



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Jarno Nevala

Windows Phone 8

Sovelluskehityksen perusteet

Tietojenkäsittely
2013

TIIVISTELMÄ

Tekijä	Jarno Nevala
Opinnäytetyön nimi	Windows Phone 8 -sovelluskehityksen perusteet
Vuosi	2013
Kieli	suomi
Sivumäärä	35
Ohjaaja	Raija Tuomaala

Opinnäytetyön tarkoitus on tutkia Windows Phone 8 -sovelluskehitystyökaluja ja antaa perusteet sovelluskehittämisen aloittamiseen. Työn teoriaosuus kertoo Windows Phone 8 -alustan ominaisuuksista ja laitevaatimuksista sovelluskehittäjän näkökulmasta. Aineistona käytetään Windows Phone Dev Centerin materiaalia. Työssä kerrotaan sovelluksen julkaisuprosessista ja Windows Phone SDK 8.0 -työkaluista, sekä esitetään niiden asennus. Lopuksi esitetään uuden sovellusprojektin aloittaminen ja esimerkkisovellus.

Microsoft tarjoaa kattavat ja hyvät työkalut Windows Phone -ohjelmointiin. Ohjelmien käyttöliittymän rakentaminen on tehty helpoksi ja yksinkertaiseksi lisäämällä XAML-elementtejä raahaa-ja-pudota -menetelmällä. Windows Phone Dev Center tarjoaa paljon hyviä artikkeleja ja esimerkkikoodeja.

ABSTRACT

Author	Jarno Nevala
Title	The Development Basics of Windows Phone 8
Year	2013
Language	Finnish
Pages	35
Name of Supervisor	Raija Tuomaala

The target of this thesis was to give an overview of the Windows Phone 8 development tools and to produce introduction material for beginners. The theoretical part of the thesis introduced Windows Phone 8 platform features and system requirements from the application developer's point of view. The material that has been used is from Windows Phone Dev Center. The thesis described the process of publication and introduced Windows Phone SDK 8.0 tools, as well as showed the installation process. Finally, a new application project initiation and an example application were demonstrated.

Microsoft provides comprehensive and useful tools for programming Windows Phone -developments. The construction of the user interface has been made easy and simple by adding XAML elements using the drag-and-drop method. Windows Phone Dev Center offers a lot of good articles and code samples.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KÄSITTEET JA LYHENTEET	4
1 JOHDANTO.....	5
2 WINDOWS PHONE -KÄYTTÖJÄRJESTELMÄ	6
2.1 Windows Phone -käyttöliittymän ominaisuudet.....	7
2.1.1 Modern UI.....	7
2.1.2 Näyttö.....	8
2.1.3 Kartat.....	9
2.1.4 Kamera	9
2.1.5 Pilvipalvelut ja Live Connect.....	9
2.1.6 NFC - lähitunnistustekniikka	10
2.2 Käyttöjärjestelmän ydin	10
2.2.1 Shared Core	10
2.2.2 Native Code Support	11
2.3 Laitevaatimukset	11
3 SOVELLUSTEN JULKAISU WINDOWS PHONE STOREEN.....	12
3.1 Rekisteröityminen sovelluskehittäjäksi	13
3.2 Julkaisuprosessi.....	15
4 SOVELLUSKEHITYS.....	17
4.1 C# ja XAML -ohjelmointikielet.....	18
4.1.1 C#.....	18
4.1.2 XAML	18
4.2 Windows Phone -sovelluksen elinkaari	18
4.3 Windows Phone SDK 8.0 -työkalut.....	20
4.3.1 Visual Studio 2012 Express for Windows Phone	20
4.3.2 Microsoft Blend 2012 Express for Windows Phone.....	20
4.3.3 Windows Phone Emulator.....	20
4.3.4 Windows Phone Developer Registration -työkalu.....	21
4.3.5 Windows Phone Store Test Kit.....	22

4.4 Työkalujen asennus.....	22
4.5 Uuden Windows Phone -projektin aloittaminen ja sovelluksen osat.....	24
4.6 Esimerkkisovellus	26
4.6.1 Sovelluksen C#-koodi	28
5 YHTEENVETO	32
LÄHTEET.....	34

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1.	Windows Phone 8:n interaktiiviset tiilet ja päänäköymä	s.	6
Kuvio 2.	Windows Phone Store	s.	11
Kuvio 3.	Rekisteröityminen Windows Phone -sovelluskehittäjäksi	s.	13
Kuvio 4.	Sovelluksen elinkaari	s.	18
Kuvio 5.	Windows Phone SDK 8.0 -asennusmanageri	s.	22
Kuvio 6.	Visual Studio MainPage ja Solution Explorer	s.	24
Kuvio 7.	Kartat-sovellus ajettuna emulaattorilla	s.	25
Kuvio 8.	Kartat-sovelluksen XAML-koodi	s.	26

KÄSITTEET JA LYHENTEET

.Net Framework	Microsoftin valmistama ohjelmistokomponenttikirjasto, jota VisualStudio.NET-ympäristössä kehitetyt ohjelmistot käyttävät.
Direct-X	Microsoftin Windows-käyttöjärjestelmälle tarkoitettu ohjelmointirajapinta tietokoneohjelman ja laitteiston välille. Se tarjoaa yhtenäisen rajapinnan mm. 3D-grafiikkaa, ääntä ja ohjauslaitteita varten.
Hyper-V	Microsoftin kehittämä virtualisointitekniikka, joka on yhdistelmä ohjelmisto- ja laitteistotekniikkaa. Ominaisuus luo virtuaalikoneita yksittäiseen koneeseen siten, että se toimii niin kuin se olisi monta konetta.
NFC	Near Field Communication on lähitunnistustekniikka, jota voidaan käyttää puhelimen yhdistämiseen muihin laitteisiin ja informaation jakamiseen näiden välillä.
XAP	Silverlight Application Package on asennuspaketti Windows Phone –käyttöjärjestelmälle. Asennuspaketti lähetetään Windows Phone Storeen julkaistaessa sovellusta tai se voidaan asentaa muistikortilta testausvaiheessa.

Lyhenteet

API	Application Programming Interface, ohjelmointirajapinta
HTML	Hypertext Markup Language, hypertekstin merkintäkieli
NTFS	New Technology File System, tiedostojärjestelmä
UI	User Interface, käyttöliittymä
XAML	Extensible Application Markup Language, metamerkintäkieli

1 JOHDANTO

Tämä opinnäytetyö on johdanto Windows Phone 8 -alustaan ja sovelluskehitykseen. Työn tilaaja on Vaasan ammattikorkeakoulu. Sisällössä käydään läpi Windows Phonen historiaa sekä käyttöjärjestelmän ominaisuuksia ja laitevaatimuksia sovelluskehittäjälle tärkeistä näkökulmista. Tämän jälkeen esitetään sovelluskehittäjätilin luomista ja kerrotaan sovelluksen julkaisuprosessista sekä siihen liittyvistä vaatimuksista Windows Phone Storeen. Työssä ei julkaista sovellusta.

Sovelluksia Windows Phone 8:lle voidaan julkaista monella eri ohjelmointikielellä, mutta työssä käsitellään vain C# ja XAML -ohjelmointikieliä. Näistä ohjelmointikielistä ei kerrota syvällisemmin, koska opinnäytetyötä hyödyksi käytettävältä odotetaan entuudestaan kokemusta C#-ohjelmoinnista.

Ohjelmointikielten jälkeen kerrotaan hieman sovelluksen elinkaaresta, jonka jälkeen Windows Phone -sovelluskehitystyökalut esitellään ja asennetaan. Lopuksi esitetään uuden sovellusprojektin luominen Visual Studiossa sekä selitetään sovelluksen tiedostojen tarkoitukset. Työn lopussa esitetään esimerkisovellus koodeineen.

2 WINDOWS PHONE -KÄYTTÖJÄRJESTELMÄ

Windows Phone -käyttöjärjestelmän kehitys alkoi 2008, kun Microsoft uudisti mobiilistrategiaansa. Microsoft luopui aiemmasta kömpelöksi osoittautuneesta Windows Mobile -käyttöjärjestelmästä. Uusi käyttöjärjestelmä aloitettiin puhtaalta pöydältä ja Microsoft päätti katkaista käyttöjärjestelmien sovellusten yhteensopivuuden keskenään. Vanha ulkonäöllisesti ja ohjelmarajapintaisesti pöytäkoneiden Windows-käyttöjärjestelmiä muistuttava Windows Mobile haluttiin korvata käyttäjäystävällisellä käyttöjärjestelmällä, joka soveltuu paremmin moderniin puhelimeen huomioiden mm. kosketusnäytön ja sosiaalinen median. (Kolakowski 2010)

Ensimmäinen versio Windows Phone 7 julkaistiin lokakuussa 2010, ja se on saanut muutamia päivityksiä. Toukokuussa 2011 julkaistiin Windows Phone 7.5 (Mango), jonka merkittävimpiä uudistuksia olivat Internet Explorer 9 -mobiiliversio ja Windows Live SkyDrive -tuki. 2012 julkaistu Tango -päivitys tehtiin korjaamaan ohjelmistovikoja ja laskemaan laitevaatimuksia vähemmän tehokkaille laitteille.

