyviin oikealle ja vasen laita sisältää suunnistuslinkit kaikille sivuille (kuvio 15.).

ETUSIVULLE

Japanin kirjoitusjärjestelmä koostuu hiraganoista, katakanoista ja kiinalaisista merkeistä eli kanjeista. Hiraganoilla kirjoitetaan useimmat japanin kieliset sanat ja ne opetellaan ensimmäisenä. Katakanoilla kirjoitetaan vierasperäiset nimet ja sanat sekä tietyt japanilaisperäiset sanat, kuten jotkin eläin- ja kasvilajit ja ääntä tai tapaa kuvaavat adverbit. Japanin hallitus on määritellyt 1945 yleisessä käytössä olevaa kanjia, jotka japanilaisen tulisi osata ja näistä 1006 opetellaan ala-asteella ja 939 yläasteella ja lukiossa.

Seuraavat harjoitukset opettavat lyhyesti perusteita ja keskittyvät opettamaan hiraganoja.

Perushiraganat	Dakuon
あ い う え お a i u e o	が き く け こ ga gi gu ge go
か き く け こ ka ki ku ke ko	ぎ じ ず せ ぞ za ji zu ze zo
さ し す せ そ sa shi su se so	だ ち づ で ど da ji zu de do
た ち つ て と ta chi tsu te to	ば ひ ぶ べ ぼ ba bi bu be bo
な に ぬ ね の na ni nu ne no	
は ひ ふ へ ほ ha hi fu he ho	Handakuon
ま み む め も ma mi mu me mo	ぱ ひ ぶ べ ぽ pa pi pu pe po
や ゃ ゅ ょ ya yu yo	
ら り る れ ろ ra ri ru re ro	
わ を wa o	
ん n	

Navigation buttons: Ensimmäinen, Edellinen, Seuraava, Viimeinen

Page list: Sivuu 1, Sivuu 2, Sivuu 3, Sivuu 4, Sivuu 5, Sivuu 6, Sivuu 7, Sivuu 8, Sivuu 9, Sivuu 10, Sivuu 11, Sivuu 12, Sivuu 13, Sivuu 14, Sivuu 15, Sivuu 16, Sivuu 17, Sivuu 18

Kuvio 15. Sovelluksen perusnäkyvä.

Etusivu sisältää selityksien lisäksi tyylivaihtoehdot sivustolle. Kyseiset tyylit vaihtavat mm. taustan ja nappien värejä. Valittu tyyli välitetään kyseiselle sivulle aina kun sivu avataan. Tyylivaihtoehdot on sijoitettu vasempaan ylänurkkaan, neliöiden muodossa, jotka ovat kyseisen tyylin väreisiä (kuvio 16.).

ETUSIVULLE

JAPANIN KIELEEN TUTUSTUMINEN

Etusivu

Harjoitukset

Merkkien opettelua

Hiraganat

Katakanat

Tämän sivuston tarkoitus on tutustuttaa käyttäjälle japanin kielen alkeet. Sivuston harjoituksilla voi oppia lyhyesti perusteita ja saada käsityksen minkälaista japanin kielen opetteleminen olisi. Näiden harjoitusten ja tehtävien avulla, ei valmistuta asiantuntijaksi.

Harjoitukset sivu sisältää usean sivun verran harjoituksia joiden avulla voit tutustua japanin kieleen.

Merkkien opettelussa on kaksi vaihtoehtoa. Ensimmäisessä pitää yrittää valita oikea vastaus kolmesta vaihtoehdosta. Kyselyä voi muuttaa siten, että siinä kysytään hiraganoja, katakanoja tai niiden käännöstä. Toisessa pitää valita oikea hiragana tai katakana vaihtoehto, kaikkien kyseisten merkkien joukosta.

Hiraganat ja Katakanat sivut sisältävät luettelon kyseisistä merkeistä.

Jos sivun merkit eivät näy, se todennäköisesti johtuu siitä, ettei koneelle ole asennettu oikeaa kielitukea.

Kuvio 16. Sovelluksen etusivun näkymä

Harjoitukset sivu (kuvio 15.) muuttuu hieman ulkonäöltään harjoituksesta riippuen. Harjoitukset ovat enimmäkseen luettavaa (17.), mutta myös kääntämisharjoituksia (kuvio 18.). Tehtäviä selataan alla olevilla suunnistusnuolilla tai oikeassa laidassa olevasta sivuluettelosta valitsemalla. Sanastonappi tuo esiin liikutettavan ikkunan, joka sisältää kyseiselle sivulle liittyvän sanaston.

これ	KORE	TÄMÄ	 Sanasto
これは いぬです。 これも いぬです。 これらは いぬです。 これらは ねこではありません。	kore wa inu desu. kore mo inu desu. korena wa inu desu. korena wa neko dewa arimasen.	Tämä on koira. Tämäkin on koira. Nämä ovat koiria. Nämä eivät ole kissoja.	Sivu 1 Sivu 2 Sivu 3 Sivu 4 Sivu 5 Sivu 6 Sivu 7 Sivu 8 Sivu 9 Sivu 10 Sivu 11 Sivu 12 Sivu 13 Sivu 14 Sivu 15 Sivu 16 Sivu 17 Sivu 18
あれ	ARE	TUO	
あれは こるまです。 あれも こるまですか。 Myönteinen vastaus はい, そうです。 あれは こるまです。 Kielteinen vastaus いいえ, こるまではありません。	are wa kuruma desu. are mo kuruma desu ka. hai, soo desu. are wa kuruma desu. iie, kuruma dewa arimasen.	Tuo on auto. Onko tuokin auto? Kyllä,(niin on). Tuo on auto. Ei, tuo ei ole auto.	
それ	SORE	SE	
それは えんぴつですか。 いいえ, ちがいます。	sore wa enpitsu desu ka. iie, chigaimasu.	Onko se lyijykynä? Ei(, ei ole/väärin).	

◀ Edellinen ▶ Seuraava

Ensimmäinen Viimeinen

Kuvio 17. Tekstipohjainen harjoitussivu.

Valitse oikea vaihtoehto			Sivu 1
こんにちは (ko n ni chi wa) <input type="radio"/> Päivää <input type="radio"/> Kiitos <input type="radio"/> Hauska tutustua	どうぞよろしく (do o zo yo ro shi ku) <input type="radio"/> Minä olen <input type="radio"/> Hauska tutustua <input type="radio"/> Saanko esittäytyä	わたしたちは (wa ta shi ta chi) <input type="radio"/> Me <input type="radio"/> Minä <input type="radio"/> Sinä	Sivu 2
はい (ha i) <input type="radio"/> Ei <input type="radio"/> Kyllä <input type="radio"/> Päivää	あなたは (a na ta wa) <input type="radio"/> Sinä olet <input type="radio"/> Minä olen <input type="radio"/> Hän on	あなたも (a na ta mo) <input type="radio"/> Sinäkin <input type="radio"/> Hän <input type="radio"/> Niinkö	Sivu 3
がくせい (ga ku se i) <input type="radio"/> Suomalainen <input type="radio"/> Opiskelija <input type="radio"/> Japanilainen	にほんじん (ni ho n ji n) <input type="radio"/> Suomalainen <input type="radio"/> Japanilainen <input type="radio"/> Amerikkalainen	そうですか (so o de su ka) <input type="radio"/> Niinkö <input type="radio"/> Niin <input type="radio"/> Ei	Sivu 4
<input type="button" value="Tarkista"/>			Sivu 5
			Sivu 6
			Sivu 7
			Sivu 8
			Sivu 9
			Sivu 10
			Sivu 11
			Sivu 12
			Sivu 13
			Sivu 14
			Sivu 15
			Sivu 16
			Sivu 17
			Sivu 18

Ensimmäinen Edellinen Seuraava Viimeinen

Kuvio 18. Toistuva pohja harjoitustehtäville.

Merkkien opettelua sisältää alkuun kaksi vaihtoehtoa, Merkkien opettelua 1 ja Merkkien opettelua 2. Merkkien opettelua 1:ssä (kuvio 19.) käyttäjä yrittää arvata kolmen vaihtoehdon joukosta oikean. Kyselyä voi muuttaa siten, että siinä arvataan joko hiraganoja, katakanoja tai niiden käännöstä. Merkkien opettelua 2 toimii samalla periaatteella, mutta sisältää kaikki merkit samaan aikaan, joiden joukosta arvottu vastaus pitää valita. ”Hiraganat” ja ”Katakana” sivut eivät sisällä mitään erikoista, vaan molemmat sisältävät nimensä mukaisen listan kaikista merkeistä ja niiden käännöksen (kuvio 20.).

i vastaa merkkiä:

い
 で
 じゆ

a = ?

