

Opinnäytetyö (AMK)

Tietotekniikka

Ohjelmistotuotanto

2013

Jim Laine

WINDOWS PHONE -SOVELLUKSEN PÄIVITYKSEN SUUNNITTELU



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikka | Ohjelmistotuotanto

Toukokuu 2013 | 47

Ohjaaja: Tiina Ferm

Jim Laine

WINDOWS PHONE -SOVELLUKSEN PÄIVITYKSEN SUUNNITTELU

Tämän opinnäytetyön tarkoituksena on tarkastella ohjelmiston suunnittelua päivittämisen helppouden näkökulmasta. Tällä tarkoitetaan esimerkiksi tilannetta, jolloin sovellukseen tehdään täysin uusi ominaisuus. Ominaisuudella on esim. päälle/pois -tila, joka halutaan tallentaa samaan tiedostoon käyttäjän asettamien aikaisempia ominaisuuksia koskevien säätöjen kanssa.

Työssä kuvataan Microsoftin Windows Phone -mobiilikäyttöjärjestelmää, joka on uuden näköinen, nopeasti kasvava alusta, sekä siihen liittyviä mahdollisuuksia. Microsoft Windows 8:n tuoteperhe julkaistiin vuoden 2012 lopussa ja siinä yhteydessä julkaistiin .NET-kirjastoon 4.5-päivitys, joka tuo kehittäjille lisää työkaluja sovelluskehitykseen.

Lisäksi selvennetään, minkälaisen prosessin kehittäjä joutuu käymään läpi, jotta sovelluksen julkaiseminen Windows Phone Marketplaceen on mahdollista. Työssä kerrotaan, kuinka julkaisuprosessissa tarvittavat sertifikaatit hankitaan ja miten Pubcenter mainonta API:n avulla voidaan esittää mainoksia sovelluksessa.

Työ kuvaa erityisesti sovelluksen rakennetta, jotta jatkokehittäminen on mahdollisimman vaivatonta ja ongelmattonta. Tarkoituksena on saada uudet ominaisuudet toimimaan saumattomasti vanhojen kanssa, jotta päivitys ei häiritse käyttäjien olemassa olevia asetuksia eikä tuota kehittäjälle turhaa työtä versiokohtaisella koodilla.

Opinnäytetyön käytännön osassa tehdään päivitys Key Zone -sovellukseen. Key Zonessa on sovellettu jo aiemmin opittuja asioita sovelluksen rakenteen suunnittelun osalta, mikä teki päivityksen tekemisen helpoksi. Päätteeksi julkaistaan Key Zone 1.1 -päivitys Windows Phone Marketplaceessa.

ASIASANAT:

Windows Phone, Pubcenter, XNA framework, Microsoft, mobiili, sovelluskehitys

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Software Engineering

May 2013 | 47

Instructor: Tiina Ferm

Jim Laine

DESIGNING WINDOWS PHONE APP UPDATE

The aim of this thesis is look into the design of program with the aim of keeping updating procedure as easy as possible. For example, when a totally new feature is created to an app that can be set to on/off and the state is saved in the same file that contains the user defined settings from earlier version.

The study focuses on Microsoft Windows Phone -mobile operating system, which has a fresh outlook and is a rapidly growing platform. While Microsoft published the Windows 8 product family the .NET programming library got updated to 4.5 as well, bringing more tools for a program developer.

In addition this thesis clarifies what kind of process the developer needs to go through with to get an app published on Windows Phone Marketplace as well as how to get the needed certificates and setting up the Pubcenter API to provide ads into the developed app.

The focus is on the program structure to make the further developing as trouble-free as possible. The effort is to make the new features to work seamlessly with the previous ones, without developer making extra work with version specific code.

The practical part of this thesis concentrates on updating the app called Key Zone. Key Zone has been designed with the above mentioned structure which made it easy to update the app. Finally Key Zone 1.1 is published on Windows Phone Marketplace.

KEYWORDS:

Windows Phone, Pubcenter, XNA framework, Microsoft, mobile, app development

SISÄLTÖ

SANASTO	6
1 JOHDANTO	8
2 WINDOWS PHONE -KÄYTTÖJÄRJESTELMÄ	10
2.1 .NET framework	11
2.1.1 .NET:n versioiden 4.5 ja 4.0 erot	12
2.1.2 MSIL ja CLR	12
2.2 XNA framework	12
3 KEHITYSTYÖKALUT	14
3.1 Dev Center	14
3.2 Pubcenter	16
3.3 Subversion	18
4 PÄIVITYS TAPAHTUMANA	20
4.1 Windows Phone Marketplace	20
4.2 Loppukäyttäjän näkökulmasta	20
4.3 Kehittäjän näkökulmasta	21
4.3.1 Entertainment Software Rating Board – ESRB	23
4.3.2 Pan European Game Information – PEGI	24
5 JULKAISTUN SOVELLUKSEN PÄIVITTÄMINEN	27
5.1 Huomioitavia asioita	27
5.1.1 Olemassa oleva käyttäjäkunta	27
5.1.2 Uudet ominaisuudet	28
5.1.3 Sovelluksen tallennustiedostojen tietorakenne	29
5.1.4 Marketplace alueisiin kohdentaminen	30
5.1.5 Versionhallinta projektissa	31
5.1.6 Testaus	32
5.1.7 Sertifiointi ja Microsoftin vaatimusten täyttö	33
5.2 Miksi halutaan päivittää?	34
6 KEY ZONE -PÄIVITYS	36
6.1.1 Key Zone 1.0 versio	38