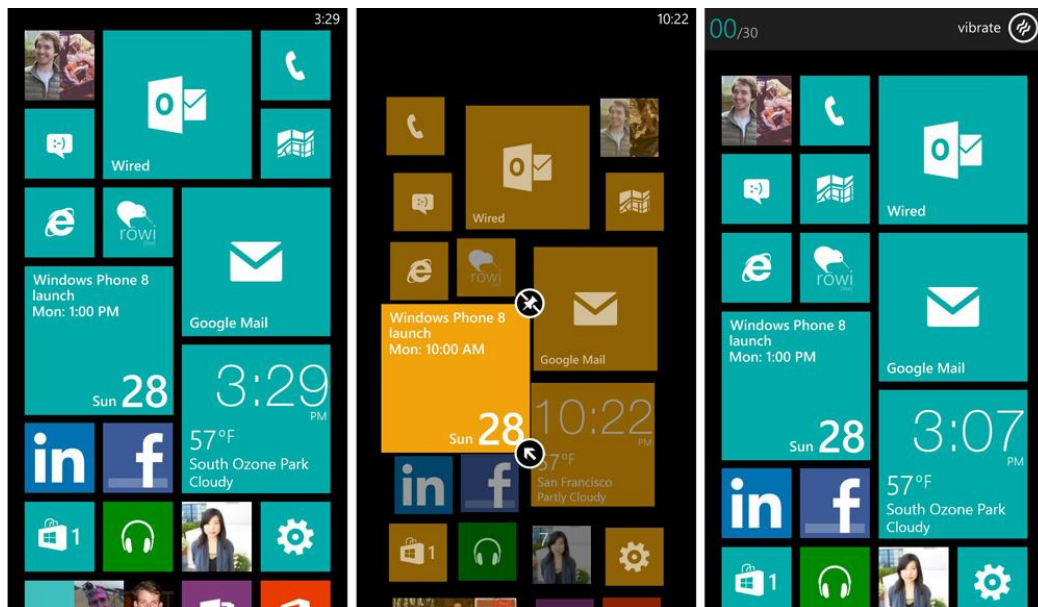
Viimeisin Windows Phone 7.8 -versio julkaistiin samoihin aikoihin yhdessä Windows Phone 8 kanssa. Päivitys lainasi ominaisuuksia Windows Phone 8 -versiosta, kuten päivitetty aloitusnäyttö. Päivityksen tarkoitus on pidentää Windows Phone 7 puhelinten elinikää, koska nämä laitteet eivät täytä Windows Phone 8:n laitevaatimuksia. Windows Phone 8 julkaistiin lokakuussa 2012, ja se on viimeisin Windows Phone -versio. (Bhushan 2013)

2.1 Windows Phone -käyttöliittymän ominaisuudet

2.1.1 Modern UI

Ennen Metro UI:na tunnettu, Modern UI on design-kieli, jota Microsoft käyttää nykyään tuotteissaan. Alunperin Windows Phonelle suunniteltu Modern UI:n tyylin periaate on korostaa tekstiä ja informaatiota ennen kuvia. Microsoft kuvailee tyylin olevan "tasainen, nopea, moderni" ja "uudistunut" verraten kuvakepohjaisiin käyttöliittymiin Windowsissa, Androidissa ja iOS:ssä. Animaatiot ovat suuressa osassa navigoinnissa ja Live Tile:ssä. (Bhushan 2013)

Käyttöliittymän tyyliissä korostuu tiilimäinen rakenne. Nämä ns. Live Tile:t näyttävät interaktiivista tietoa Windows Phonen päänäkymässä ja niiden näpäyttäminen käynnistää "tilen" sovelluksen. Kuten kuvassa 1 nähdään käyttäjä voi valita, mitkä sovellukset näkyvät aloitusnäkymässä sekä järjestellä ja muuttaa tiilejä kolmeen eri kokoon. Järjestelmän asetuksista on mahdollisuus vaihtaa tiilien aksenttiväri haluamukseen.



Kuva 1. Windows Phone 8:n interaktiiviset tiilet ja päänäkymän muokkaaminen

Järjestelmä lataa pilvipalveluista informaatiota muistiin ja päivittää ne tiileihin. Lukitusnäyttö tukee kolmannen osapuolen sovellusten ilmoituksia, jonka teksti,

kuvake ja luvut haetaan kyseisen sovelluksen tiilestä. Käyttäjä voi valita mitkä sovellukset näkyvät ilmoitusalueella. (Microsoft 2013 A)

2.1.2 Näyttö

Puhelimessa on kiihtyvyyssanturi, joka mittaa missä asennossa puhelinta pidetään. Järjestelmä säätää näytön kuvan vastaamaan puhelimen asentoa. Sovelluskehittäjällä on mahdollisuus suunnitella sovellus pysty- sekä vaakasuuntaiseksi. Käyttäjällä on myös mahdollisuus lukita näytön asento.

Windows Phone tukee kolmea näytön tarkkuutta:

Nimi	Resoluutio	Kuvasuhde
• WVGA	800 x 480	15:9
• WXGA	1280 x 768	15:9
• 720p	1280 x 720	16:9

Yksi Windows Phonen vahvuuksia on sovellusten käyttöliittymän automaattinen skaalaus. Sovelluskehittäjän ei tarvitse murehtia sovelluksen käyttöliittymän muokkaamisesta eri näytön tarkkuuksille. Käytettäessä vektorigrafiikkaa (esim. XAML tai HTML) järjestelmä automaattisesti skaalaa käyttöliittymän ottaen jokaisen pikselin käyttöön välttääkseen pikselöitymisen. Tämä mahdollistaa sovellusten toimivuuden laitteiden näytöillä, joita ei ole vielä valmistettu. Tätä ominaisuutta hyödyntäessä on suositeltavaa käyttää kuvia, joissa on suuri resoluutio. (Greenfield 2013)

2.1.3 Kartat

Nokian kartat korvaavat aiemmat Bing-kartat Windows Phone 8:ssa. Bing Maps API:a on silti mahdollisuus vielä hyödyntää, ja vanhemmat tätä käyttävät sovellukset jatkavat toimintaansa. Nokia Maps tarjoaa viimeistellympää ja täsmällisempää karttadataa sekä uuden 3D-tilan ja laitteistokiihdytteisen renderöinnin sulavampaan suoritukseen. Uusi WinPRT -pohjainen sovellusrajapinta paikannukselle tuo Nokia Maps -kontrollit Windows Phone -alustalle ja tarjoaa uuden toiminnon, joka generoi karttojen ja navigoinnin tapaisia ominaisuuksia kolmannen osapuolen sovellukseen käyttäen API-kutsua.

2.1.4 Kamera

Windows Phone 8 tarjoaa kaksi kamera API:a joita sovelluskehittäjä voi hyödyntää. Ensimmäinen PhotoCamera API sisältää perusominaisuudet kamerasovelluksen käynnistämiseen ja kuvaamiseen sovelluksessa sekä kontrollit kuvan resoluutioon, salama- ja tarkennusasetuksiin. PhotoCaptureDevice API antaa edistyksellisemmät ominaisuudet kuten:

- ISO-arvon ja valotuksen asetukset
- Manuaalinen tarkennus
- Multiframe-kuvaus
- Linssit

Linssit voivat olla erilaisia efektejä, filttäreitä tai älykkäitä sovelluksia esim. viivakoodinlukija. Kamerapohjaiset sovellukset on integroitu entistä paremmin Windows Phone 8:n kamerasovellukseen. Esimerkiksi linssisovellukset on sijoitettu kuvaustilassa oman pikavalikon taakse. (Windows Phone Network 2012)

2.1.5 Pilvipalvelut ja Live Connect

Windows Phone käyttää Microsoftin Live-tunnuksia välittääkseen informaatiota pilvipalvelun ja eri sovellusten välillä. Järjestelmä sisältää integroidun Skydrive -pilvipalvelun, johon puhelimen valokuvat, dokumentit ja kansiot päivittyvät automaattisesti. Muita tätä yhteyttä käyttäviä sovelluksia ovat Hotmail-

sähköposti, Office, Skype, Kalenteri ja Ihmiset (kontaktit). Käyttäjällä on mahdollisuus vaikuttaa Live-yhteyttä käyttävien sovellusten yhteysasetuksiin. Live Connect on kokoelma ohjelmointirajapintoja ja kontroleja jotka mahdollistavat sovelluskehittäjiä samaan yhteyden Live-palveluiden ytimeen.

2.1.6 NFC - lähitunnistustekniikka

Windows Phone tukee Near Field Communication -tekniikkaa, jota voidaan käyttää eri tarkoituksiin kuten: Puhelimen yhdistäminen muihin laitteisiin, informaation jakoon NFC-älyjulistaiden kanssa sekä eri maksupäätteiden välillä Kukkaro-sovelluksella.