Harjoittele Katakanoja

Kuvio 19. ”Merkkien opettelua 1” harjoitus

ETUSIVULLE						ETUSIVULLE																
	ア	A	イ	I	ウ	U	エ	E	オ	O		あ	A	い	I	う	U	え	E	お	O	
<i>Etusivu</i>	カ	KA	キ	KI	ク	KU	ケ	KE	コ	KO	<i>Etusivu</i>	か	KA	き	KI	く	KU	け	KE	こ	KO	
<i>Harjoitukset</i>	サ	SA	シ	SHI	ス	SU	セ	SE	ソ	SO	<i>Harjoitukset</i>	さ	SA	し	SHI	す	SU	せ	SE	そ	SO	
<i>Merkkien opettelua</i>	タ	TA	チ	CHI	ツ	TSU	テ	TE	ト	TO	<i>Merkkien opettelua</i>	た	TA	ち	CHI	つ	TSU	て	TE	と	TO	
<i>Hiraganat</i>	ナ	NA	ニ	NI	ヌ	NU	ネ	NE	ノ	NO	<i>Hiraganat</i>	な	NA	に	NI	ぬ	NU	ね	NE	の	NO	
<i>Katakana</i>	ハ	HA	ヒ	HI	フ	FU	ヘ	HE	ホ	HO	<i>Katakana</i>	は	HA	ひ	HI	ふ	FU	へ	HE	ほ	HO	
	マ	MA	ミ	MI	ム	MU	メ	ME	モ	MO		ま	MA	み	MI	む	MU	め	ME	も	MO	
	ヤ	YA			ユ	YU			ヨ	YO		や	YA			ゆ	YU		よ	YO		
	ラ	RA	リ	RI	ル	RU	レ	RE	ロ	RO		ら	RA	り	RI	る	RU	れ	RE	ろ	RO	
	ワ	WA								ヲ		わ	WA								を	O
	ン	N										ん	N									
	ガ	GA	ギ	GI	グ	GU	ゲ	GE	ゴ	GO		が	GA	ぎ	GI	ぐ	GU	げ	GE	ご	GO	
	ザ	ZA	ジ	JI	ズ	ZU	ゼ	ZE	ゾ	ZO		ざ	ZA	じ	JI	ず	ZU	ぜ	ZE	ぞ	ZO	

Kuvio 20. Hiragana ja katakana listat.

5.2 Toteutustekniikka

Toteutukseen käytettiin Microsoft Silverlight 3 ja Microsoft Expression Blend 3 versioita. Ohjelmoinnin toteutus tapahtui suurimmaksi osaksi käyttäen Visual Studiota, mutta tietyissä kohdissa oli apuna Microsoft Expression Blend. Blend:ä käytettiin tyylien kehittämiseen. Taustavärit, napit ja muita kontrolleja oli luotu ja muokattu Blend:ssä, jotka sitten lisättiin itse projektiin. Napit ovat Ellipse, Rectangle tai Path muotoja. Ellipse on ellipsi, Rectangle on neliö ja Path on vapaasti piirrettävä muoto. Muodosta tehtiin Blend:n avulla kontrolli ja lisättiin projektiin. Kaikki muu ohjelmointi tapahtui Visual Studiossa joko XAML:ssä tai C#:lla. Aiheesta oli alkuun olemassa suunnitelma, mutta kehitys kulki enemmän kokeilupohjalta ja suunnitelma muuttui moneen kertaan alkuperäisestä.

5.3 Luokat ja metodit

Ohjelmointityö sisältää yhden luokkadokumentin (DragAndDrop), jota on käytetty harjoitustehtävissä oleviin sanastoihin. Tämä luokka mahdollistaa sanaston liikuttamisen hiirellä. Luokka käyttää Canvas kontrollia, jotta sisältöä voi liikuttaa näytöllä vapaasti. Liikutettava sisältö muuttuu läpinäkyväksi, kun sen liikuttaminen on alkanut. Luokan koodin voi nähdä liitteessä 4.

Merkkien opetteluun liittyvässä koodissa on käytetty metodia, joka luo listan hiraganoista ja niiden käännöksen. Katakanoille on käytetty samaa metodia, ainoana erona hiraganojen tilalla ovat katakanat. Metodi käyttää listan luomiseen luokkaa, joka ottaa vastaan japanilaisen merkin ja merkkiä vastaavan käännöksen. Metodia ja luokkaa voi katsoa liitteestä 5. Merkkien opettelua sisältää myös metodin, jolla hallitaan tehtävämuotoa. Tässä metodissa määritellään mitä käyttäjän pitää tehdä, esim. kysytäänkö tehtävässä hiraganoja vai katakanoja. Tätä metodia voi tarkastella liitteessä 6. Tapahtumat keskittyvät tarkistamaan vastauksia ja muuttamaan hiiren kursoria, kursorin ollessa napin päällä.

Harjoitukset sivu sisältää metodin, joka listaa kaikki harjoitukset ja lisää ne Items list-kontrolliin. Tämän koodi löytyy liitteestä 7. Kyseinen metodi käyttää listauksessa apunaan luokkaa, joka sisältää itse harjoitussivun ja nimen, joka näytetään Items list-kontrollin näkyvässä. Tapahtumat koskevat näiden sivujen selaamista ja vierityspalkin liikuttamista.

Kaikki harjoitussivut ovat ”kovakoodattuja”, kuin myös kaikki muukin sovelluksen sisältö. Harjoitussivut ovat suurimmaksi osaksi tekstipohjaisia, joten niissä ei ole käytetty metodeja. Sivut, joissa on japanin kielen sanasto, käytetään mainittua DragAndDrop-luokkaa. Harjoitustehtävät sisältävät tarkistustapahtuman, jonka avulla lasketaan pisteet. Tämän tapahtuman koodi on liitteessä 8.

TemplatePage toimii sovelluksen pohjana, jossa sivuja selataan. Tämä sisältää useita tapahtumia, mutta ei lainkaan metodeja. Tapahtumat ovat suurimmaksi osaksi sivustolla suunnistamista varten. Tämä sisältää myös tapahtuman, jossa vaihdetaan sivuston teemaa. Teeman vaihtuessa muokataan listaa, joka sisältää kaikki käytettävät tyylit (Style). Liitteessä 9. on teemanvaihtotapahtuma.

Sivustojen välillä liikkuu lista, jossa on kaikki käytettävät tyylit (Style). Tämän teemalistan sisältö määritellään oletuksena MainPage User Control:ssa ja sitä voidaan vaihtaa TemplatePage:ssa. Avatessa uutta sivua, välitetään sen User Control:lle teemalista, jonka perusteella sivun ulkonäkö määritellään. Teemalistan tyylit on luotu projektin App.xaml dokumentissa. Liitteessä 10. on osa App.xaml dokumentin sisällöstä. Liitteestä 11. löytyy kaikki ohjelmointityössä olevat luokat ja User Control:t.

5.4 Testaus

Virheiden löytäminen ja korjaaminen on tapahtunut suurimmaksi osaksi ”erehdys ja ymmärrys” menetelmällä. Sivuston suunnittelu on aloitettu kolme kertaa alusta, koska Silverlight on vielä uusi ja sen opettelu tapahtui samaan aikaan kuin sivustoa työstettiin. Alkuun tehtiin vain käyttöliittymä, joka varmasti toimii ja sen jälkeen kehitettiin ulkonäköä. Sovellukselle tehtiin eri tyylivaihtoehtoja eli teemoja, joista voi valita mieleisen ulkonäön sivustolle.

Valmista työtä on testattu ulkopuolisten avulla, jotka ovat kokeilleet toimivuutta omilla koneillaan. Selvisi ettei japanin kielen merkit näy, ellei kielituki ole asennettu koneelle. Sivustoa on testattu Mozilla Firefox ja Internet Explorer selaimilla, joissa se on toiminut moitteettomasti. Työtä on testattu joka vaiheessa, kun jokin uusi osa-alue oli saatu valmiiksi. Aina jonkinlaisen toimivan kokonaisuuden valmistuttua, testattiin koko projektin toimivuutta ja tehtiin tarpeelliset korjaukset, ennen kuin uusia toimintoja lisättiin. Työn sisältöä ja sen tarkoituksen saavuttamista on testattu vapaaehtoisten toimesta, joiden antaman palautteen perusteella on tehty korjauksia. Käyttäjien kokemuksesta on kirjoitettu seuraavassa osiossa.