6.1.2 Key Zone 1.1 versio	40
6.1.3 Key Zone jatkokehittäminen	43
7 YHTEENVETO	45
LÄHTEET	46
 KUVAT	
Kuva 1. Windows tuoteperheiden kehitys Windows 8 tuoteperheeseen asti.	11
Kuva 2. Dev Center, sovellusten latausmäärät.	15
Kuva 3. Pubcenter application & ad unit ID.	17
Kuva 4. Pubcenter mainobanneri -kategoriat.	17
Kuva 5. Microsoft Advertising SDK referenssi.	18
Kuva 6. Dev Center, päivityksen lähettäminen Marketplaceen.	22
Kuva 7. ESRB -ikäluokitus.	24
Kuva 8. PEGI -ikäluokitus.	25
Kuva 9. Tiedostomuutoksien hallinta.	29
Kuva 10. XML-tiedoston tietorakenteen luonti.	30
Kuva 11. Jakelumaat, Windows Phone Marketplace.	31
Kuva 12. Key Zone 1.0 Windows Phone Marketplace.	37
Kuva 13. Polymorphism.	38
Kuva 14. Key Zone, pisteiden lasku -bugi.	42
Kuva 15. Key Zone 1.1 Windows Phone Marketplace.	43

SANASTO

WP7	Windows Phone 7.0, 7.5, 7.8 -mobiilikäyttöjärjestelmät
WP8	Windows Phone 8.0 -mobiilikäyttöjärjestelmä
IOS	Applen mobiilikäyttöjärjestelmä
Android	Googlen mobiilikäyttöjärjestelmä
C#	ohjelmointikieli
C	ohjelmointikieli
C++	ohjelmointikieli
VB	Visual Basic -ohjelmointikieli
API	ohjelmistorajapinta
.NET	virtuaalinen ohjelmointikehys
CLR	ajoympäristö (Common Language Runtime)
MSIL	esikäännettyä ohjelmakoodia (Microsoft Intermediate Language)
XNA	pelinkehityskehys, jota voi käyttää Windows, Xbox360 tai Windows Phone -pelien kehittämiseen
XML	merkintäkieli/standardi (Extensible Markup Language)
SVN	Subversion-versionhallintaohjelmisto
Repositorio	paikka, johon versionhallintaohjelmiston tekemä versiointi tapahtuu
Commit	versionhallinnassa tehtävä projektitiedostojen muutosten lähetys itse repositorioon

Revert	paluu edelliseen versionhallinnan repositorioversioon
IDE	ohjelmointiympäristö (Integrated Development Environment)
SDK	ohjelmistokehityspaketti (Software Development Kit)
Open source	avointa lähdekoodia, joka on vapaasti levitettävissä mukana tulevan lisenssin rajoittein
Bug, Bugi	ohjelmointivirhe lähdekoodissa

1 JOHDANTO

Opinnäytetyössä käsitellään Windows Phone -sovelluksen tekemiseen vaadittavia ympäristöjä, jonka jälkeen keskitytään päivittämisen kannalta oleellisiin ympäristöihin, sekä niiden toiminnallisuuksiin. Lopuksi kerrotaan omakohtaiset kokemukset aikaisemmin julkaistun mobiilisovelluksen päivittämisestä ja syvennytään mahdollisiin ongelmiin, joita matkalla sovelluksen versiosta 1.0 versioon 1.1 tulee vastaan.

Windows Phone -alustan ominaisuuksia avataan havainnollistaen lähinnä alustan tuomia mahdollisuuksia ohjelmistokehittämisen kannalta. Näitä ovat mm. käytettävät ohjelmointikielet ja ympäristöt.

Käytännön osuudessa käytettävät ympäristöt pyritään havainnollistamaan lukijalle, jotta tämä tietää mistä on kyse ennen käytännön osuuden alkamista. Tärkeimpiä Windows Phone spesifisiä ympäristöjä toteutuksen osalta ovat XNA framework, Dev Center ja Windows Phone Marketplace, joita pyritään avaamaan tarkemmin ja kattavammin. Näin tehdään koska ympäristöihin ja ympäristöjen ominaisuuksiin viitataan jatkuvasti, mitä pidemmälle työtä luetaan.

Tässä työssä tuodaan esille erinäisiä asioita ja ehdotuksia, mitä kannattaa alun alkaen ottaa huomioon, kun suunnitellaan sovellusta julkaistavaksi. Huomioissa painotetaan sovelluksen päivittämisen tekemistä vaivattomaksi, kun sovellus on jo julkaistuna. Luvussa 5.1.3 pyritään esimerkin kautta avaamaan, miten helppoa on uuden muuttujan arvon lisääminen XML-tiedostoon, jonka rakenne on suunniteltu helposti päivitettäväksi.

Käytännön toteutuksen alustana toimii Windows Phone - mobiilikäyttöjärjestelmä, josta vielä tarkennettuna luvussa 2.2 enemmän avattu XNA framework -pelinkehitysympäristö. Syitä, miksi valitsin Windows Phonen projektini alustaksi on ensinnäkin opiskelijaksi tunnistautumisen kautta saatu ilmainen kehittäjälisenssi Microsoft Dev Centeriin. Ohjelmointikielenä C# vaikutti mieluisalta verrattuna IOS:n käyttämään object C:hen tai Androidissa

käytettävään Javaan, koska olin tutustunut kieleen jo aiemmin. Kasvava alusta ja uudenlainen ulkoasu muihin käyttöjärjestelmiin verrattuna herätti mielenkiintoa jo erilaisuuden vuoksi, mutta myös koska kehittäjä määrä oli vielä melko pieni, oli mahdollisuudet nousta pinnalle onnistuneen sovelluksen avulla.