2.2 Käyttöjärjestelmän ydin

Saman käyttöjärjestelmäytimen jakaminen (Shared Core) tarkoittaa, että sovelluskehittäjät voivat tehdä helposti sovelluksia, jotka toimivat pöytäkoneissa, tableteissa sekä Windows Phonessa. Aiemmin käytetty Windows CE -pohjainen ydin korvataan Windows Phone 8:ssa Windows NT -käyttöjärjestelmäytimellä, joka on sama kuin Windows 8 -käyttöjärjestelmässä. Käyttäjä voi siis nähdä samoja sovelluksia eri alustoilla. Vaihdos on vaikuttanut laitteistovaatimuksiin ja tuo mukanaan uusia hyviä ominaisuuksia. (Belfiore 2012)

2.2.1 Shared Core

Jaettu ydin tarkoittaa:

- Käyttöjärjestelmäkomponentit kuten kernel, verkko, grafiikka (Direct-X), tiedostojärjestelmä (NTFS) ja multimedia ovat samoja Windows 8- ja Windows Phone 8 –alustojen välillä.
- Laitekehittäjät työskentelevät samalla ajurimallilla molemmissa alustoissa.
- Parempi tietoturva
- Moniydintuki ja parempi sovellusten tausta-ajo.
- Yhtenäinen ydin helpottaa Windows Phone -alustaa mukautumaan tulevaisuudessa. (Belfiore 2012)

2.2.2 Native Code Support

Windows Phone 8 sisältää täyden tuen C ja C++ -ohjelmointikielille, joka helpottaa sovellusten kehittämistä useammille alustoille. Tämä tarkoittaa myös tukea suosituille pelimoottoreille kuten Havok Vision Engine, Autodesk Scaleform, Audiokinetic Wwise, Unity, kuten myös DirectX-pohjaiselle pelikehitykselle. (Belfiore 2012)

Windows Phonen sovellustarjontaa on pidetty niukkana verrattaen iOS:n ja Androidiin. Windows Phone 8:n ja sen sisältämän Native Code Supportin tultua sovellustarjonta on noussut merkittävästi. iOS ja Android -sovelluskehittäjillä on nyt paremmat mahdollisuudet kääntää sovelluksiaan Windows Phonelle.

2.3 Laitevaatimukset

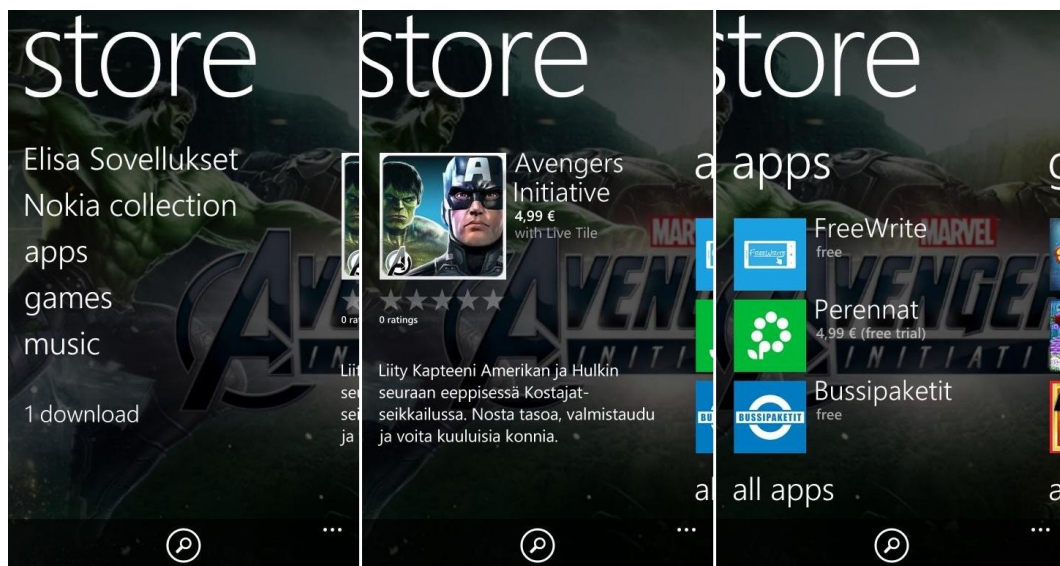
Ennen mobiilisovelluskehitys oli haastavampaa laitteiden komponenttien välisten eroavaisuuksien takia. Usein jouduttiin muokkaamaan sovellus samaakin käyttöjärjestelmää käyttävien laitteiden välillä kullekin laitteelle sopivaksi.

Windows Phone 8 -laitteille on asetettu tarkat laitevaatimukset, joka helpottaa sovelluskehitystä. Ulkoisesti jokaisessa laitteessa tulee olla takaisin, "Start", etsi, kamera, virta, volume ylös ja alas napit. Laitevaatimukset ovat:

- Qualcomm Snapdragon S4 dual-core tai Snapdragon 800 -suoritin
- Vähintään 512 MB RAM-muistia WVGA-puhelimissa ja 1 GB 720p / WXGA / 1080p - tukeville puhelimille
- DirectX -näytönohjain
- MultiTouch -kosketusnäyttö neljän pisteen monipistetunnistuksella
- Vähintään 4 GB flashmuistia
- Micro-USB 2.0 sekä 3.5mm-stereoliitäntä kolmen napin tuella
- Vähintään VGA-tason kamera Led tai Xenon salamalla
- GPS, kiihtyvyyssanturi, valotunnistin ja värinämoottori
- WLAN (802.11b/g) ja Bluetooth. (WMPoweruser 2012)

3 SOVELLUSTEN JULKAISU WINDOWS PHONE STOREEN

Nykyään jokaisesta älypuhelimesta löytyy käyttöjärjestelmäänsä integroitu sovelluspalvelu, joka antaa mahdollisuuden selata ja ladata kolmansien osapuolten sovelluksia. Microsoftin Windows Phone Store -palvelun sovelluksia pääsee selaamaan puhelimella tai kirjautumalla tietokoneen selaimella osoitteeseen www.windowsphone.com, josta on mahdollisuus lähettää asennuskäskeä puhelimeen.



Kuva 2. Windows Phone Store

Windows Phone -sovelluskehittäjäksi voi ryhtyä rekisteröitynyt yksityishenkilö tai yritys, mutta kauppaan asti pääsevien sovellusten tulee ensin täyttää tiettyjä sertifiointiprosesseja, joilla sovellus testaan luotettavaksi ja käyttöjärjestelmän asettamien vaatimusten täyttäneeksi. Windows Phone pystyy ajamaan vain sovelluksia, jotka on jaettu Windows Phone Storen tai Windows Phone Dev Centerin kautta.

3.1 Rekisteröityminen sovelluskehittäjäksi

Ennen sovellusten julkistamista Windows Phone Storeen tulee ensin hankkia sovelluskehittäjätili. Tällä käytännöllä varmistetaan sovelluskehittäjän oikea henkilöllisyys ja vältetään haittaohjelmien leviäminen. Henkilöllisyyden varmistamiseksi tulee omistaa Microsoft-tili sekä antaa luottokorttitiedot, joilla myös liittymismaksu peritään.

Rekisteröityminen aloitetaan Windows Phone Dev Centerissä, joka on portaali sovelluskehittäjille, missä tarjotaan tietoa kuinka luoda, hallita ja julkaista sovelluksia. Osoitteessa: <https://dev.windowsphone.com/en-us/dashboard> ohjataan kirjautumaan Microsoft-tilillä tai luomaan sellainen, jos ei ennestään omista. Kirjautuminen ohjaa takaisin Windows Phone Dev Centerin sivuille, jossa ensimmäisenä pyydetään hyväksymään Windows Phone Store -sovelluskehittäjän sopimus. Sivut ja niillä esiintyvät käyttöehdot ovat englanniksi. Sopimus hyväksytään klikkaamalla "Accept and continue". Tämän jälkeen pyydetään Microsoft-tilin varmistuskoodi, jolla varmistetaan tili omaksi.

Varmistuksen jälkeen ohjataan itse tilinluontiprosessiin, jonka ensimmäisessä kohdassa valitaan kotimaa sekä tilin tyyppi. Tilityyppejä ovat Individual tai Company. Yritystyyppi maksaa 75 euroa/vuosi ja tarjoaa lisäominaisuuksia verraten yksityishenkilötyyppiin. Yksityishenkilöille suunnattu tilityyppi maksaa 14 euroa/vuosi. Yksityishenkilötyypin tunnukset Microsoft tarjoaa korkeakouluopiskelijoille ilmaiseksi, josta myöhemmin lisää. Tilityypin valinnan jälkeen siirrytään Account info -kohtaan, jossa täytetään kehittäjän perustiedot sekä sovelluskehittäjän julkaisijanimi, joka näkyy kaupassa.

Seuraavassa vaiheessa pyydetään taas hyväksymään Windows Phone Store -sovelluskehittäjän sopimus, jonka hyväksymisen jälkeen siirrytään vaiheeseen, jossa ilmoitetaan rekisteröitymisen hinta. Tässä vaiheessa on mahdollisuus antaa rekisteröitymiskoodi, jolla hinnan saa muutettua ilmaiseksi. Koodin saamiseksi korkeakouluopiskelijan tulee tietää noutaa koodi erilliseltä sivulta. Tätä mahdollisuutta ei kyseissä kohdassa kerrota eikä linkkiä tämän koodin noutamiseen anneta.