5.5 Käyttäjien kokemus

Käyttäjille oli annettu kysely, johon he antoivat arvostelun ja palautteen. Palautteessa he kertoivat mitkä asiat toimivat ja mitä pitää muuttaa. Sivuston käyttö ilmeni helpoksi käyttää eikä kelläkään tuntunut olevan ongelmaa sen kanssa. Sisällön määrästä kysyttiin oliko sitä liikaa tai liian vähän ja kaikki vastasivat määrän olevan keskitasoa. Ulkonäköä osa piti keskitasoa parempana ja osa huonompana. Sisällön asettelu oli käyttäjien mieleen, joka oli selkeä ja järkevä. Tiettyjä muutoksia tehtiin sivustolle palautteen mukaan. Etusivun ulkonäkö sai suurimman muokkauksen. Etusivu oli aiemmin muista sivuista eriävän näköinen, eikä se käyttänyt samaa pohjaa kuin kaikki muut. Etusivun toteutuksesta tehtiin samanlainen kuin kaikista muistakin sivuista. Etusivulta poistettiin linkit muille sivuille ja tekstiä lisättiin. Harjoitusten ja tehtävien fonttikokoa muutettiin, että niitä on helpompi lukea. Harjoitusten suunnistamiseen tarkoitettujen nappien järjestystä muutettiin ja niille lisättiin teksti, jossa sanotaan mitä se tekee. Merkkien opettelemisen ensimmäistä harjoitusta muutettiin pysymään samassa tehtävässä, kunnes annetaan oikea vastaus.

6 ANALYYSI JA JOHTOPÄÄTÖKSET

Alkusuunnitelmassa suurin virhe oli Japanin kielen alkeiden laajuuden aliarvioiminen, joten harjoitusten määrää jouduttiin vähentämään roimasti ja sivustosta tuli vain tutustumisväline aiheeseen. Virtuaalinäppäimistö oli myös alkuun tarkoituksena ottaa mukaan, mutta se ilmeni tarpeettomaksi ja hankalaksi toteuttaa. Sivustolla oli tarkoitus olla tehtäviä, joihin olisi pitänyt kirjoittaa japania. Nämä tehtävät jätettiin pois, koska harjoitusten määrää karsittiin ja käyttäjillä ei ole välttämättä käytössään japanin kielen näppäimistöä.

Silverlight:n käyttö oli helppoa, kun sen sai sisäistettyä. Syyskuussa 2009 julkaistiin Silverlight 3, joten ohjelmointityö kuului ensimmäisten sillä tehtyjen sovellusten joukkoon. Versio 3 toi Silverlight:lle valtavasti lisäyksiä ja korjauksia. Tiedon hankkiminen oli aluksi hankalaa, sillä melkein kaikki tieto koski aiempaa versiota. Ohjelmointityön näkökulmasta Silverlight:ssa ei ilmennyt mitään puutteita. Expression Blend:ä käytettiin ulkonäön kehittämiseen ja kontrollien luomiseen.

Silverlight ja Expression Blend ovat hyviä työkaluja sivustojen luomiseen, mutta vaativat jonkin verran opettelua. On vaikea sanoa onko helpompi työskennellä suoraan XAML:ään kirjoittamalla vai käyttämällä graafista Blend:ä sivujen luomiseen. Sivustoa luodessa ei ole tarpeen käyttää vain yhtä, vaan Blend:ä ja Visual Studioa voi käyttää samaan aikaan.

Teknillisesti sivusto toimii kuten pitääkin, mutta se että ajaako sivusto tarkoituksensa, riippuu hyvin paljon myös sen käyttäjästä. Huonosti motivoitunut käyttäjä ei opi sovelluksen avulla mitään. Sivustoa on testattu Mozilla Firefoxilla ja Internet Explorerilla, joissa se toimii moitteettomasti. Japanilaiset merkit eivät näy, jos tietokoneelle ei ole asennettu kielitukea. Sivustolla liikkumien sujuu helposti ja latausajat ovat pieniä.

Sivuston sisältö ei ole raskas joten se selaaminen tulee olemaan käyttäjälle nopeaa. Sivustolla olevista väriteemoista käyttäjä voi valita mieleisensä, joten lukeminen on käyttäjän mieleen. Harjoitusten selaaminen on nopeaa ja sivujen välillä ei ole rasittavia latausaikoja. Käyttäjän on helpompi löytää sivu, mitä hän jäi viimeksi lukemaan. Sivusto täyttää tarkoituksena, jos käyttäjä on oikeasti kiinnostunut siitä. Teknillisesti sivusto toimii, mutta se ei kykene maagisesti opettamaan kieltä henkilölle, joka ei ole kiinnostunut tai motivoitunut. Sivusto ei suinkaan valmista käyttäjää ulkomaanmatkalle Japaniin, vaan sen tarkoitus on vain tutustuttaa kieleen ja sen opetteluun.

7 YHTEENVETO

Ohjelmointityön alkusuunnitelman epärealistisuus ilmeni äkkiä ja suunnitelmia tuli muutettua kahteen kertaan. Vaikka päätin pienentää työtä vähentämällä harjoitusten määrää, työmäärä oli silti suuri. Huomioon ottaen, etten ole opiskellut japanin kieltä, muuten kuin itsenäisesti perusteita ja opettelin Silverlight:n ja Expression Blend:n käytön samaan aikaan, olen jotakuinkin tyytyväinen aikaansaannokseeni. Tekisin kaiken todennäköisesti erilailla, jos aloittaisin alusta. Ei siksi että olen pettynyt nykyiseen, vaan koska olen varma että osaan tehdä nyt paremman. C#-ohjelmointia lukuun ottamatta, kaikki oli uutta ja eteneminen hidasta ja kömpelöä. Suurin virhe työssä oli opetettavan asian sisällön aliarvioiminen. Silverlight:n käytön sain sisäistettyä, mutta kielen opettajaksi on vielä pitkä matka.

Oma mielipiteeni Silverlight:n käytöstä on positiivinen. Se on helppo ja monipuolinen käyttää, kunhan sen saa sisäistettyä. Huonoin puoli on mahdollisesti se, etteivät monet tiedä mikä Silverlight on. Monet Internetissä liikkujat todennäköisesti karttavat sivuja, jotka vaativat uusien lisäosien asentamista, jonka kaikki Silverlight sivut vaativat. Mozilla Firefox saattaa vaatia käyttäjää ottamaan Silverlight liitännäisen käyttöön manuaalisesti, muuten sivut eivät toimi. Tästä Firefox ei ilmoita, joten vastaus pitää etsiä Internetistä tai kysymällä kaverilta. Tällä hetkellä Silverlight ja Expression ovat monille tuntemattomia ja niitä oudoksutaan.

LÄHTEET

- Karppinen Takako, 2003. Japanin kielen alkeet. Oy Finn Lectura Ab, Helsinki.
- Moroney Laurence, 2008. Introducing Microsoft Silverlight 2. Microsoft Press, Washington , Redmond.
- Adobe, 2009. Adobe Flash CS4 Professional. <http://www.adobe.com/products/flash/> (Luettu 7.12.2009).
- C# Station, 2009. What is C"? <http://www.csharp-station.com/> (Luettu 7.12.2009).
- CMS WiRE, 2009. IIS Media Services Offer Smooth Video Streaming for Silverlight. <http://www.cmswire.com/cms/web-cms/iis-media-services-offer-smooth-video-streaming-for-silverlight-004167.php> (Luettu 7.12.2009).
- Garret Jesse James, 2005. Ajax: A New Approach to Web Application. Artikkele adaptive path:issa 18.2.2005. <http://www.adaptivepath.com/ideas/essays/archives/000385.php> (Luettu 31.8.2009)
- Jukka, 2009. Silverlight 3 tuo parannuksia videon laatuun ja suorituskykyyn. http://blogs.msdn.com/dpe_finland/archive/2009/03/25/silverlight-3-tuo-parannuksia-videon-laatuun-ja-suorituskykyyn.aspx (Luettu 1.9.2009).
- Kyrnin Jennifer, 2009. About Microsoft Expression Web. <http://webdesign.about.com/od/expressionweb/p/aaexpressionweb.htm> (Luettu 21.10.2009).
- Lehto Tero, 2008. Microsoft Silverlight 2 valmistui. Artikkele tietokone.fi:ssä 14.10.2008. http://www.tietokone.fi/uutta/uutinen.asp?news_id=35285 (Luettu 1.9.2009).
- Loomis Jay, Wasson Mike, 2007. VC-1 Technical Overview. <http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx> (Luettu 30.11.2009).
- Loponen Tapio, 2007. Silverlight 1.0. Artikkele tietokone.fi:ssä 7.9.2007. <http://www.tietokone.fi/softa/tiedosto.asp?id=13400> (Luettu 1.9.2009).
- Microsoft Corporation, 2009. Microsoft Silverlight Release History. Webdokumentti. Saatavilla: <http://download.microsoft.com/download/F/D/8/FD8F5947-78FF-40A5-9A34-5E35243C12E4/40115/Microsoft%20Silverlight%20Release%20History.html> (Luettu 31.8.2009).
- Microsoft Expression, 2009. <http://www.microsoft.com/expression/> (Luettu 2.9.2009).
- Microsoft Expression, 2009. Microsoft Expression Web 3. http://www.microsoft.com/Expression/products/Web_Overview.aspx (Luettu 21.10.2009).