Päivitys tapahtuman toteutus tehdään Windows Phone -peliin nimeltä Key Zone, jonka ideasta ja toiminnallisuudesta kerrotaan luvussa 6. Peli on ollut julkaistuna hieman yli puoli vuotta ja latausmäärät olivat jo lähteneet laskuun, minkä vuoksi se haluttiin päivittää. Peliä päivitettäessä halutaan, ettei päivitetty versio kadota aikaisemmin tallennettuja tietoja vaan käyttää näitä hyödykseen. Tällaisia tietoja ovat Key Zonen kohdalla pelaajan tallentamat asetukset sekä parhaat tulokset.

2 WINDOWS PHONE -KÄYTTÖJÄRJESTELMÄ

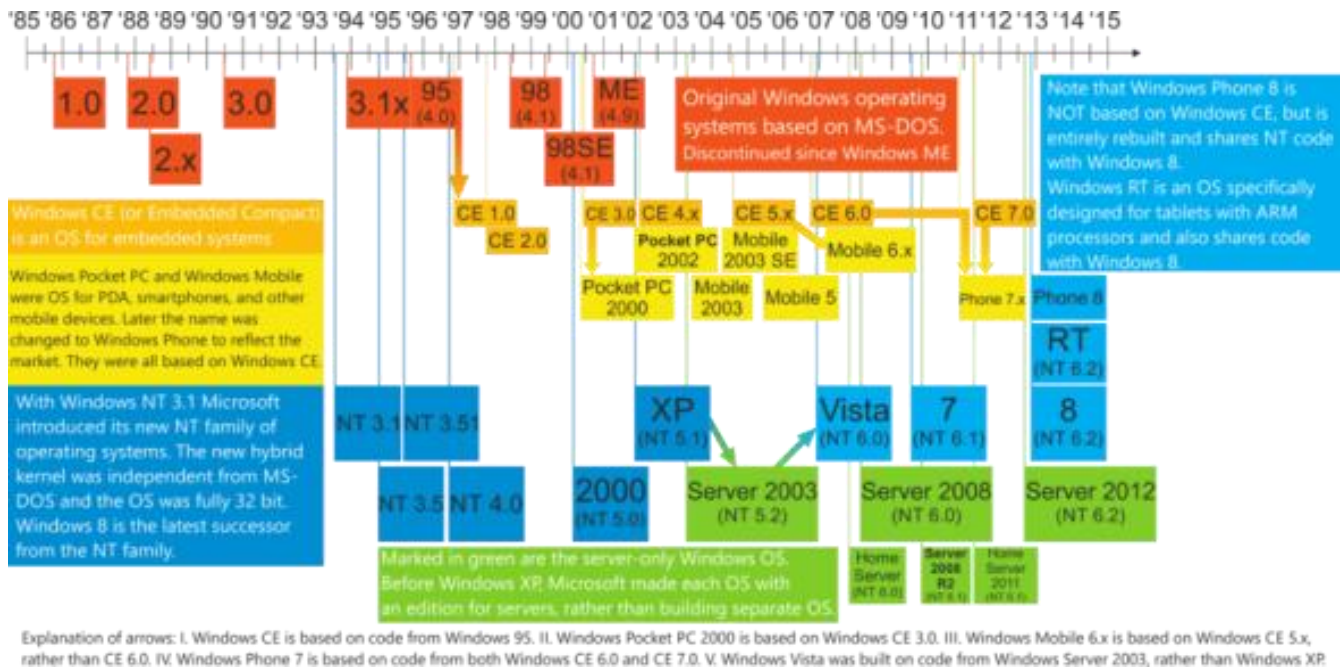
Windows Phone on Microsoftin kehittämä suljetun lähdekoodin mobiilikäyttöjärjestelmä. Käyttöjärjestelmä tunnetaan sen laatikkomaisesta Metro käyttöliittymästä.

Windows Phone 7 perustuu Windows CE käyttöjärjestelmäyttimeen, joka on suunniteltu toimimaan rajallisilla määrillä muistia ja pienitehoisilla prosessoreilla. Windows CE -käyttöjärjestelmän pääasiallisista käyttökohteista mainitsemisen arvoisia ovat kämmentietokoneet, jotka vastaavat toiminnoiltaan nykykäsitystä puhelimesta.[1]

Windows Phone 7.0 versio julkaistiin euroopassa 2010 elokuussa. Alkuperäisen Windows Phone 7.0 julkaisun jälkeen on käyttöjärjestelmään tullut 7.5(Mango) päivitys sekä viimeisimpänä 7.8 päivitys. [1]

Microsoft julkaisi Windows Phone 8 -mobiilikäyttöjärjestelmänsä ennen 7.8 päivitystään, jossa on mm. paljon laajempi tuki näytön resoluutioille kuin Windows Phone 7 -puhelimissa, joissa se oli rajattu kokoon 480x800 (vaaka x pysty). Windows Phone 8 mahdollistaa moniytimellisten prosessoreiden käytön tuoden lisää tehoa kehittäjien käytettäväksi. WP8 käyttöjärjestelmäydin ei ole Windows CE pohjalle rakennettu kuten WP7, vaan se perustuu Windows NT:n pohjalle. Windows NT:n käyttöjärjestelmäyttimeen perustuminen tuo enemmän yhteensopivuutta Windows 8 -käyttöjärjestelmien ja Windows Phone 8 -mobiilikäyttöjärjestelmän välille, sillä ne kuuluvat samaan ohjelmistoperheeseen. [1]

Kuva 1 havainnollistaa tarkemmin, kuinka käyttöjärjestelmän kehitys on päätynyt nykyiseen tilanteeseen asti. Siitä voi huomata, kuinka Windows CE:n pohjalta lähtöisin oleva WP 7 ei kuulu samaan haaraan kuin Windows 8 -tuoteperheen julkaisun yhteydessä julkaistu WP 8.