Opiskelijan tulee hankkia koodi Microsoft DreamSpark -sivuilta osoitteesta: <https://www.dreamspark.com/Student/Windows-Phone-8-App-Development.aspx> tai navigoimalla aloitussivun alhaalta kohdasta "start developing for Windows Phone 8". Sivun lopussa kohdassa "Publish your app" löytyy korostettu linkki "Get your registration code". Tämä ohjaa sivuille, jossa luodaan DreamSpark -tunnukset ja tunnistaudutaan opiskelijaksi koulun sähköpostin avulla.

Tunnistautumisen jälkeen saadaan koodi ja palataan takaisin rekisteröitymisprosessin hintavaiheeseen. Saatu koodi syötetään sille tarkoitettuun registration code -kenttään ja klikataan sen alla "update total". Tämän jälkeen hinnan pitäisi päivittyä ilmaiseksi. Seuraavaksi syötetään luottokorttitiedot, jonka jälkeen päästään rekisteröitymisen yhteenvedoon.

The screenshot shows the 'Purchase' page in the Windows Dev Center. At the top, there's a navigation bar with 'Windows | Dev Center - Windows Store apps' and a search bar. Below this is a breadcrumb trail: 'Account type > Account info > Agreement > Price > Payment > Purchase > Thank you'. The main heading is 'Purchase'. The page is divided into two columns. The left column contains 'Account information' with details for 'Jarno Nevala', including email and phone number. The right column shows pricing: 'Your registration price' at 14.00 EUR, 'Applicable taxes and fees' at 0.00 EUR, and 'Your total' at 0.00 EUR. Below the pricing, there's a section for 'Publisher display name' (Jarno Nevala) and a 'Confirm your order' section with a note to 'Click Purchase to finish.' At the bottom, there are two buttons: 'Back' and 'Purchase'.

Purchase	
Account information	Your registration price
Jarno Nevala	14.00 EUR
Osoite...	Applicable taxes and fees
Puhelin	0.00 EUR
Sähköposti	Your total
	0.00 EUR
Publisher display name	
Jarno Nevala	
Confirm your order	
Click Purchase to finish.	
Back	Purchase

Kuva 3. Rekisteröityminen Windows Phone -sovelluskehittäjäksi

Yhteenvedon vahvistamisen jälkeen ohjataan takaisin Windows Phone Dev Centerin sivuille ja tili on valmis käytettäväksi. Muiden kuin opiskelijoiden tarvitsee vielä vahvistaa luottokortti, jolla suoritetaan rekisteröityminen. Jos tarkoituksena on tehdä maksullisia sovelluksia, niin kehittäjän tulee vielä antaa

tili, johon sovelluksista ansaitut rahat suoritetaan sekä määritellä veroprofiili. Nämä asetukset ovat myöhemmin muutettavissa.

3.2 Julkaisuprosessi

Ennen sovelluksen julkistamista Windows Phone Storeen tulee sen ensin läpäistä Microsoftin määrittämä sertifiointiprosessi. Prosessilla testataan sovelluksen laatua eri osa-alueilla, kuten sovelluksen toiminnallisuus, sisältö, tietoturva ja tekniset vaatimukset. Esimerkki kriteerinä sovelluksen toiminnallisuuteen on takaisin-napin toiminnallisuus siten, että nappia painamalla sovelluksessa navigoidaan edelliselle sivulle ja sovelluksen pääsivulta ohjelman tulee sulkeutua. Jokaisessa sovelluksessa, jossa käytetään sijaintitietoja, tulee sovelluksen asetuksissa olla käyttäjällä mahdollisuus kytkeä tämä toiminto pois. Sertifiointin asettamat vaatimukset ovat luettavissa Windows Phone Dev Centerissä osoitteessa:

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184843%28v=vs.105%29.aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184843%28v%3Dvs.105%29.aspx)

Ennen sovelluksen lähettämistä kannattaa tarkistaa, että sovellus täyttää vaatimukset, ja testata sovellusta kehitystyökalujen mukana tulevalla Windows Phone 8 -emulaattorilla. Sovelluksesta on mahdollista julkistaa testiversio Dev Centerissä ennen varsinaista Storeen julkaisua. Tämä niin sanottu beta testing -prosessi menee automatisoidun ja nopean sertifiointiprosessin läpi ja antaa latauslinkin, jonka voi lähettää testaajille. Beta-versio on voimassa 90 päivää. (Microsoft 2013 B)

Sovelluksen julkaiseminen aloitetaan Dev Centerin etusivulta löytyvällä Submit app -linkillä. Julkaisu tapahtuu vaiheittain, joissa annetaan sovellukseen liittyvät tiedot, kuten sovelluksen nimi, kuvaus ja myyntitiedot. Prosessin aikana tulee ladata kehitystyökaluilla valmistetut sovelluksen XAP-asennuspaketti, tile-kuvakkeet sekä kuvakaappaukset Storeen. Julkaisuprosessin lopussa sovellus lähetetään Microsoftille testiin, joka kestää noin viikon. Sovellus julkaistaan testin jälkeen Storessa tai sille määritettynä päivämääränä.

Sertifioinnin läpäisemättömästä sovelluksesta lähetetään raportti, josta selviää puutteet sekä annetaan ohjeita niiden korjaamiseen. Korjausten jälkeen julkaisuprosessi toistetaan. (Microsoft 2013 C)

Sovelluksia on myös mahdollisuus julkaista Dev Centerin välityksellä kohdistetusti siten, että ne ovat piilossa Windows Phone Storessa. Targeted app distribution -prosessi toimii kuten betatestaus, jossa latauslinkin voi antaa valikoiduille henkilöille käyttöön. Yritykset voivat myös hyödyntää tätä ominaisuutta siten, että vain yrityksen työntekijöillä on oikeudet asentaa sovellus puhelimiinsa. (Microsoft 2013 D)

4 SOVELLUSKEHITYS

Aikaisempi kokemus .NET ja XAML -ohjelmoinnista on eduksi aloittaessa Windows Phone -sovellusten kehittämistä. XAML:ia käytetään sovellusten käyttöliittymän rakentamisessa ja C# tai Visual Basic -ohjelmointikieltä sovelluksen koodissa. Pelien kehittämiseen on käytössä Microsoftin DirectX-ohjelmointirajapinta ja C++ -ohjelmointikieli.

Microsoft tarjoaa ilmaiset työkalut Windows Phone -sovelluskehitykseen. Windows Phone Software Development Kit (SDK) 8.0 -paketti on ladattavissa Windows Phone Dev Centeristä. Paketti sisältää karsitun Visual Studio Express 2012 -version, jos koneella ei ennestään ole Visual Studio -kehitystyökaluja. Windows Phone SDK 8.0 -komponentit ovat seuraavat:

- Visual Studio 2012 Express for Windows Phone
- Microsoft Blend 2012 Express for Windows Phone
- Projektimallit ja sovelluskirjastot (API) sovelluskehitykseen
- Windows Phone 8.0 ja 7.1 emulaattorit
- Windows Phone Developer Registration -työkalu
- Windows Phone Store Test Kit

Microsoftin uusin versio Visual Studio 2013 sisältää sisäänrakennetun tuen Windows Phone -sovelluskehitykseen, mutta vain Windows Phone 8 -alustalle. Jos haluaa tehdä sovelluksia vanhemmalle Windows Phone 7 -alustalle, tulee tällöin käyttää Visual Studio 2012 -versiota. Opinnäytetyössäni käytän ilmaisia Windows Phone (SDK) 8.0:n tarjoamia työkaluja.

Tässä luvussa kerron C# ja XAML-ohjelmointikielistä sekä sovelluksen elinkaaresta, jonka jälkeen esittelen kehitystyökalut ja niiden asennuksen. Lisäksi kerron tyypillisen Windows Phone -sovelluksen aloittamisesta ja siihen liittyvistä osista ja ominaisuuksista. Lopuksi esittelen esimerkisovelluksen.