- Microsoft Silverlight, 2009. All Features.
<http://www.microsoft.com/silverlight/overview/top-features/all-features.aspx#out> (Luettu 23.11.2009).
- Microsoft Silverlight, 2009. What is Silverlight?,
<http://www.microsoft.com/silverlight/overview/default.aspx> (Luettu 23.11.2009).
- Microsoft Silverlight, 2009. WHAT'S NEW IN SILVERLIGHT 3?
<http://silverlight.net/getstarted/silverlight3/default.aspx> (Luettu 1.9.2009).
- msdn, 2009. UserControl Class. <http://msdn.microsoft.com/en-us/library/system.windows.forms.usercontrol.aspx> (Luettu 30.11.2009).
- Muhammed Usama Alam, 2009. Flash vs. Silverlight : What Suits your needs best?
<http://www.smashingmagazine.com/2009/05/09/flash-vs-silverlight-what-suits-your-needs-best/> (Luettu 23.11.2009).
- RustyBrick, 2008. Definitions. <http://www.rustybrick.com/definitions.php> (Luettu 7.12.2009).
- Sai Stuff to Developers, 2007. History of Silverlight and Importance of it.
<http://saimaterial.wordpress.com/2007/10/27/history-of-silverlight-and-importance-of-it/> (Luettu 31.8.2009).
- Sanakirjani.fi, 2009. Affiksi. <http://www.sanakirjani.fi/kaannos/Isolaattikieli> (Luettu 30.11.2009).
- ScottGu, 2009. First Look at IronRuby. . Artikkele ScottGun blogissa 23.7.2007.
<http://weblogs.asp.net/scottgu/archive/2007/07/23/first-look-at-ironruby.aspx> (Luettu 7.12.2009).
- ScottGu, 2009. Silverlight 3 released. Artikkele ScottGun blogissa 10.7.2009.
<http://weblogs.asp.net/scottgu/archive/2009/07/10/silverlight-3-released.aspx> (Luettu 1.9.2009).
- Silja, 2006 – 2009. Japanin kielestä. <http://www.kanjikaveri.net/yleista/japani.php> (Luettu 23.11.2009).
- Silja, 2006 – 2009. Kirjoitusjärjestelmä.
<http://www.kanjikaveri.net/yleista/kirjoitusjarjestelma.php> (Luettu 23.11.2009).
- Small Business Computing.com, 2009. Rich Internet Application.
http://sbc.webopedia.com/TERM/R/Rich_Internet_Application.html (Luettu 7.12.2009).
- Small Business Computing.com, 2009. Visual Basic.
http://sbc.webopedia.com/TERM/V/Visual_Basic.html (Luettu 7.12.2009).
- Small Business Computing.com, 2009. WMA.
<http://sbc.webopedia.com/TERM/W/WMA.html> (Luettu 7.12.2009).
- Sun Microsystems, 2009. JavaScript Language for Java Developers.
<http://www.sun.com/training/catalog/courses/WJO-2123.xml> (Luettu 30.11.2009).

Vcodex, 2009. Overview of H.264/AVC. <http://www.vcodex.com/h264overview.html> (Luettu 7.12.2009).

Whatis?, 2008. WYSIWYG. http://whatis.techtarget.com/definition/0,,sid9_gci213392,00.html (Luettu 21.10.2009).

WindowsClient.NET, 2009. Windows Presentation Foundation. <http://windowsclient.net/wpf/> (Luettu 30.11.2009).

xprblog, 2007. EXPRESSION STUDIO RTM & SILVERLIGHT NEWS. Artikkele MSDN Blogissa 30.4.2007. <http://blogs.msdn.com/expression/archive/2007/04/30/expression-studio-rtm-silverlight-news.aspx> (Luettu 2.9.2009)..

LIITE LUETTELO

- Liite 1. Tutoriaalin lopullinen sisältö
- Liite 2. Valmis tutoriaali
- Liite 3. Tutoriaalin lisäharjoitus
- Liite 4. Drag-And-Drop – luokka
- Liite 5. Merkkien listausmetodi ja luokka
- Liite 6. Tehtävämallin määrittely
- Liite 7. Harjoitussivujen lisäys
- Liite 8. Tehtävän tarkistus
- Liite 9. Teemanvaihtotapahtuma
- Liite 10. App.xaml dokumentti
- Liite 11. Luokat ja User Control:t

TUTORIAALIN LOPULLINEN SISÄLTÖ

MainPage.xaml.

```

<UserControl x:Class="Tutoriaali.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">
  <Grid x:Name="LayoutRoot"
    MouseLeftButtonUp="LayoutRoot_MouseLeftButtonUp" Background="White">

    <Grid.RowDefinitions>
      <RowDefinition Height="1*" />
      <RowDefinition Height="6*" />
    </Grid.RowDefinitions>

    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="100" />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>

    <Canvas x:Name="Etusivu"
      Grid.Column="0"
      Grid.Row="0"
      Background="Aqua"
      MouseLeftButtonDown="Etusivu_MouseLeftButtonDown"
      MouseLeftButtonUp="Etusivu_MouseLeftButtonUp">

      <TextBlock Text="Etusivulle"
        FontSize="16"
        FontFamily="Arial"
        FontWeight="Bold" />
    </Canvas>

    <StackPanel Orientation="Vertical"
      Grid.Column="0"
      Grid.Row="1">
      <Button x:Name="btnEtusivu"
        Content="Etusivu"
        Height="30"
        Click="btnEtusivu_Click" />
      <Button x:Name="btnToinenSivu"
        Content="Toinen Sivu"
        Height="30"
        Click="btnToinenSivu_Click" />
    </StackPanel>

    <ScrollViewer x:Name="scrlView"
      Grid.Column="1"
      Grid.Row="1"
      Background="Aquamarine"
      VerticalScrollBarVisibility="Auto"
      HorizontalScrollBarVisibility="Auto"

```

```

        MouseWheel="scrlView_MouseWheel">

        </ScrollViewer>

    </Grid>
</UserControl>

```

MainPage.xaml.cs

```

private bool _bOk = false;

public MainPage()
{
    InitializeComponent();
}

private void btnToinenSivu_Click(object sender, RoutedEventArgs e)
{
    scrlView.Content = new ToinenSivu();
}

private void btnEtusivu_Click(object sender, RoutedEventArgs e)
{
    scrlView.Content = null;
}

private void LayoutRoot_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{
    _bOk = false;
}

private void Etusivu_MouseLeftButtonDown(object sender,
MouseButtonEventArgs e)
{
    _bOk = true;
}

private void Etusivu_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{
    if (_bOk)
    {
        scrlView.Content = null;
    }
}

private void scrlView_MouseWheel(object sender, MouseWheelEventArgs e)
{
    if (e.Delta < 0)
        scrlView.ScrollToVerticalOffset(scrlView.VerticalOffset + 30);
    else
        scrlView.ScrollToVerticalOffset(scrlView.VerticalOffset - 30);
}