Kuva 1. Windows tuoteperheiden kehitys Windows 8 tuoteperheeseen asti. [2]

2.1 .NET framework

.NET on Microsoftin kehittämä erittäin laaja ohjelmistokirjasto, joka tukee erittäin montaa eri ohjelmointikieltä. Windows Phone kehittäjä pystyy hyödyntämään suurta osaa .NET frameworkista, vaikka osa metodeista ja luokista toimii hieman eri tavoin johtuen puhelin spesifisistä kirjastoista.

.NET on luotu toimimaan Windows -alustoilla ja Windows Phone 7:n osalta sitä voi hyödyntää käyttämällä C# tai VB -ohjelmointikieliä. Windows Phone 8 mahdollistaa myös natiivikoodin(C, C++) käytön, jonka Microsoft toivoo houkuttelevan lisää ohjelmistokehittäjiä WP leiriin. [3]

Vartenotettavana vaihtoehtona mainittakoon myös Mono -projekti, jonka tarkoituksena on laajentaa .NET myös muille kuin Windows -pohjaisille alustoille. Kehittyessään pidemmälle projekti helpottaisi monelle alustalle kehittämistä hyödyntäen suurimmilta osin samaa .NET koodia. [4]

2.1.1 .NET:n versioiden 4.5 ja 4.0 erot

Vaihto .NET 4.5 frameworkiin tapahtui samalla kun Microsoft julkaisi Windows 8 tuoteperheensä. Edeltävät Windows Phone 7 -mobiilikäyttöjärjestelmä ja Windows 7 käyttivät .NET 4.0 frameworkia.

.NET 4.5 tärkeimpiä uusia ominaisuuksia on asynkronisen operoinnin muutos. Asynkronisuus tehtiin myös paljon helpommaksi luomalla uusia metodityyppejä uuden `async` -mallin alle, joka yksinkertaistaa entistä `BackgroundWorker` -luokan kautta pyöräytettyä asynkronista operointia. [5]

Microsoft lisäsi Zip kompressoinnin vasta 4.5 version myötä .NET frameworkiin, vähentääkseen kolmasien osapuolten kirjastojen käyttöä. Myös monia muita parannoksia ja toimintoja lisättiin ohjelmointikehykseen, jotka helpottavat ohjelmistokehittäjän työtä ja vähentävät mahdollisten tietoturvariskien syntymistä käyttämällä .NET omia kirjastoja kolmansien osapuolien aikaansaannoksien puolesta. [5]

2.1.2 MSIL ja CLR

MSIL on Microsoftin kehittämää esikäännettyä ohjelmakoodia, johon kaikki .NET pohjaiset kielet ensin käännetään. MSIL mahdollistaa .NET frameworkin usean ohjelmointikielen tuen.

CLR komponentti kääntää .NET koodatut ohjelmisto komponentit ajonaikana esikäännetystä ohjelmakoodista (MSIL) binäärikoodiksi virtuaalikoneella. Syntynyttä binäärikoodia on mahdollista suorittaa suoraan halutulla alustalla. [6]

2.2 XNA framework

Toinen kahdesta ympäristöstä, jolla voidaan kehittää ohjelmia Windows Phonelle. XNA on pelinkehitykseen tarkoitettu pelkistetty versio Microsoftin Direct X -pelikehitysympäristöstä. XNA framework on kehittynyt 4.0 versioon

asti ja sen kehitystä ei näillä näkymin enää jatketa, sillä WP8 löytyy jo Direct X tuki. [7]

Direct X ympäristön keskittyessä pelien näyttävyyteen ja parhaimpien pelien luontiin, pyritään XNA frameworkilla saamaan indie tyyppisiä pelejä aikaan. Itse kehittäminen XNA:lla on suoraviivaisempaa ja sillä kehitetyt pelit saadaan helposti portattua Microsoftin muille alustoille. Suoraviivaisuus on tarkoituksella tehty, jotta kehittäjät voivat keskittyä olennaisempaan, eli käytettävyyden hiomiseen. [7]

XNA:ta on mahdollista hyödyntää tällä hetkellä kehitettäessä pelejä Windows, Windows Phone -käyttöjärjestelmille tai Xbox 360 -pelikonsolille. Peli on mahdollista konvertoida alustalta toiselle hyvin pienin muutoksin. Parhaimmassa tapauksessa ohjelmistokehittäjän ei tarvitse kuin skaalata grafiikat ja muuttaa input esim. kosketuksella toimivasta käyttämään ohjainta. [7]

3 KEHITYSTYÖKALUT

Matkalla ohjelman kehittämisestä julkaisemiseen asti tarvitaan paljon muutakin kuin pelkkää ohjelmakoodin kirjoittamista, jota on mahdollista suorittaa vaikkapa komentorivillä. Alkuun on tarpeen löytää kehittäjän kannalta toimiva IDE, josta löytyy kääntäjä Windows Phone -ympäristöön. Projektin aikana käytetään enimmäkseen Visual Studio 2010 IDE ja lopuksi Visual Studio 2012, joista löytyy kääntäjä ja emulaattori Windows Phone ympäristöön. IDE:n valinnan jälkeen kannattaa tutustua julkaisu mahdollisuuksiin mm. Marketplace, Store ja kuinka itse julkaisu prosessi tapahtuu.