4.1 C# ja XAML -ohjelmointikielet

4.1.1 C#

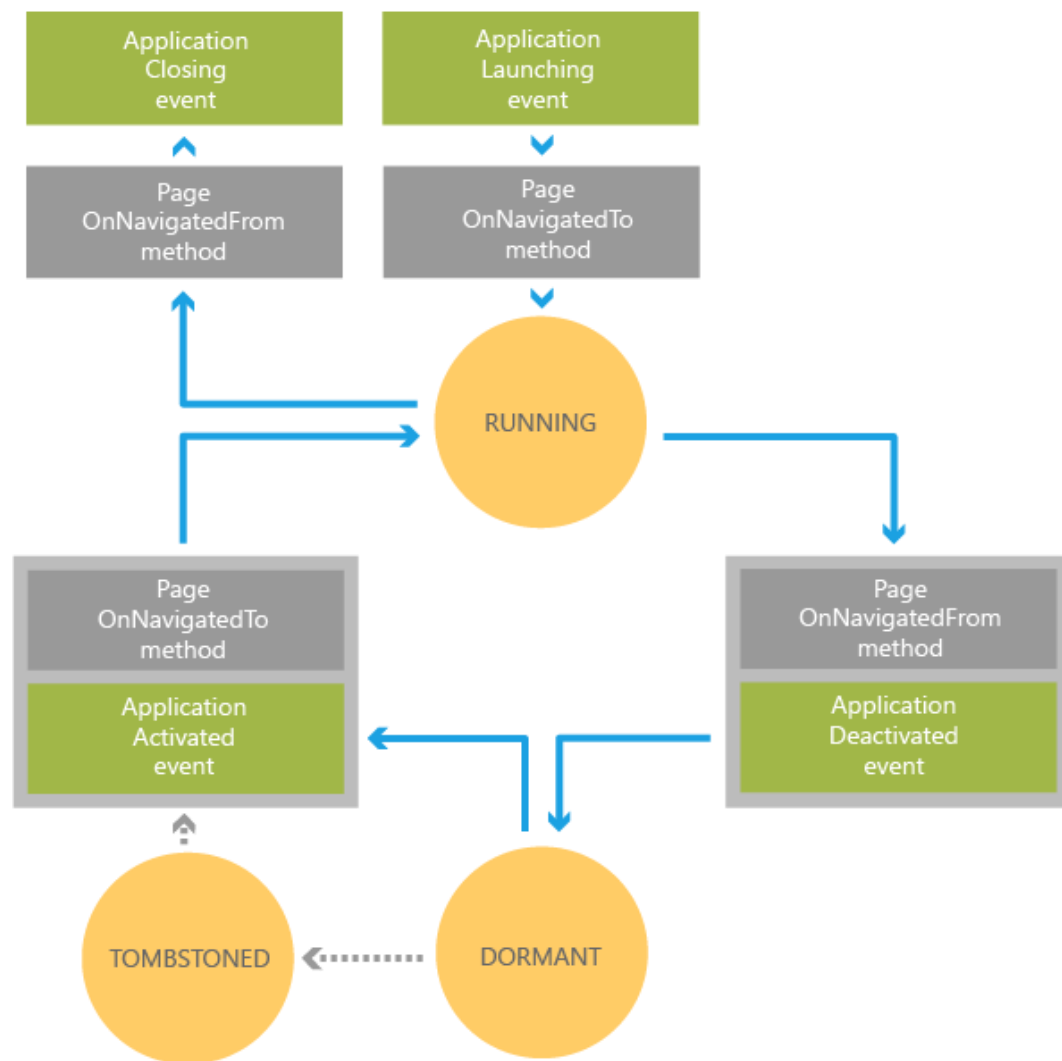
C# on yksinkertainen, moderni, olio-pohjainen ohjelmointikieli, joka julkaistiin vuonna 2000. Se käyttää Microsoftin luomaa .NET Framework -ohjelmistokomponenttikirjastoa. C#-kehittämiseen vaikuttivat muut olio-pohjaiset kielet kuten C++ ja Java. Se yhdistää Javan ja MS Visual Basicin helppokäyttöisyyden sekä C++:n tehokkuuden. Nykään C# on yksi suosituimmista ohjelmointikielistä. Viimeisin versio ja Windows Phone 8:n tukema C# 5.0 julkaistiin elokuussa 2012. (Microsoft 2013 E)

4.1.2 XAML

XAML eli Extensible Application Markup Language on Microsoftin XML-pohjainen merkintäkieli. Sitä käytetään käyttöliittymien graafisen puolen toteuttamiseen .NET Framework -sovelluksissa. XAML muistuttaa paljolti HTML -merkintäkieltä, joka löytyy nettisivun lähdekoodista. XAML on tärkeä osa Windows Phone sovelluskehitystä. XAML -pohjaisia käyttöliittymiä voi rakentaa Windows Phone SDK 8:n mukana tulevalla Microsoft Blendillä. (Microsoft 2013 F)

4.2 Windows Phone -sovelluksen elinkaari

Windows Phonessa vain yksin sovellus on käynnissä kerrallaan. Tämä varmistaa, että sovellus, jota parhaillaan käytetään, toimii sulavasti ja reagoi nopeasti. Kun käyttäjä navigoi pois sovelluksesta, se joko pysäytetään tai suljetaan riippuen tavasta, miten ohjelmasta on poistuttu. Windows Phone tarjoaa erilaisia tapahtumia ja ohjelmointirajapintoja, joilla sovelluskehittäjä voi hallita sovelluksen aktivoitumista ja sulkeutumista. Kuvassa 5 esitetyssä sovelluksen elinkaareissa olevat pallot ovat sovelluksen tiloja, ja suorakulmiot esittävät joko sovellus- tai sivutason tapahtumia, joissa sovellus käsittelee tilojaan.



Kuva 4. Sovelluksen elinkaari. (Microsoft 2013 G)

Windows Phone sovelluksessa käyttäjän navigointi aiheuttaa tapahtumia, jotka johtavat sovelluksen eri tiloihin. Tiloja ovat ajaminen, dormant ja tombstoned. Tiloihin johtavia tapahtumia ovat käynnistys, deaktivointi, aktivointi ja sulkeminen. Dormant-tila eli eräänlainen horrostila kytkeytyy, kun käyttäjä navigoi pois sovelluksesta. Tässä tilassa sovelluksen prosessit ovat pysäytettynä, mutta sovelluksen tila pysyy muistissa aina siihen asti, kun sovellukseen palataan.

Jos uusia sovelluksia avataan sen jälkeen, kun sovellus on asettunut dormant-tilaan, tulee niille vapauttaa muistia. Tämä johtaa dormant-tilan vaihtumista

tombstoned-tilaan. Tombstoned-tilassa sovellus suljetaan, mutta käyttöjärjestelmä säilyttää sovelluksen navigointi-tilan ja deaktivointi-tapahtumankäsittelijän tallentamia tietoja. Käyttöjärjestelmä säilyttää tietoja viidelle sovellukselle. Kun käyttäjä palaa tombstone-tilaan asettuneeseen sovellukseen, niin sovellus uudelleen käynnistyy ja palauttaa tallennetun tilan. Tilan palautus kestää hieman pitempään, kuin dormant-tilasta palaaminen. (Microsoft 2013 G)

4.3 Windows Phone SDK 8.0 -työkalut

4.3.1 Visual Studio 2012 Express for Windows Phone

Visual Studio on Microsoftin ohjelmankehitysympäristö. Visual Studio 2012 on mahdollisuus tehdä sovelluksia Windows 8:lle, mutta koska Windows Phone 8 julkaistiin Windows 8:n jälkeen, tulee siihen ladata ja asentaa Windows Phone SDK 8.0. Paketti asentaa olemassa olevaan Visual Studio 2012 Professional, Premium tai Ultimate -versioon liitännäisinä Windows Phone -sovelluskehitykseen tarvittavat työkalut. Jos ei aiempaa Visual Studiota löydy, niin Windows Phone SDK 8.0 asentaa karsitun Visual Studio 2012 Express for Windows Phone -version.

4.3.2 Microsoft Blend 2012 Express for Windows Phone

Microsoft Blend on käyttöliittymäsuunnitteluun tarkoitettu työkalu. Blend tarjoaa kehittyneemmät työkalut käyttöliittymän visuaaliseen osan rakentamiseen. Visual Studiosta löytyy perustyökalut käyttöliittymän luontiin, mutta Blendillä on mahdollisuus tehdä tyylikkäämpiä ja animoituja käyttöliittymiä. Blend vapauttaa käyttöliittymäsuunnittelijan toteuttamaan sovelluksen ulkoasua samalla, kun sovelluskehittäjä työskentelee koodin parissa Visual Studiossa.

4.3.3 Windows Phone Emulator

Windows Phone Emulator on työpöytäsovellus, joka emuloi Windows Phone -laitetta. Emulaattori tekee mahdolliseksi ohjelman ajamisen ja testaamisen virtuaalisessa laitteessa ilman fyysistä laitetta, kumminkin tarjoten vastaavaa suorituskykyä. Yksi suurin emulaattorin tuoma etu on, että sovelluksen

prototyypppejä voi testata ilman pelkoa, että fyysinen testilaitte vahingoittuisi. Ennen julkaisemista sovellus kannattaa kuitenkin vielä testata emulaattorin lisäksi fyysisellä laitteella.

Sovellusta voi testata eri emulaattoreilla valiten käyttöjärjestelmäversion ja näytön resoluutioiden väliltä. Oletus emulaattori Visual Studiossa on Emulator WVGA 512MB, joka vastaa muistirajoitteista Windows Phone 8 -laitetta. Tämä oletus kannustaa sovelluskehittäjiä suuntaamaan sovelluksen mahdollisimman monelle Windows Phone -laitteelle.

Visual Studion Tools-valikosta löytyy Simulation Dashboard -työkalu, jonka tarkoituksena luoda odottamattomissa tilanteita sovellusta testatessa. Yleensä sovellus kehitetään ja testataan optimaalisissa olosuhteissa, jonka jälkeen se julkaistaan ja toivotaan, että se toimii odotetusti laitteissa. Tällä työkalulla on tarkoitus taata, että sovellus toimii odottamattomissa tilanteissa. Simulation Dashboardilla voi kontrolloida verkkoliikenteen nopeutta ja signaalin vahvuutta sekä lukitusnäyttöä ja huomautuksia.

4.3.4 Windows Phone Developer Registration -työkalu

Kuten aiemmin mainitsin, sovellus kannattaa testata myös oikealla puhelimella ennen julkaisua. Jotta sovelluksen saa asennettua ja ajettua puhelimessa, tulee se rekisteröidä sovelluskehitykseen. Tämän niin sanotun lukituksen poistamiseen tarvitaan sovelluskehittäjän tiliä ja Windows Phone SDK 8.0:n mukana tulevaa Windows Phone Developer Registration -työkalua.