```

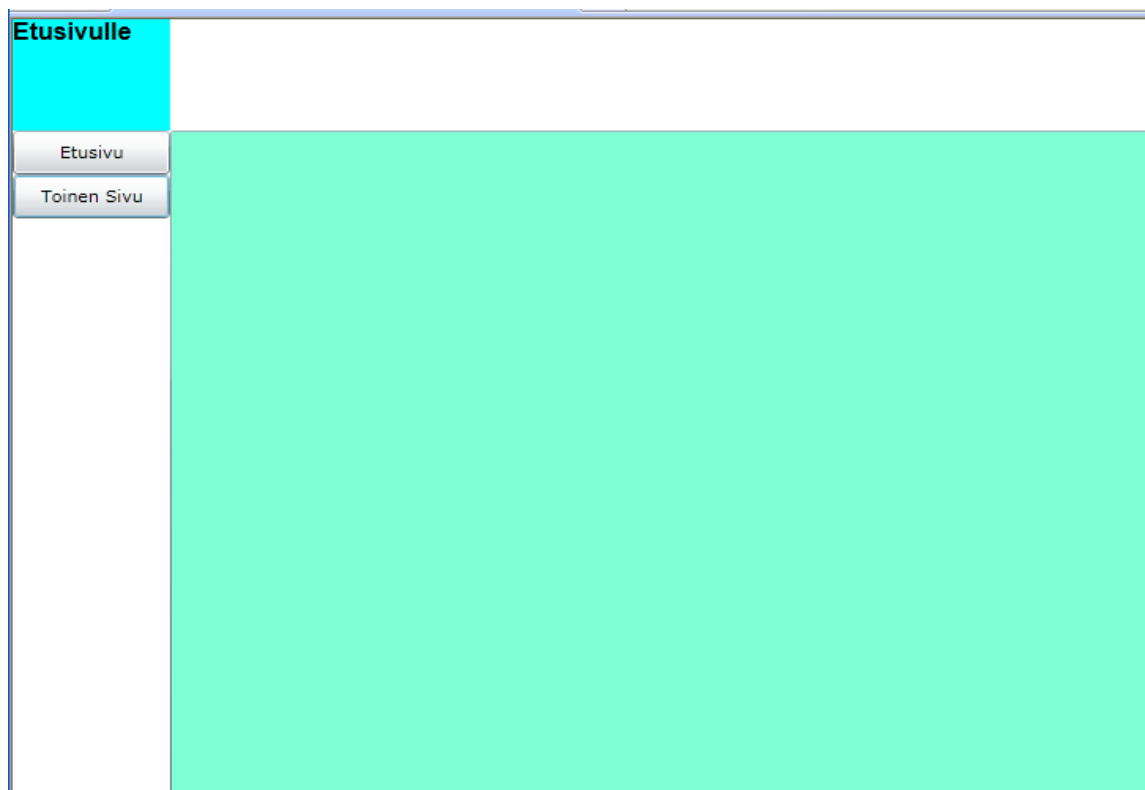
ToinenSivu.xaml

```
<UserControl x:Class="Tutoriaali.ToinenSivu"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Grid x:Name="LayoutRoot" Background="Black">

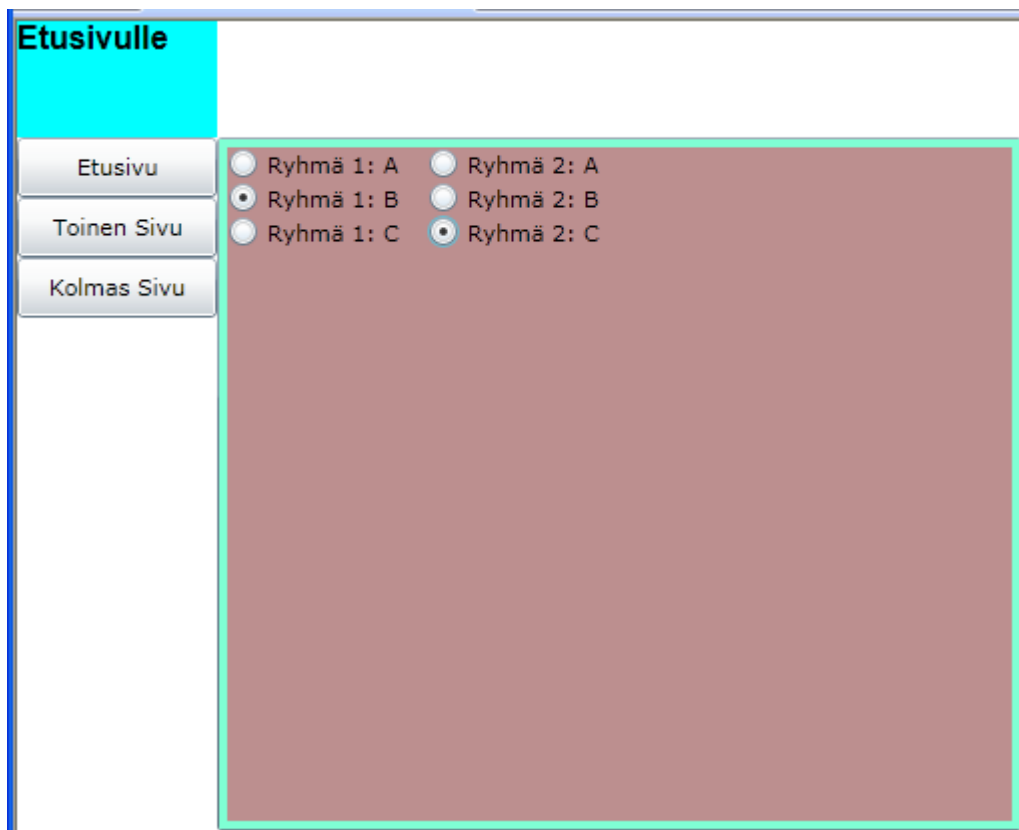
    <TextBlock FontSize="100"
      Foreground="White">
      <TextBlock.Text>
Terve!
Tämä
on
toinen
sivu!
      </TextBlock.Text>
    </TextBlock>

  </Grid>
</UserControl>
```

VALMIS TUTORIAALI



TUTORIAALIN LISÄHARJOITUS



MainPage.xaml ja MainPage.xaml.cs

```
<StackPanel Orientation="Vertical"
  Grid.Column="0"
  Grid.Row="1">
  <Button x:Name="btnEtusivu"
    Content="Etusivu"
    Height="30"
    Click="btnEtusivu_Click"/>
  <Button x:Name="btnToinenSivu"
    Content="Toinen Sivu"
    Height="30"
    Click="btnToinenSivu_Click"/>
  <Button x:Name="btnKolmasSivu"
    Content="Kolmas Sivu"
    Height="30"
    Click="btnKolmasSivu_Click"/>
</StackPanel>

private void btnKolmasSivu_Click(object sender, RoutedEventArgs e)
{
    scr1View.Content = new KolmasSivu();
}
```


DRAG-AND-DROP - LUOKKA

```

public class DragAndDrop : Canvas
{
    Point _pBegin;
    Point _pCurrent;
    bool _bDrag = false;
    public DragAndDrop() : base()
    {
        this.MouseLeftButtonDown += new
        MouseButtonEventHandler(DragDropPanel_MouseLeftButtonDown);
        this.MouseLeftButtonUp += new
        MouseButtonEventHandler(DragDropPanel_MouseLeftButtonUp);
        this.MouseMove += new MouseEventArgs(DragDropPanel_MouseMove);
        this.Cursor = Cursors.Hand;
    }

    void DragDropPanel_MouseMove(object sender, MouseEventArgs e)
    {
        if (_bDrag)
        {
            _pCurrent = e.GetPosition(null);
            double dX =
            System.Convert.ToDouble(this.GetValue(Canvas.LeftProperty));
            double dY =
            System.Convert.ToDouble(this.GetValue(Canvas.TopProperty));
            this.SetValue(Canvas.LeftProperty, dX + _pCurrent.X - _pBegin.X);
            this.SetValue(Canvas.TopProperty, dY + _pCurrent.Y - _pBegin.Y);
            _pBegin = _pCurrent;
        }
    }

    void DragDropPanel_MouseLeftButtonUp(object sender,
    MouseButtonEventArgs e)
    {
        if (_bDrag)
        {
            this.Opacity *= 2;
            this.ReleaseMouseCapture();
            _bDrag = false;
        }
    }

    void DragDropPanel_MouseLeftButtonDown(object sender,
    MouseButtonEventArgs e)
    {
        FrameworkElement c = sender as FrameworkElement;
        _bDrag = true;
        _pBegin = e.GetPosition(null);
        c.Opacity *= 0.5;
        c.CaptureMouse();
    }
}