3.1 Dev Center

Microsoftin sovelluskehittäjä yhteisö, joka vielä vuoden 2012 keväänä tunnettiin nimellä App Hub. Täältä saa hankittua lisenssin, jonka avulla on mahdollista julkaista omia ohjelmia Windows Phone -puhelimille tai Xbox 360 -pelikonsolille.

Dev Center keskustelufoorumista Windows Phone tai Xbox 360 kehittäjä voi hakea apua koskien sovelluskehitystä. Kaikki rekisteröityneet käyttäjät saavat tunnukset foorumin käyttämiseen, joka mahdollistaa avun kysymisen muilta kehittäjiltä tai Microsoftin moderaattoreilta, mikäli kysymys on tarpeeksi hankala.

Julkaistujen tai julkaisuun menevien ohjelmien hallinnointi tapahtuu niinkään täällä. Dev Centerissä on mahdollista seurata julkaistujen ohjelmien latausmääriä, kaatumismääriä ja arvosteluja.

Latausten numeerisia määriä havainnollistetaan Dev Centerissä graafeilla, joista saa vielä kohdennettua hiirellä tietylle päivälle kohdistuneet latausmäärät. Kuvassa 2 näkyy latausmääriä havainnollistava graafi, jossa on kohdistettuna 1.11.2012 (laatikko keskellä graafia) ja siihen päivään mennessä relevantit tiedot. Graafeja voi myös rajata kattamaan vain tiettyä valittua aikaväliä ja tutkia sitä, vaikka päivittäisten latausten osalta.



Kuva 2. Dev Center, sovellusten latausmäärät.

Käyttäjien antamat arvostelut näkyvät omalla välilehdellä. Arvostelut saa jaoteltua maittain ja mikäli siihen on annettu kirjallinen kommentti itselle vieraalla kielellä, saa sen käännettyä käyttäen Microsoft Translator:ia halutulle kielelle.

Kehittäjän ollessa rekisteröitynyt Dev Center sivustoon ja omistaessa voimassa olevan lisenssin, saa hän vähintään kuukauden välein Registered Developer Newsletter sähköpostiviestin Microsoftilta. Tämä sisältää yleensä ajankohtaista tietoa alustan päivityksistä, uusista jakelumaista, muutoksista palveluihin mm. Pubcenter, WP Markeplace tai tulevista tapahtumista koskien Windows Phone - mobiilikäyttäjärjestelmää. Viestissä esitellään yhteistyökumppaneita ja heidän kehittämiään rajapintoja. Esimerkkinä esiteltävistä rajapinnoista Photon - palvelin API, joka tarjoaa kehittäjille tietyllä määrällä käyttökuormaa palvelua ilmaiseksi ja sen ylittyessä palvelu maksaa jonkin verran antaen näin

mahdollisuuden saada sovellus aluilleen ennen sen ylläpito kustannuksien kehittymistä.

3.2 Pubcenter

Microsoftin kehittämän mainonta ohjelmiston hallintasivusto, jolla on mahdollista hallinnoida mainosten jakelua eri käyttökohteissa joissa kehittäjä on asettanut Microsoftin mainonta ohjelmiston käyttöön. Näitä ovat mm. internet sivustot ja Windows Phone -sovellukset.

Mainosten laittamiseksi omaan WP sovellukseen tarvitsee ensinnäkin ladata Microsoft Advertising SDK for Windows Phone -paketti, joka asentuu Visual Studio laajennoksena (extension). Paketin asentamisen ja Visual Studio restartin jälkeen on mahdollista ottaa mainonta API käyttöön. [8, s. 368 – 371]

Advertising SDK:n asentamisen jälkeen tulee luoda tunnukset Microsoft Pubcenteriin tai käyttää olemassa olevia live tunnuksiaan. Kirjautumisen tai tunnuksien luomisen jälkeen Pubcenterissa valitaan setup välilehti ja register application sekä create ad unit (katso kuva 3), joista saadaan ApplicationID ja AdUnitID. Kumpikin id tarvitaan mikäli halutaan saada mainoksia sovellukseen tai peliin asetettuihin bannereihin, sekä tuloja niiden avulla.

The screenshot shows the Microsoft Advertising PubCenter dashboard. At the top, there's a navigation bar with 'Reports', 'Setup', and 'Accounts'. Below that, a sub-navigation bar includes 'Overview', 'Applications', 'Ad units', 'Global ad exclusions', and 'Channels'. The main content area is titled 'Application advertising' and provides instructions on viewing recently created Windows Phone and Windows 8 applications and ad units. It features three tables: 'Applications' with columns for Application name, Application ID, and Device type; 'Mobile application ad units' with columns for Ad unit name, ID, and Display type; and 'Windows 8 application ad units' with columns for Ad unit name, ID, Application name, and Display type. A 'Global ad exclusions' table is also present with columns for Ad exclusion type and Excluded items. On the right side, there are two green callout boxes: 'Help on this page' with a link 'What is an ad unit?' and 'Integrate' with instructions on downloading the Microsoft Advertising SDK for Windows Phone and Windows 8.

Kuva 3. Pubcenter application & ad unit ID.