Ohjatussa prosessissa puhelin käynnistetään ja liitetään USB-kaapelilla koneeseen sekä kirjaudutaan sovelluskehittäjän tunnuksilla ohjelmaan. Windows Phone 7.1 -puhelimien rekisteröimisessä tulee asentaa lisäksi Zune-ohjelma, jota käytetään tietokoneen ja puhelimen yhteyden hallintaan.

4.3.5 Windows Phone Store Test Kit

Windows Phone Store Test Kit käy läpi automaattisia ja manuaalisia testejä, jotka auttavat selvittämään tuleeko sovellus läpäisemään Storen sertifiointiprosessit. Testit voidaan ajaa kohdentaen Windows Phone 8 tai Windows Phone 7.1 -alustoille. Jos sovellus ei läpäise testejä, niin Store Test Kit antaa tietoja ja ohjeita sovelluksen korjaamiseen. (Microsoft 2013 H)

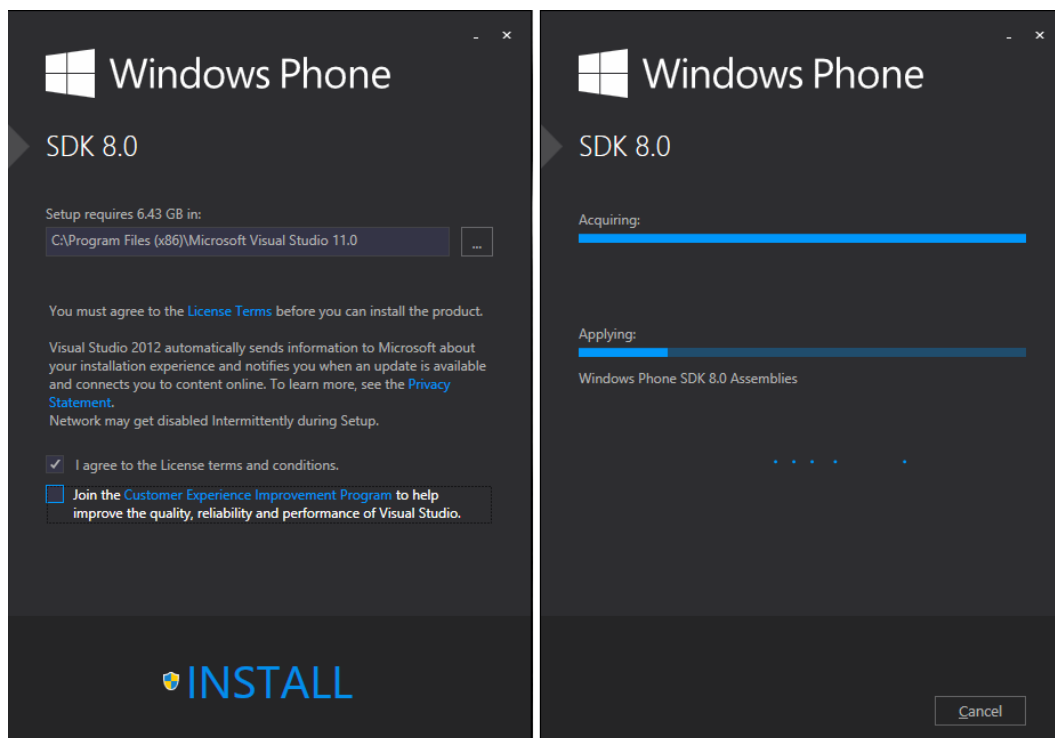
4.4 Työkalujen asennus

Windows Phone SDK 8.0 laitevaatimukset ovat seuraavat:

- Windows 8 64-bit -käyttöjärjestelmä
- 64-bittinen prosessori
- 6.5 GB vapaata kovalevytilaa
- 4 GB RAM-muistia

Käytettäessä Windows Phone SDK:n emulaattoria laitevaatimukset ovat tiukemmat, koska emulaattori toimii virtuaalikoneena Windows 8 Pro:n Hyper-V -ominaisuuden kautta. Toimiakseen emulaattori tarvitsee Windows 8 Professional x64 -version ja koneen prosessorin tulee tukea Second Level Address Translation (SLAT) -ominaisuutta.

Windows Phone SDK 8.0 -asennusmanageri on ladattavissa Windows Phone Dev Centeristä osoitteesta: <http://dev.windowsphone.com/en-us/downloadsdk>. Tiedosto on vain yhden megabitin kokoinen. Tiedoston ajaessa tulee kuvan 4 mukainen näkymä, jossa valitaan asennushakemisto ja vahvistetaan käyttöehdot.



Kuva 5. Windows Phone SDK 8.0 -asennusmanageri

Asennushakemiston ja käyttöehtojen jälkeen napautetaan "install", jonka jälkeen asennusmanageri alkaa ladata tiedostoja verkosta noin 1.6 GB edestä. Itse asennus kestää noin 20 minuuttia. Asennuksen valmistuttua voidaan Visual Studio käynnistää klikkaamalla "Launch". Asennus on mahdollista ladata myös ISO-levykuvatiedostona, josta sen voi polttaa DVD-levylle tai ajaa virtuaaliasemalla.

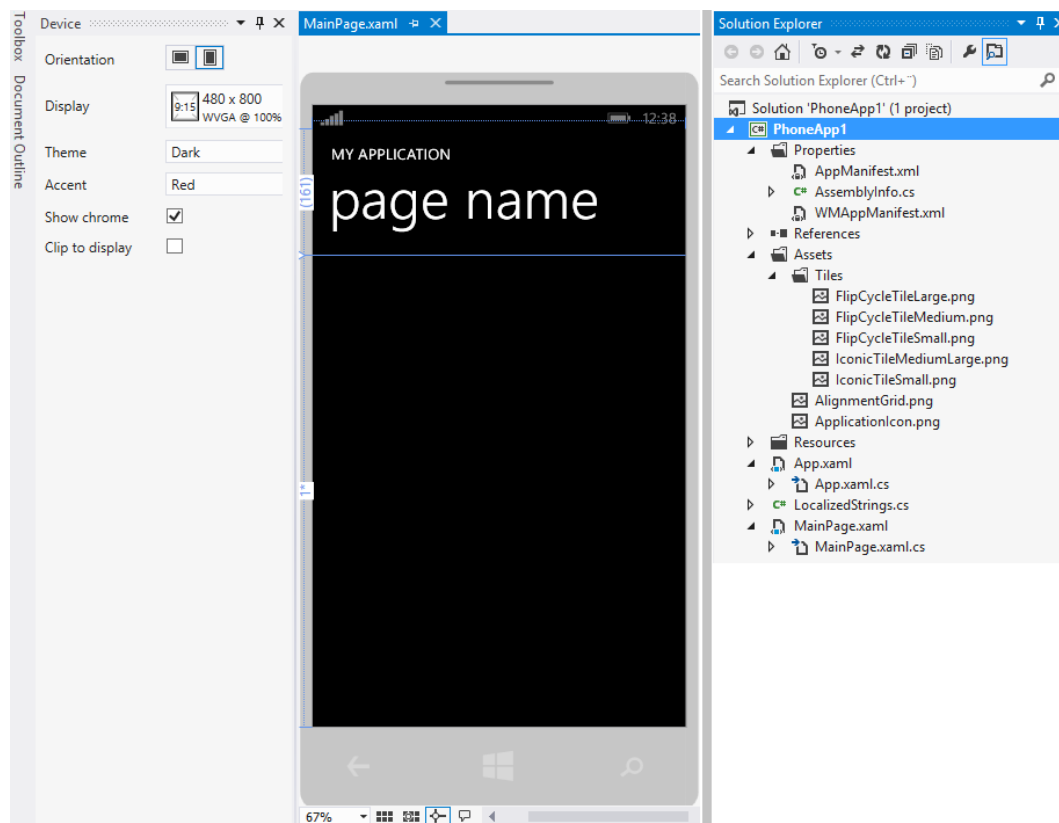
Työkalujen asennus oli helppoa, mutta minulla Windows Phone Emulatorin toimivuus vaati vielä pari vaihetta. Asennusmanagerin viimeisessä ruudussa annettiin tieto, että tietokoneeni prosessori tukee SLAT-ominaisuutta, mutta se tulee kytkeä päälle. Kytkeäkseen SLAT-tuen päälle minun täytyi käynnistää kone uudelleen ja edetä käynnistysnäkyvästä emolevyn laitteisto-ohjelmistoon (BIOS). SLAT-asetuksia ei löytynyt kyseisellä nimellä, mutta pienen selvittelyn jälkeen prosessorin asetuksista löytyi AMD:n vastaava tekniikka nimeltä Secure Virtual Machine (SVM). SLAT-ominaisuus esiintyy eri piirivalmistajilla omilla nimillään ja vanhemmista suorittimista se puuttuu kokonaan.