```

MERKKIEN LISTAUSMETODI JA LUOKKA

```
private void SetKatakanaList()
{
    _listKatakana = new List<Katakana_Vastaus>();

    _listKatakana.Add(new Katakana_Vastaus("a", "ア"));
    _listKatakana.Add(new Katakana_Vastaus("i", "イ"));
    _listKatakana.Add(new Katakana_Vastaus("u", "ウ"));
    _listKatakana.Add(new Katakana_Vastaus("e", "エ"));
    _listKatakana.Add(new Katakana_Vastaus("o", "オ"));

    _listKatakana.Add(new Katakana_Vastaus("ka", "カ"));
    _listKatakana.Add(new Katakana_Vastaus("ki", "キ"));
    _listKatakana.Add(new Katakana_Vastaus("ku", "ク"));
    _listKatakana.Add(new Katakana_Vastaus("ke", "ケ"));
    _listKatakana.Add(new Katakana_Vastaus("ko", "コ"));

    _listKatakana.Add(new Katakana_Vastaus("sa", "サ"));
    _listKatakana.Add(new Katakana_Vastaus("shi", "シ"));
    _listKatakana.Add(new Katakana_Vastaus("su", "ス"));
    _listKatakana.Add(new Katakana_Vastaus("se", "セ"));
    _listKatakana.Add(new Katakana_Vastaus("so", "ソ"));

    _listKatakana.Add(new Katakana_Vastaus("ta", "タ"));
    _listKatakana.Add(new Katakana_Vastaus("chi", "チ"));
    _listKatakana.Add(new Katakana_Vastaus("tsu", "ツ"));
    _listKatakana.Add(new Katakana_Vastaus("te", "テ"));
    _listKatakana.Add(new Katakana_Vastaus("to", "ト"));

    _listKatakana.Add(new Katakana_Vastaus("na", "ナ"));
    _listKatakana.Add(new Katakana_Vastaus("ni", "ニ"));
    _listKatakana.Add(new Katakana_Vastaus("nu", "ヌ"));
    _listKatakana.Add(new Katakana_Vastaus("ne", "ネ"));
    _listKatakana.Add(new Katakana_Vastaus("no", "ノ"));

    _listKatakana.Add(new Katakana_Vastaus("ha", "ハ"));
    _listKatakana.Add(new Katakana_Vastaus("hi", "ヒ"));
    _listKatakana.Add(new Katakana_Vastaus("fu", "フ"));
    _listKatakana.Add(new Katakana_Vastaus("he", "ヘ"));
    _listKatakana.Add(new Katakana_Vastaus("ho", "ホ"));
}
```

```
_listKatakana.Add(new Katakana_Vastaus("ma", "マ"));
_listKatakana.Add(new Katakana_Vastaus("mi", "ミ"));
_listKatakana.Add(new Katakana_Vastaus("mu", "ム"));
_listKatakana.Add(new Katakana_Vastaus("me", "メ"));
_listKatakana.Add(new Katakana_Vastaus("mo", "モ"));

_listKatakana.Add(new Katakana_Vastaus("ya", "ヤ"));
_listKatakana.Add(new Katakana_Vastaus("yu", "ユ"));
_listKatakana.Add(new Katakana_Vastaus("yo", "ヨ"));

_listKatakana.Add(new Katakana_Vastaus("ra", "ラ"));
_listKatakana.Add(new Katakana_Vastaus("ri", "リ"));
_listKatakana.Add(new Katakana_Vastaus("ru", "ル"));
_listKatakana.Add(new Katakana_Vastaus("re", "レ"));
_listKatakana.Add(new Katakana_Vastaus("ro", "ロ"));

_listKatakana.Add(new Katakana_Vastaus("wa", "ワ"));
_listKatakana.Add(new Katakana_Vastaus("o", "ヲ"));
_listKatakana.Add(new Katakana_Vastaus("n", "ン"));

_listKatakana.Add(new Katakana_Vastaus("ga", "ガ"));
_listKatakana.Add(new Katakana_Vastaus("gi", "ギ"));
_listKatakana.Add(new Katakana_Vastaus("gu", "グ"));
_listKatakana.Add(new Katakana_Vastaus("ge", "ゲ"));
_listKatakana.Add(new Katakana_Vastaus("go", "ゴ"));

_listKatakana.Add(new Katakana_Vastaus("za", "ザ"));
_listKatakana.Add(new Katakana_Vastaus("ji", "ジ"));
_listKatakana.Add(new Katakana_Vastaus("zu", "ズ"));
_listKatakana.Add(new Katakana_Vastaus("ze", "ゼ"));
_listKatakana.Add(new Katakana_Vastaus("zo", "ゾ"));

_listKatakana.Add(new Katakana_Vastaus("da", "ダ"));
_listKatakana.Add(new Katakana_Vastaus("ji", "ヂ"));
_listKatakana.Add(new Katakana_Vastaus("zu", "ヅ"));
_listKatakana.Add(new Katakana_Vastaus("de", "デ"));
_listKatakana.Add(new Katakana_Vastaus("do", "ド"));

_listKatakana.Add(new Katakana_Vastaus("ba", "バ"));
_listKatakana.Add(new Katakana_Vastaus("bi", "ビ"));
_listKatakana.Add(new Katakana_Vastaus("bu", "ブ"));
```

```
_listKatakana.Add(new Katakana_Vastaus("be", "ベ"));
_listKatakana.Add(new Katakana_Vastaus("bo", "ボ"));

_listKatakana.Add(new Katakana_Vastaus("pa", "パ"));
_listKatakana.Add(new Katakana_Vastaus("pi", "ピ"));
_listKatakana.Add(new Katakana_Vastaus("pu", "プ"));
_listKatakana.Add(new Katakana_Vastaus("pe", "ペ"));
_listKatakana.Add(new Katakana_Vastaus("po", "ポ"));

_listKatakana.Add(new Katakana_Vastaus("kya", "キャ"));
_listKatakana.Add(new Katakana_Vastaus("kyu", "キュ"));
_listKatakana.Add(new Katakana_Vastaus("kyo", "キョ"));

_listKatakana.Add(new Katakana_Vastaus("sha", "シャ"));
_listKatakana.Add(new Katakana_Vastaus("shu", "シュ"));
_listKatakana.Add(new Katakana_Vastaus("sho", "ショ"));

_listKatakana.Add(new Katakana_Vastaus("cha", "チャ"));
_listKatakana.Add(new Katakana_Vastaus("chu", "チュ"));
_listKatakana.Add(new Katakana_Vastaus("cho", "チョ"));

_listKatakana.Add(new Katakana_Vastaus("nya", "ニャ"));
_listKatakana.Add(new Katakana_Vastaus("nyu", "ニユ"));
_listKatakana.Add(new Katakana_Vastaus("nyo", "ニョ"));

_listKatakana.Add(new Katakana_Vastaus("hya", "ヒャ"));
_listKatakana.Add(new Katakana_Vastaus("hyu", "ヒユ"));
_listKatakana.Add(new Katakana_Vastaus("hyo", "ヒョ"));

_listKatakana.Add(new Katakana_Vastaus("mya", "ミャ"));
_listKatakana.Add(new Katakana_Vastaus("myu", "ミユ"));
_listKatakana.Add(new Katakana_Vastaus("myo", "ミョ"));

_listKatakana.Add(new Katakana_Vastaus("rya", "リャ"));
_listKatakana.Add(new Katakana_Vastaus("ryu", "リュ"));
_listKatakana.Add(new Katakana_Vastaus("ryo", "リョ"));

_listKatakana.Add(new Katakana_Vastaus("gya", "ギャ"));
_listKatakana.Add(new Katakana_Vastaus("gyu", "ギユ"));
_listKatakana.Add(new Katakana_Vastaus("gyo", "ギョ"));

_listKatakana.Add(new Katakana_Vastaus("ja", "ジャ"));
_listKatakana.Add(new Katakana_Vastaus("ju", "ジュ"));
```

```
_listKatakana.Add(new Katakana_Vastaus("jo", "ジョ"));

_listKatakana.Add(new Katakana_Vastaus("ja", "ジャ"));
_listKatakana.Add(new Katakana_Vastaus("ju", "ジュ"));
_listKatakana.Add(new Katakana_Vastaus("jo", "ジョ"));

_listKatakana.Add(new Katakana_Vastaus("bya", "ビャ"));
_listKatakana.Add(new Katakana_Vastaus("byu", "ビュ"));
_listKatakana.Add(new Katakana_Vastaus("byo", "ビョ"));

_listKatakana.Add(new Katakana_Vastaus("bya", "ピャ"));
_listKatakana.Add(new Katakana_Vastaus("byu", "ピュ"));
_listKatakana.Add(new Katakana_Vastaus("byo", "ピョ"));
}

private class Hiragana_Vastaus
{
    public string Hiragana;
    public string Aakkosissa;
    public Hiragana_Vastaus(string Aakkosissa, string Hiragana)
    {
        this.Hiragana = Hiragana;
        this.Aakkosissa = Aakkosissa;
    }
}
```