Luodulle sovellus kohtaiselle mainosbannerille (Ad unit) kehittäjä voi määrittellä minkä tyyppisiä mainoksia halutaan näyttää. Mainos tyyppiä voidaan valita yhteensä kolme, eikä sillä ole merkitystä miltä tasolta. Eli on mahdollista valita Yksi ylempään tason laji ja kaksi alemman tason toisesta mainos lajista.

Mainokset ovat lajiteltuina kahteen eri tasoon. Kuvassa 4 nähdään kuinka aihealueiden valinta toimii. Aivan vasemmalta löytyvät ylätasot, keskellä valittuna olevat sisältämät alalajit ja oikealla valitut. Tier 1 kattaa yleisemmällä tasolla mitä siinä olevat Tier 2 alatasot käsittävät.

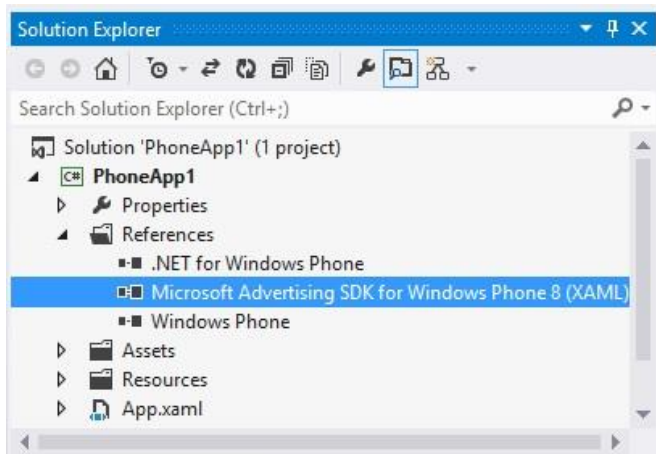
Ad categories [?](#)

Select up to three categories from the tier 1 and tier 2 categories. Click a tier 1 name to see the tier 2 categories within it. These categories set the general guidelines for what types of ads appear in your ad unit. Some ads might apply to more than one category or otherwise appear to be out of place.

Tier 1 categories	Tier 2 categories	Selected categories
<input checked="" type="checkbox"/> Arts & Entertainment (7) <input type="checkbox"/> Automotive (23) <input type="checkbox"/> Business (12) <input type="checkbox"/> Careers (11) <input type="checkbox"/> Education (15) <input type="checkbox"/> Family & Parenting (9) <input type="checkbox"/> Food & Drink (18) <input type="checkbox"/> Health & Fitness (45) <input type="checkbox"/> Hobbies & Interests (31)	<input type="checkbox"/> Books & Literature <input type="checkbox"/> Celebrity Fan/Gossip <input type="checkbox"/> Fine Art <input type="checkbox"/> Humor <input type="checkbox"/> Movies <input type="checkbox"/> Music <input type="checkbox"/> Television	<input checked="" type="checkbox"/> Sports <input checked="" type="checkbox"/> Technology & Computing <input checked="" type="checkbox"/> Arts & Entertainment

Kuva 4. Pubcenter mainobanneri -kategoriat.

Aikaisempien vaiheiden jälkeen asetetaan omassa Visual Studio projektissa referenssi viittaamaan Microsoft Advertising SDK for Windows Phone -laajennukseen (extension), kuten kuvassa 5. Vasta sen jälkeen on mahdollista kutsua tarvittavaan Adcontrol -luokkaa, johon määritellään mainosten koko vaaka- x pystyakselilla, Pubcenterissa tehty ApplicationID ja AdUnitID.



Kuva 5. Microsoft Advertising SDK referenssi. [9]

Mainosten kautta saatujen tulojen tarkkaileminen onnistuu niin ikään Pubcenter:in kautta. Tulojen ansaitseminen perustuu kahteen eri tilastoon; Impressions ja CTR% eli Click Through Rate prosentti. CTR% kertoo kuinka suurella todennäköisyydellä käyttäjä erehtyy painamaan mainosta verrattaen mainosten näyttämisen määrään. Edeltävien perusteella lasketaan eCPM eli Effective cost-per-thousand impressions, joka kertoo kuinka paljon mainoksen näyttäminen kustantaa mainostajalle tuhatta esityskertaa kohden. [10]

3.3 Subversion

Subversion on versionhallintaohjelmisto, joka helpottaa ohjelman kehitystä projektin laajentuessa. Mahdollistaa revertoinnin edelliseen versioon mikäli jotain menee pieleen uusimmassa commitissa. Jokaisen commitin ohessa jätettävä viesti informoi kehittäjää, millaisia muutoksia ohjelmakoodiin on tehty uusimman päivityksen yhteydessä.

Linux puolella Subversion toimii täysin komentorivipohjaisena versiona, jolla suoritetaan täysin samat operaatiot kuin Windowsin graafisella. Käyttöjärjestelmästä riippumattomuus mahdollistaa ohjelmakoodin muutoksien lähettämisen eli committien tekemisen samaan repositorioon eri käyttöjärjestelmistä.

Projektissa käytettiin TortoiseSVN -ohjelmaa, joka on yksi monesta Subversion clientista Windows -käyttöjärjestelmälle. Ohjelma integroituu osaksi Windowsin resurssienhallintaa ja mahdollistaa tällöin käyttämisen graafisesti, vaikka käytettävät komennot ovat täysin samat kuin komentoriviversiossa.

4 PÄIVITYS TAPAHTUMANA

4.1 Windows Phone Marketplace

Windows Phone Marketplace on paikka, josta peruskäyttäjät saavat ladattua sovelluksensa puhelimeen. Se on myöskin ainoa paikka mistä puhelimeen, josta ei ole poistettu lukitusta kehitys käyttöön (developer unlock), on mahdollista saada ladattua ja asennettua sovelluksia.