Tämän jälkeen kokeilin emulaattoria testisovelluksella, mutta se ei suostunut käynnistymään, ja Visual Studio antoi erilaisia virheilmoituksia. Tämä johtui siitä, että koneellani on Windows 8.1 joka aiheutti yhteensopivuusongelmia. Asennettaessa Windows Phone SDK 8.0 -työkaluja Windows 8.1 -käyttöjärjestelmälle, tulee Visual Studioon ladata päivitys nimeltä Visual Studio 2012 Update 3 tai uudempi. Linkki päivitykseen löytyy Windows Phone SDK 8.0:n lataussivulta kohdasta "Install instructions" tai Microsoft Download Centeristä. Päivitys tulee asentaa vasta Windows Phone SDK 8.0 asennuksen jälkeen.

4.5 Uuden Windows Phone -projektin aloittaminen ja sovelluksen osat

Uusi projekti luodaan Visual Studion vasemmassa laidassa olevasta Start Page -palkista tai File -valikosta klikkaamalla New Project. New Project -ikkunassa määritetään projektin tallennushakemisto ja nimi sekä valitaan sovelluspohja (Templates). Valittavana on useita vaihtoehtoja C#, Visual Basic sekä C++ -ohjelmointikielillä. Valitaan C# Windows Phone App, jonka jälkeen päätetään kummalle Windows Phone alustaversiolle (WP8.0 tai WP7.1) sovellus toteutetaan.

Visual Studion näkymä on tuttu sitä aiemmin käyttäneille. Työkalut ovat piilossa vasemmassa palkissa ja Solution Explorer on oikealla. Uusia seikkoja ovat esimerkiksi Windows Phone Emulaattori yläpalkissa ja Device-palkki vasemmalla, josta voi valita näytön resoluution, teeman sekä näytön suunnan. Sovelluksen päänäkymä (MainPage.xaml) näkyy puhelimenä (Kuva 6), jonka näyttö on muokkausaluetta. Tähän näkymään voi lisätä elementtejä Toolbox-palkista raahaa ja pudota -menetelmällä ja ne ovat sijoitettavissa niin sanottuja gridline-viivoja apuna käyttäen.



Kuva 6. Visual Studio MainPage ja Solution Explorer

Visual Studio luo uuteen sovellusprojektiin valmiiksi Windows Phone -sovelluksessa tarvittavat kansiot ja tiedostot. Nämä ovat nähtävissä Solution Explorer -palkissa (Kuva 6). Sovelluksen MainPage.xaml -tiedosto on ensimmäisenä esillä. MainPage.xaml on sovelluksen oletussivu ja kuten aiemmin mainitsin tähän lähes tyhjälle sivulle voi alkaa lisäämään XAML-elementtejä. Tiedoston alta löytyy MainPage.xaml.cs, joka sisältää koodin. Sen koodi sisältää vain MainPage()-luontimetodin.

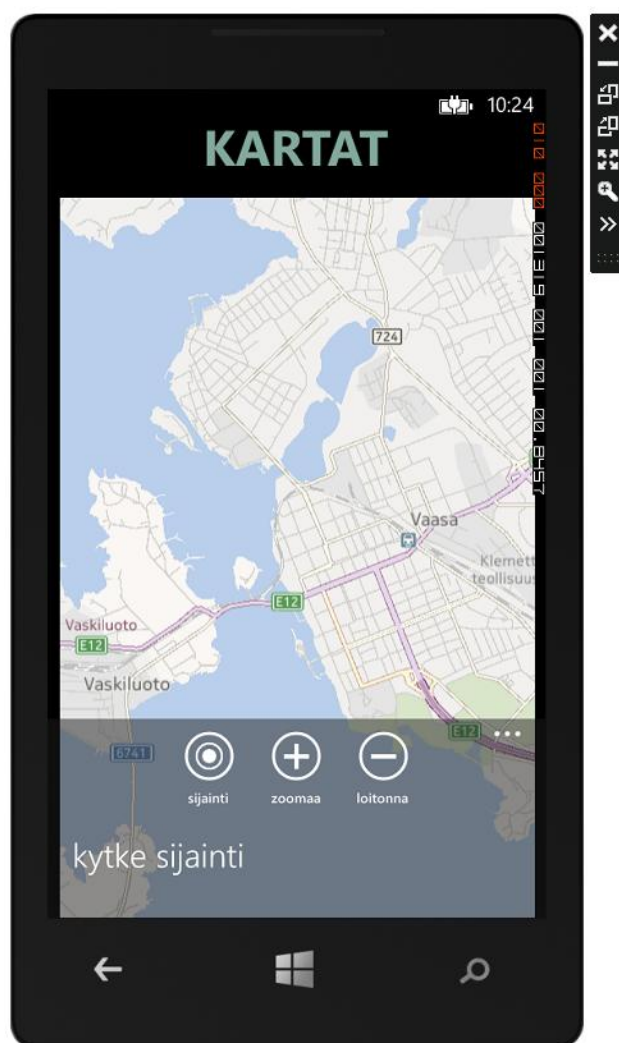
Properties-kansio sisältää WMAppManifest.xml -tiedoston, joka sisältää tietoja sovelluksesta sekä asetukset ja ominaisuudet, joita sovellus käyttää esimerkiksi GPS. AssemblyInfo.cs -tiedosto sisältää informaatiota, kuten otsikko, kuvaus, yhtiö ja versionumero. (Microsoft 2013 I)

Assets-kansiosta löytyy sovelluksen tile-kuvakkeet jotka näkyvät puhelimen valikossa ja aloitusnäytöstä. Nämä tulee muuttaa ennen sovelluksen julkaisua. App.xaml-tiedosto sisältää sovelluksen resurssit ja objektit sovelluksen elinkaaren

hallintaan. Itse elinkaaren metodit löytyy App.xaml.cs-tiedostosta. Metodeja ovat esimerkiksi Application_Launching() ja Application_Deactivated().

4.6 Esimerkkisovellus

Kartat-esimerkkisovellus on yksinkertainen ja sopii hyvin ensimmäiseksi testisovelluksiksi aloiteltaessa Windows Phone -sovelluskehitystä. Sovellus käyttää puhelimen sijaintitietoja ja lataa kartat Internetistä. Sovellus on toteutettu käyttäen Visual Studiota sekä C#:a ja XAML:ta.



Kuva 7. Kartat-sovellus ajettuna emulaattorilla

Sovelluksen käyttöliittymässä on kartta ja kolme painiketta. Kaksi zoomaus-kontrollia ja sijainti, jolla kytketään laitteen paikannus käyttöön tai pois. Sama sijaintikontrolli löytyy myös valikosta.

```
<!-- Kartat -esimerkkisovellus-->
<phone:PhoneApplicationPage
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:maps="clr-namespace:Microsoft.Phone.Maps.Controls;assembly=Microsoft.Phone.Maps"
    x:Class="Kartat.MainPage"
    mc:Ignorable="d"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="**"/>
        </Grid.RowDefinitions>

        <TextBlock HorizontalAlignment="Right" Margin="0,-10,28,10" TextWrapping="Wrap"
            Text="KARTAT" VerticalAlignment="Bottom" Foreground="#FF83AA9E"
            FontSize="48" FontWeight="Bold" Width="302" Height="56"/>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Margin="12,73,10,0" Grid.RowSpan="2">

            <maps:Map x:Name="Kartta" LandmarksEnabled="False" Loaded="Kartat_Loaded"/>

        </Grid>
    </Grid>
</phone:PhoneApplicationPage>
```

Kuva 8. Kartat-sovelluksen XAML-koodi

MainPage.XAML -tiedossa on vain TextBlock - ja Map -XAML-elementit. Valikon painikkeet on toteutettu C#-koodin puolella ApplicationBarIconButton - luokalla. Kartan ja sijainnin toteuttamiseen sovellus käyttää MapOverlay- ja GeoCoordinates -luokkia.

4.6.1 Sovelluksen C#-koodi

```

using System;
using System.Windows;
using Microsoft.Phone.Shell;
using Microsoft.Phone.Controls;
using Kartat.Resources;
using Microsoft.Phone.Maps;
using Microsoft.Phone.Maps.Controls;
using System.Device.Location; // Tarjoaa sijaintitieto-luokat.
using Windows.Devices.Geolocation;
using System.Windows.Shapes;
using System.Windows.Media;

namespace Kartat
{
    public partial class MainPage : PhoneApplicationPage
    {
        const int MIN_ZOOM_LEVEL = 1;
        const int MAX_ZOOM_LEVEL = 20;

        ToggleStatus SijaintiStatus = ToggleStatus.ToggledOff;

        GeoCoordinate currentLocation = null;
        MapLayer locationLayer = null;

        // Luontimetodi.
        public MainPage()
        {
            InitializeComponent();

            // Valikko.
            BuildLocalizedApplicationBar();

            //Sijainti.
            HaeSijainti();
        }

        private void Kartat_Loaded(object sender, RoutedEventArgs e)
        {
            MapsSettings.ApplicationContext.ApplicationId = "<applicationid>";
            MapsSettings.ApplicationContext.AuthenticationToken = "<authenticationtoken>";
        }
    }
}