TEHTÄVÄMALLIN MÄÄRITTELEMINEN

```

private void SetOpettelu()
{
    if (_bKaannosAakkosista)
    {
        if (_bHarjoitusHiragana)
        {
            List<Hiragana_Vastaus> listArvaukset = new
            List<Hiragana_Vastaus>();

            Random random = new Random();
            Hiragana_Vastaus hira = _listHiragana[random.Next(0, 106)];
            tbVertaus.Tag = hira;
            tbVertaus.Text = hira.Aakkosissa;
            listArvaukset.Add(hira);

            while (listArvaukset.Count < 3)
            {
                bool ok = true;
                hira = _listHiragana[random.Next(0, 106)];
                foreach (Hiragana_Vastaus hiraTarkistus in listArvaukset)
                {
                    if (hira == hiraTarkistus)
                    {
                        ok = false;
                        break;
                    }
                }
                if (ok)
                    listArvaukset.Add(hira);
            }

            int ii = 3;
            foreach (Button btn in stackVaihtoehdot.Children)
            {
                int index = random.Next(0, ii);
                if (index < listArvaukset.Count)
                {
                    btn.Tag = listArvaukset[index];
                    btn.Content = listArvaukset[index].Hiragana;
                    listArvaukset.RemoveAt(index);
                    ii--;
                }
            }
        }
        else
        {
            List<Katakana_Vastaus> listArvaukset = new
            List<Katakana_Vastaus>();

            Random random = new Random();
            Katakana_Vastaus kata = _listKatakana[random.Next(0, 106)];
            tbVertaus.Tag = kata;
            tbVertaus.Text = kata.Aakkosissa;
            listArvaukset.Add(kata);
        }
    }
}

```

```

while (listArvaukset.Count < 3)
{
    bool ok = true;
    kata = _listKatakana[random.Next(0, 106)];
    foreach (Katakana_Vastaus kataTarkistus in listArvaukset)
    {
        if (kata == kataTarkistus)
        {
            ok = false;
            break;
        }
    }
    if (ok)
        listArvaukset.Add(kata);
}

int ii = 3;
foreach (Button btn in stackVaihtoehdot.Children)
{
    int index = random.Next(0, ii);
    if (index < listArvaukset.Count)
    {
        btn.Tag = listArvaukset[index];
        btn.Content = listArvaukset[index].Katakana;
        listArvaukset.RemoveAt(index);
        ii--;
    }
}
}

else
{
    if (_bHarjoitusHiragana)
    {
        List<Hiragana_Vastaus> listArvaukset = new
        List<Hiragana_Vastaus>();

        Random random = new Random();
        Hiragana_Vastaus hira = _listHiragana[random.Next(0, 106)];
        tbVertaus.Tag = hira;
        tbVertaus.Text = hira.Hiragana;
        listArvaukset.Add(hira);

        while (listArvaukset.Count < 3)
        {
            bool ok = true;
            hira = _listHiragana[random.Next(0, 106)];
            foreach (Hiragana_Vastaus hiraTarkistus in listArvaukset)
            {
                if (hira == hiraTarkistus)
                {
                    ok = false;
                    break;
                }
            }
            if (ok)
                listArvaukset.Add(hira);
        }
    }
}

```

```
int ii = 3;
foreach (Button btn in stackVaihtoehdot.Children)
{
    int index = random.Next(0, ii);
    if (index < listArvaukset.Count)
    {
        btn.Tag = listArvaukset[index];
        btn.Content = listArvaukset[index].Aakkosissa;
        listArvaukset.RemoveAt(index);
        ii--;
    }
}
else
{
    List<Katakana_Vastaus> listArvaukset = new
    List<Katakana_Vastaus>();

    Random random = new Random();
    Katakana_Vastaus kata = _listKatakana[random.Next(0, 106)];
    tbVertaus.Tag = kata;
    tbVertaus.Text = kata.Katakana;
    listArvaukset.Add(kata);

    while (listArvaukset.Count < 3)
    {
        bool ok = true;
        kata = _listKatakana[random.Next(0, 106)];
        foreach (Katakana_Vastaus kataTarkistus in listArvaukset)
        {
            if (kata == kataTarkistus)
            {
                ok = false;
                break;
            }
        }
        if (ok)
            listArvaukset.Add(kata);
    }

    int ii = 3;
    foreach (Button btn in stackVaihtoehdot.Children)
    {
        int index = random.Next(0, ii);
        if (index < listArvaukset.Count)
        {
            btn.Tag = listArvaukset[index];
            btn.Content = listArvaukset[index].Aakkosissa;
            listArvaukset.RemoveAt(index);
            ii--;
        }
    }
}
}
```


HARJOITUSSIVUJEN LISÄYS

```
private void OwnInitialize()
{
    EllipseAnimation.Begin();
    List<HarjoitusSivut> listHarjoitukset = new
    List<HarjoitusSivut>();
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 1", new
    Harjoitukset.HarjoitusSivu1(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 2", new
    Harjoitukset.HarjoitusSivu2(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 3", new
    Harjoitukset.HarjoitusSivu3(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 4", new
    Harjoitukset.HarjoitusSivu4(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 5", new
    Harjoitukset.HarjoitusSivu5(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 6", new
    Harjoitukset.HarjoitusSivu7(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 7", new
    Harjoitukset.HarjoitusSivu8(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 8", new
    Harjoitukset.HarjoitusSivu9(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 9", new
    Harjoitukset.HarjoitusSivu6(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 10", new
    Harjoitukset.HarjoitusSivu10(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 11", new
    Harjoitukset.HarjoitusSivu11(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 12", new
    Harjoitukset.HarjoitusSivu12(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 13", new
    Harjoitukset.HarjoitusSivu16(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 14", new
    Harjoitukset.HarjoitusSivu17(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 15", new
    Harjoitukset.HarjoitusSivu18(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 16", new
    Harjoitukset.HarjoitusSivu13(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 17", new
    Harjoitukset.HarjoitusSivu14(_listStyles)));
    listHarjoitukset.Add(new HarjoitusSivut("Sivu 18", new
    Harjoitukset.HarjoitusSivu15(_listStyles)));

    listSivut.ItemsSource = listHarjoitukset;;
    listSivut.SelectedIndex = 0;
}
```

TEHTÄVÄN TARKISTUS

```
private void btnTarkistus_Click(object sender, RoutedEventArgs e)
{
    int ii = 0;

    if (VastausGroup1.IsChecked == true)
        ii++;
    if (VastausGroup2.IsChecked == true)
        ii++;
    if (VastausGroup3.IsChecked == true)
        ii++;
    if (VastausGroup4.IsChecked == true)
        ii++;
    if (VastausGroup5.IsChecked == true)
        ii++;
    if (VastausGroup6.IsChecked == true)
        ii++;
    if (VastausGroup7.IsChecked == true)
        ii++;
    if (VastausGroup8.IsChecked == true)
        ii++;
    if (VastausGroup9.IsChecked == true)
        ii++;

    txblPisteet.Text = ii + "/9";
}
```

TEEMANVAIHTOTAPAHTUMA

```

void gridStyles_MouseLeftButtonUp(object sender, MouseButtonEventArgs
e)
{
    if ((_bStylePressed == true) && (_sStyleName ==
((Grid)e.OriginalSource).Name))
    {
        if (((Grid)e.OriginalSource).Style ==
App.Current.Resources["StyleBackground1"] as Style)
        {
            _listStyles[0] = App.Current.Resources["StyleBackground1"] as
Style;
            _listStyles[1] = App.Current.Resources["TextBlockStyle1"] as
Style;
            _listStyles[2] = App.Current.Resources["StyleBackgroundSanasto1"]
as Style;
            _listStyles[3] = App.Current.Resources["StyleEllipse1"] as Style;
            _listStyles[4] = App.Current.Resources["StyleRectangleButton1"]
as Style;
            _listStyles[5] = App.Current.Resources["StylePageLink1"] as
Style;
            _listStyles[6] = App.Current.Resources["StyleList1"] as Style;
            _listStyles[7] = App.Current.Resources["StyleRadioButton1"] as
Style;
        }
        else if (((Grid)e.OriginalSource).Style ==
App.Current.Resources["StyleBackground2"] as Style)
        {
            _listStyles[0] = App.Current.Resources["StyleBackground2"] as
Style;
            _listStyles[1] = App.Current.Resources["TextBlockStyle1"] as
Style;
            _listStyles[2] = App.Current.Resources["StyleBackgroundSanasto2"]
as Style;
            _listStyles[3] = App.Current.Resources["StyleEllipse2"] as Style;
            _listStyles[4] = App.Current.Resources["StyleRectangleButton2"]
as Style;
            _listStyles[5] = App.Current.Resources["StylePageLink1"] as
Style;
            _listStyles[6] = App.Current.Resources["StyleList1"] as Style;
            _listStyles[7] = App.Current.Resources["StyleRadioButton1"] as
Style;
        }
        else if (((Grid)e.OriginalSource).Style ==
App.Current.Resources["StyleBackground3"] as Style)
        {
            _listStyles[0] = App.Current.Resources["StyleBackground3"] as
Style;
            _listStyles[1] = App.Current.Resources["TextBlockStyle3"] as
Style;
            _listStyles[2] = App.Current.Resources["StyleBackgroundSanasto3"]
as Style;
            _listStyles[3] = App.Current.Resources["StyleEllipse3"] as Style;
            _listStyles[4] = App.Current.Resources["StyleRectangleButton3"]
as Style;
        }
    }
}