Marketplaceen on mahdollista päästä käsiksi internetin välityksellä, mikäli haluaa selata puhelimen sovelluksia tietokoneella. Tällöin on helpompaa kirjoittaa hakuja käyttäen tietokoneen näppäimistöä, sekä selata sovelluksia suuremmalta ruudulta. Miinuksena todettakoon kuvankaappauksien selaaminen, sillä huomattava määrä sovelluksista toimii vaakasuuntaisesti ja tietokoneella selatessa kuvankaappaukset aukeavat aina pystysuuntaisesti.

Puhelimesta löytyy oma Marketplace -sovellus, joka mahdollistaa sovelluskaupan selailun. Puhelimella on myös mahdollista avata sovelluskauppa nettiselaimen välityksellä, mutta sieltä ohjataan silti suoraan Marketplace -sovellukseen vaikka linkin kirjoittaisikin suoraan nettiselaimen osoitepalkkiin.

4.2 Loppukäyttäjän näkökulmasta

Itse käyttäjille päivittäminen tapahtuu Microsoftin sovelluskaupan, eli Windows Phone Marketplacen kautta. Se ilmoittaa loppukäyttäjälle Windows Phone Marketplace -sovelluksen kautta, kun johonkin jo ladattuun ja asennettuun sovellukseen on saatavilla päivitys.

Päivityksestä näytettävä live tile ilmoitus saattaa tulla jopa päivää myöhemmin, kuin päivitys todellisuudessa on Marketplaceen ilmestynyt. Tämä aiheuttaa

mahdollisuuden selata Marketplacea kyseiseen sovellukseen itse puhelimella ja huomata päivityksen olevan saatavilla.

4.3 Kehittäjän näkökulmasta

Ohjelmistokehittäjä kirjautuu Windows Dev Centeriin, jolla hallinnoidaan kaikkia julkaistuja sovelluksia.

Kehittäjä lähettää uuden xap-tiedoston Dev Centeriin, joka on rakennettu (built) versio Windows Phone -sovelluksesta. Tiedostomuoto sisältää sen mitä on Visual Studioissa projektiin lisätty, eli muita tiedostoja ei tarvitse lähettää Dev Centeriin. Esimerkiksi kehittäjän ei tarvitse erikseen lähettää musiikkitiedostoa, mikäli sovelluksessa soi taustalla kokoajan mp3-muotoinen kappale.

Aikasemmin asetettuja tietoja on mahdollista muokata, kuten mm. seloste teksti, hakemiseen suunnatut sanat ja kuvankaappaukset. Kuvasta 6 huomataan kuinka osa tiedoista on pakollisia ja tekevät julkaisusta mahdotonta mikäli ei niitä täyty. Kaikkein oleellisinta on päivittää selosteteksti vastaamaan tapahtuneita muutoksia sovellukseen, jotta sovelluksen lataamista harkitseva käyttäjä olisi tietoinen mitä on lataamassa.



Secure dev.windowsphone.com/en-us/AppSubmission/Hub

Update app




Key Zone

You've spent hours developing and designing your app, and now it's time for the rest of the world to experience your masterpiece. In just two steps we'll gather the information we need to successfully launch your app in the Windows Phone Store. [Learn more](#) about the steps for successfully submitting your app.

Required

-  **App info**
Give your app an alias, price it, and enter other relevant info
-  **Upload and describe your XAP(s)**
For each XAP in your app, this is where you'll enter descriptions and upload screenshots that will showcase your app in the Store.

Optional

-  **Add in-app advertising**
Getting paid through ads? It's all here.
-  **Market selection and custom pricing**
For apps, you have the option to define different pricing and availability for different countries/regions.
-  **Map services**
Get the token required to use map services in your app.

[Show any errors and warnings](#)

Kuva 6. Dev Center, päivityksen lähettäminen Marketplaceen.

Kehittäjällä on mahdollisuus kirjoittaa lokalisoituna Marketplacessa näkyvä teksti, joka tarkoittaa käytännössä, että tietyssä maassa se näkyy halutulla kielellä. Näin saadaan helposti kirjoitettua teksti lokalisoitulla kielellä ja halutussa muodossa maihin joihin sovellus on laitettu jaettavaksi. Tekstiä ei ole pakko lokalisoida vaan on myös mahdollista tehdä vain englanninkielinen versio ja esittää teksti englanniksi jokaisen maan marketplace:ssa.

Eräs suurimmista ongelmista tulee vastaan vasta ikärajojen kanssa. Niitä ei vielä App Hub aikoihin ollut ylipäänsä käytössä Windows Phone Marketplacessa tai ainakaan niitä ei vaadittu kehittäjän ladatessa puhelimelle tarkoitettua sovellustaan sivustolle.

Windows Phone Marketplacessa olevista ikärajoja tarkkailevista organisaatioista vain muutamalla on käytössä internetin välityksellä täytettävä lomake, näistä tunnetuimmat ovat ESRB ja PEGI. Lomake täytetään perustiedoilla kehittäjästä, ohjelmasta ja ohjelman sisällöstä, joita ovat mm väkivaltaisuus, kiroilu. Perustietojen täytön jälkeen asetetaan oma arvio ikäluokituksesta, joka on voimassa kunnes tarkistetaan pitääkö se paikkansa. Muut organisaatiot vaativat vielä paperisen kaavakkeen täyttämistä ja sen lähettämistä fyysisesti haluttuun osoitteeseen. Microsoft on asettanut vaatimuksen, että jokaiseen maahan johon sovellusta halutaan jakaa tulee kehittäjän olla hankkinut maassa toimivan organisaation luokitus, jotta tämän maan marketplace:ssa saa sovellustaan jakaa.