```

```

void KytkeSijanti(object sender, EventArgs e)
{
    switch (SijaintiStatus)
    {
        case ToggleStatus.ToggledOff:
            ShowLocation();
            CenterMapOnLocation();
            SijaintiStatus = ToggleStatus.ToggledOn;
            break;
        case ToggleStatus.ToggledOn:
            Kartta.Layers.Remove(locationLayer);
            locationLayer = null;
            SijaintiStatus = ToggleStatus.ToggledOff;
            break;
    }
}

void ZoomIn(object sender, EventArgs e)
{
    if (Kartta.ZoomLevel < MAX_ZOOM_LEVEL)
    {
        Kartta.ZoomLevel++;
    }
}

void ZoomOut(object sender, EventArgs e)
{
    if (Kartta.ZoomLevel > MIN_ZOOM_LEVEL)
    {
        Kartta.ZoomLevel--;
    }
}

private void ShowLocation()
{
    //Luo pienen pisteen osoittamaan sijaintia.
    Ellipse myCircle = new Ellipse();
    myCircle.Fill = new SolidColorBrush(Colors.Blue);
    myCircle.Height = 20;
    myCircle.Width = 20;
    myCircle.Opacity = 50;

    MapOverlay myLocationOverlay = new MapOverlay();
    myLocationOverlay.Content = myCircle;
    myLocationOverlay.PositionOrigin = new Point(0.5, 0.5);
    myLocationOverlay.GeoCoordinate = currentLocation;

    locationLayer = new MapLayer();
    locationLayer.Add(myLocationOverlay);

    // Lisää pisteen kartalle.
    Kartta.Layers.Add(locationLayer);
}

```

```

private async void HaeSijanti()
{
    // Hankkii nykyisen sijainnin.
    Geolocator myGeolocator = new Geolocator();
    Geoposition myGeoposition = await myGeolocator.GetGeopositionAsync();
    Geocoordinate myGeocoordinate = myGeoposition.Coordinate;
    currentLocation = CoordinateConverter.ConvertGeocoordinate(myGeocoordinate);
}

private void CenterMapOnLocation()
{
    Kartta.Center = currentLocation;
}

#endregion

// Luo valikon.
private void BuildLocalizedApplicationBar()
{
    ApplicationBar = new ApplicationBar();
    ApplicationBar.Opacity = 0.5;

    // Sijainti nappula.
    AppBarIconButton appBarButton = new AppBarIconButton(new Uri("/Assets/AppBar/location.png", UriKind.Relative));
    appBarButton.Text = AppResources.AppBarToggleLocationButtonText;
    appBarButton.Click += KytkeSijanti;
    ApplicationBar.Buttons.Add(appBarButton);

    // Zoomaa nappula.
    appBarButton = new AppBarIconButton(new Uri("/Assets/AppBar/zoomin.png", UriKind.Relative));
    appBarButton.Text = AppResources.AppBarZoomInButtonText;
    appBarButton.Click += ZoomIn;
    ApplicationBar.Buttons.Add(appBarButton);

    // Loitonna nappula.
    appBarButton = new AppBarIconButton(new Uri("/Assets/AppBar/zoomout.png", UriKind.Relative));
    appBarButton.Text = AppResources.AppBarZoomOutButtonText;
    appBarButton.Click += ZoomOut;
    ApplicationBar.Buttons.Add(appBarButton);
}

```

```
        // Kytke sijainti.
        AppBarMenuItem appBarMenuItem = new
AppBarMenuItem(AppResources.AppBarToggleLocationMenuItemText);
        appBarMenuItem.Click += KytkeSijanti;
        ApplicationBar.MenuItems.Add(appBarMenuItem);

    }

    private enum ToggleStatus
    {
        ToggledOff,
        ToggledOn
    }
}
```

5 YHTEENVETO

Microsoft tarjoaa kattavat ja hyvät työkalut Windows Phone -ohjelmointiin, vaikka mobiilialusta on vielä nuori. Microsoft-kehitystyökalut ovat pitkään olleet suosiossa, etenkin Visual Studio on ehkä paras ohjelmointiympäristö. Ohjelmien rakentaminen on tehty helpoksi ja yksinkertaiseksi lisäämällä XAML-elementtejä raahaa-ja-pudota -menetelmällä käyttöliittymäluonnostelijaan.

Minulla on aiempaa kokemusta Visual Studiosta ja C# -ohjelmoinnista koulussa käydyistä kursseista. C# -ohjelmointi .NET-kirjastojen kanssa on helpoimpia ensiaskeleita ohjelmointiin. Aiempi kokemus .NET -ohjelmoinnista on suotavaa ennen Windows Phone -ohjelmointia, sillä kyseessä on puhelinalusta, jossa tulee ottaa huomioon enemmän seikkoja, kuten muistin ja akun käyttö.

Ennen opinnäytetyön aloittamista kokeilin Windows Phone SDK 7.1 -työkaluja. Kun verrataan SDK 8.0-työkaluja tähän vanhempaan, huomaa työkalujen selvästi kehittyneen parempaan suuntaan. Työkalut ovat selkeämmät ja kattavammat SDK 8.0:ssa, ja suurin syy tähän on uudempi versio Visual Studiosta. Windows Phone 8 -emulaattorin karttatyökalujen kanssa oli ongelmia. Karttatyökalulla pystytään ohjaamaan testattavan sovelluksen sijaintitietoja, mutta SDK 8.0:ssa nämä eivät reagoineet. Testattava sovellus toimi kumminkin vanhemmalla SDK 7.1:n emulaattorilla. SDK 8.0:n tukee vanhempia sovelluksia.

Opinnäytetyöni on vain pintaraapaisu Windows Phone 8 -ohjelmoinnista, mutta rajasin mielestäni hyvin tärkeimpiä asioita aloittelevalle sovelluskehittäjälle. Materiaalia löytyy kattavasti Internetistä. Windows Phone Dev Center tarjoaa paljon materiaalia ja esimerkkejä, joskin vain englanniksi. Lisäksi löytyy sivustoja, jotka tarjoavat videokursseja, esimerkkinä channel9.msdn.com. Opin mielestäni hyvin jo pelkästään näitten sivujen tarjoamilla materiaaleilla. Suosittelen kouluja lisäämään Windows Phone -ohjelmointia kurssitarjontaan.

Microsoft on lähtenyt nyt vahvasti mukaan mobiililaitteiden kilpaan, ja se on yhtenäistänyt eri alustoja Modern-UI -käyttöliittymällään, pilvipalveluillaan sekä Windows NT -käyttöjärjestelmätimellään.

LÄHTEET

Belfiore, Joe 2012. Windows 8: Shared Core – Joe Belfiore, Windows Phone 8. Viitattu 26.9.2013. <http://channel9.msdn.com/Events/Windows-Phone/Summit/Windows-8-Shared-Core>

Bhushan, Chandra 2013. The Brief History of Windows Phone. Viitattu 19.9.2013. <http://www.blogginghits.com/2013/07/04/the-brief-history-of-windows-phone/>

Greenfield, Tim 2013. Detecting screen resolution in Windows 8 and Windows Phone apps. Viitattu 20.9.2013. <http://programmerpayback.com/2013/08/31/detecting-screen-resolution-in-windows-8-and-windows-phone-apps/>

Kolakowski, Nicholas 2010. Microsoft Explains Windows Phone 7 Lack of Compatibility. Viitattu 19.9.2013. <http://www.eweek.com/c/a/Mobile-and-Wireless/Microsoft-Explains-Windows-Phone-7-Lack-of-Compatibility-588900/>

Microsoft 2013 A: Lock screen notifications for Windows Phone 8. Viitattu 20.9.2013. <http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj207048%28v=vs.105%29.aspx>

Microsoft 2013 B: Beta testing your app and in-app products. Viitattu 8.10.2013. <http://msdn.microsoft.com/en-us/library/windowsphone/help/jj215598%28v=vs.105%29.aspx>

Microsoft 2013 C: Submit your app. Viitattu 10.10.2013. <http://msdn.microsoft.com/library/windowsphone/help/jj206724%28v=vs.105%29.aspx>

Microsoft 2013 D: Targeted app distribution. Viitattu 18.10.2013. <http://msdn.microsoft.com/en-us/library/windowsphone/help/jj619160%28v=vs.105%29.aspx>

Microsoft 2013 E: Visual C# resources. Viitattu 4.11.2013. <http://msdn.microsoft.com/en-us/vstudio/hh341490.aspx>

Microsoft 2013 F: What is XAML? Viitattu 4.11.2013. <http://msdn.microsoft.com/en-us/library/cc295302.aspx>

Microsoft 2013 G: App activation and deactivation for Windows Phone. Viitattu 5.11.2013. <http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff817008%28v=vs.105%29.aspx>

Microsoft 2013 H: Windows Phone Store Test Kit. Viitattu 6.11.2013. <http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh394032%28v=vs.105%29.aspx>

Microsoft 2013 I: App manifest file for Windows Phone. Viitattu 18.11.2013.
<http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff769509%28v=vs.105%29.aspx>

Windows Phone Network 2012. Windows Phone 8 New Features – From Leaked SDK. Viitattu 20.9.2013. <http://windowsphonenetwork.com/2012/07/windows-phone-8-new-features-from-leaked-sdk/>

WMPoweruser 2012. Minimum hardware requirements for Windows Phone 8 revealed. Viitattu 26.9.2013. <http://wmpoweruser.com/minimum-hardware-requirements-for-windows-phone-8-revealed/>