```

```

        _listStyles[5] = App.Current.Resources["StylePageLink2"] as
        Style;
        _listStyles[6] = App.Current.Resources["StyleList2"] as Style;
        _listStyles[7] = App.Current.Resources["StyleRadioButton2"] as
        Style;
    }
    else if (((Grid)e.OriginalSource).Style ==
    App.Current.Resources["StyleBackground4"] as Style)
    {
        _listStyles[0] = App.Current.Resources["StyleBackground4"] as
        Style;
        _listStyles[1] = App.Current.Resources["TextBlockStyle3"] as
        Style;
        _listStyles[2] = App.Current.Resources["StyleBackgroundSanasto4"]
        as Style;
        _listStyles[3] = App.Current.Resources["StyleEllipse4"] as Style;
        _listStyles[4] = App.Current.Resources["StyleRectangleButton1"]
        as Style;
        _listStyles[5] = App.Current.Resources["StylePageLink2"] as
        Style;
        _listStyles[6] = App.Current.Resources["StyleList2"] as Style;
        _listStyles[7] = App.Current.Resources["StyleRadioButton2"] as
        Style;
    }
    else if (((Grid)e.OriginalSource).Style ==
    App.Current.Resources["StyleBackground5"] as Style)
    {
        _listStyles[0] = App.Current.Resources["StyleBackground5"] as
        Style;
        _listStyles[1] = App.Current.Resources["TextBlockStyle1"] as
        Style;
        _listStyles[2] = App.Current.Resources["StyleBackgroundSanasto5"]
        as Style;
        _listStyles[3] = App.Current.Resources["StyleEllipse5"] as Style;
        _listStyles[4] = App.Current.Resources["StyleRectangleButton4"]
        as Style;
        _listStyles[5] = App.Current.Resources["StylePageLink1"] as
        Style;
        _listStyles[6] = App.Current.Resources["StyleList1"] as Style;
        _listStyles[7] = App.Current.Resources["StyleRadioButton1"] as
        Style;
    }
    }

    MainPage main = this.Parent as MainPage;
    main.Navigate(new TemplatePage(new Page_Etusivu(_listStyles),
    _listStyles));
}
}

```

APP.XAML DOKUMENTTI

```

<Style x:Key="TextBlockStyle1" TargetType="TextBlock">
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontWeight" Value="Bold"/>
    <Setter Property="FontStyle" Value="Normal"/>
    <Setter Property="FontFamily" Value="Comic Sans MS"/>
</Style>

<Style x:Key="StyleRadioButton1" TargetType="RadioButton">
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="Foreground" Value="Black"/>
</Style>
<Style x:Key="StyleRadioButton2" TargetType="RadioButton">
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="Foreground" Value="White"/>
</Style>

<Style x:Key="StylePageLink1" TargetType="Button">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="Button">
                <Grid>
                    <VisualStateManager.VisualStateGroups>
                        <VisualStateGroup x:Name="CommonStates">
                            <VisualState x:Name="Pressed"/>
                        </VisualStateGroup>
                    </VisualStateManager.VisualStateGroups>
                    <ContentPresenter/>
                </Grid>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
    <Setter Property="FontWeight" Value="Bold"/>
    <Setter Property="FontStyle" Value="Italic"/>
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="Foreground" Value="Black"/>
</Style>

```

LUOKAT JA USER CONTROLIT

Luokasta ja User Control:sta on kerrottu mitä muuttujia, metodeja ja tapahtumia se sisältää.

TemplatePage bool _bHeaderClick List<Style> _listStyles string _sStyleName bool _bStylePressed TemplatePage(UserControl uc, List<Style> listStyles) LayoutRoot_MouseLeftButtonUp(object sender, MouseButtonEventArgs e) HeadGrid_MouseLeftButtonUp(object sender, MouseButtonEventArgs e) HeadGrid_MouseLeftButtonDown(object sender, MouseButtonEventArgs e) btnEtusivu_Click(object sender, RoutedEventArgs e) btnHarjoitukset_Click(object sender, RoutedEventArgs e) btnMerkkienOpettelua_Click(object sender, RoutedEventArgs e) btnHiraganat_Click(object sender, RoutedEventArgs e) btnKatakanat_Click(object sender, RoutedEventArgs e) gridStyles_MouseLeftButtonUp(object sender, MouseButtonEventArgs e) gridStyles_MouseLeftButtonDown(object sender, MouseButtonEventArgs e) scrViewer_MouseWheel(object sender, MouseWheelEventArgs e) Style_MouseEnter(object sender, MouseEventArgs e) Style_MouseLeave(object sender, MouseEventArgs e)	
MainPage MainPage() Navigate (UserControl nextpage)	Page_Etusivu Page_Etusivu(List<Style> listStyles)
Page_Katakana public Page_Katakana(Style style)	Page_Hiragana Page_Hiragana(Style style)

DragAndDrop
Point _pBegin Point _pCurrent bool _bDrag
DragAndDrop() DragDropPanel_MouseMove(object sender, MouseEventArgs e) DragDropPanel_MouseLeftButtonUp(object sender, MouseButtonEventArgs e) DragDropPanel_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

Page_Merkkien_Opettelua
List<Style> _listStyles
Page_MerkkienOpettelua(List<Style> listStyles) Opettelu1_Click(object sender, RoutedEventArgs e) Opettelu2_Click(object sender, RoutedEventArgs e) btn_MouseEnter(object sender, MouseEventArgs e) btn_MouseLeave(object sender, MouseEventArgs e)

Opettelu1
List<Hiragana_Vastaus> _listHiragana List<Katakana_Vastaus> _listKatakana bool _bHarjoitusHiragana bool _bKaannosAakkosista
Opettelu1(List<Style> listStyle) SetOpettelu() btnArvaus_Click(object sender, RoutedEventArgs e) btnVaihdaHarjoitusta_Click(object sender, RoutedEventArgs e) btnVaihdaKaannos_Click(object sender, RoutedEventArgs e) SetHiraganaList() SetKatakanaList() class Hiragana_Vastaus class Katakana_Vastaus btn_MouseEnter(object sender, MouseEventArgs e) btn_MouseLeave(object sender, MouseEventArgs e)

Opettelu2
bool _bHiragana List<Hiragana_Vastaus> _listHiragana List<Katakana_Vastaus> _listKatakana
Opettelu2(List<Style> listStyles) OwnInitialize() btnMerkki_Click(object sender, RoutedEventArgs e) SetButtons() SetHiraganaList() class Hiragana_Vastaus SetKatakanaList() class Katakana_Vastaus btnVaihda_Click(object sender, RoutedEventArgs e) btnVaihda_MouseEnter(object sender, MouseEventArgs e) btnVaihda_MouseLeave(object sender, MouseEventArgs e)

Page_Harjoitukset
List<Style> _listStyles
Page_Harjoitukset(List<Style> listStyles) OwnInitialize() Navigate_Click(object sender, RoutedEventArgs e) listSivut_SelectionChanged(object sender, SelectionChangedEventArgs e) ScrollViewer_MouseWheel(object sender, MouseWheelEventArgs e) class HarjoitusSivut

HarjoitusSivu1/-4/-5/-6/-9/-12/-15/-18
HarjoitusSivuNro(List<Style> listSyles)

HarjoitusSivu2/-7/-10/-13/-16
bool _bSanasto
HarjoitusSivuNro(List<Style> listSyles) btnPiilotaSanasto_Click(object sender, RoutedEventArgs e) btnSanasto_Click(object sender, RoutedEventArgs e)

HarjoitusSivu3/-8/-11/-14/-17
HarjoitusSivu17(List<Style> listStyles) btnTarkistus_Click(object sender, RoutedEventArgs e)