4.3.1 Entertainment Software Rating Board – ESRB

ESRB -ikäluokitusjärjestelmä on käytössä Pohjois-Amerikassa (USA, Kanada). Se on eräs järjestelmä, joilla on oma internetin välityksellä täytettävä lomake ikäluokitus sertifikaatin hankkiseksi. Lomakkeen täyttämisen jälkeen lähetetään sieltä pdf -tiedostona sertifikaatti ilmoitettuun sähköpostiin sisältäen ikäluokituksen, tiedot sovelluksesta tai pelistä ja sertifikaatti koodin, jotka tarvitaan Dev Centeriin. [11]

ESRB luokittelujärjestelmä koostuu kolmesta osa-alueesta:

- suositeltu ikäluokka (Rating Categories)
- sisällön kuvaus (Content Descriptors)
- interaktiiviset elementit (Interactive Elements)

Näiden lisäksi pelistä tai sovelluksesta löytyy yleensä yhteenveto, josta löytyy vielä tarkemmin kerrottuna pelin tai sovelluksen sisällöstä ja miksi on päädytty kyseiseen ikäluokitukseen. [12]

Ikäluokkia on yhteensä kuusi:

- varhaislapsuus (Early Childhood)
- jokainen (Everyone)

- jokainen 10+ (Everyone 10+)
- nuoret (Teen)
- varttuneemmat (Mature)
- vain aikuiset (Adults only)

Kuvan 7 vasen puolisko kattaa edellä lueteltujen ikäkategorioiden symbolit. Mainittujen kuuden kategorian lisäksi on vielä olemassa RP eli rating pending, joka tarkoittaa että ESRB ei ole vielä asettanut lopullista ikäluokitusta kyseiselle pelille tai sovellukselle. Tätä kategorialaia käytetään promootio tyyppisiin tilaisuuksiin, eli peli tai sovellus on vielä kehitysvaiheessa, mutta tiedetään ESRB -ikäluokituksen tulevan käyttöön. [12]



Kuva 7. ESRB -ikäluokitus. [12]

Sisällön kuvaus kertoo tarkentavasti mistä syistä luokiteltuun kategoriaan on päädytty. Kuvauksesta käy ilmi kielenkäyttöön liittyvät muuttujat, väkivaltaisuus ja vastaavat mikäli rajoittavia syitä on. Kuvan 7 keskimmaisessä osa-alueessa on yhdenlainen sisällön kuvaus. [12]

Interaktiiviset elementit kertovat minkätyyppistä tapahtumaa peliin tai sovellukseen liittyy. Interaktiiviset elementit, kuvan 7 oikean puolimmainen osa-alue, käsittävät tapahtumia kuten mm. käyttäjätietojen jakaminen, sijainnin käyttäminen jne. [11]

4.3.2 Pan European Game Information – PEGI

PEGI -ikäluokitusjärjestelmä on yleisesti käytössä Euroopan alueella, jonka jo nimen perusteella voi arvella. Euroopan lisäksi PEGI -järjestelmä on käytössä

myös Venäjän alueella. Kuten ESRB voi PEGI sertifikaatin hakulomakkeen täyttää internetin välityksellä. Lomakkeen täytön jälkeen niinkään lähetetään sertifikaatti ilmoitettuun sähköpostiin sisältäen tiedot sovelluksesta/pelistä, kehittäjästä, alustasta. Edeltävien lisäksi 4x6 merkistä koostuvan PEGI URN ja ikäluokituksen, jotka tarvitaan sertifikaatin lähetyksessä Dev Centeriin. [11]

PEGI -luokittelujärjestelmä koostuu kahdesta osa-alueesta:

- ikäraajat (Age Levels)
 - kuva 8 oikealla ylhäällä
- sisältösymbolit (Descriptors)
 - kuva 8 alapalkki



Kuva 8. PEGI -ikäluokitus. [13]

Ikäraajat ovat melko suoraviivaisia kertoen suositusrajan, joka yleensä vastaa suoraan miten rankkaa mm. väkivaltaa tai kiroilua sovelluksessa esiintyy. Poikkeuksena ovat toki PEGI 3 ja PEGI 7 merkinnät. PEGI 7 merkinnän ovat saaneet pelottavia ääniä tai kohtauksia sisältävät, jotka muutoin kuuluisivat PEGI 3 merkinnän alle. [13]

Sisältösymboleita on kahdeksan, lueteltuna vasemmalta oikealle kuvasta 8:

- väkivalta (Violence)
- kiroilu (Bad Language)
- kauhu (Fear)
- seksi (Sex)
- huumeet (Drugs)

- syrjintä (Discrimination)
- uhkapeli (Gambling)
- verkkopeli (Online)

Ikäraja on asetettu aina yhteen sopivaksi valittuun, mutta sisältösymboleita peli tai sovellus kohtaisesti saattaa olla useampi. Useampien sisältösymboleiden ansiosta on mahdollista tietää tarkemmin minkä aihealueen takia on päädytty kyseiseen ikärajaan. [13]

5 JULKAISTUN SOVELLUKSEN PÄIVITTÄMINEN

Harvemmin mikään sovellus on sen 1.0 version julkaisun jälkeen täysin valmis. Tämän vuoksi tulisi jo ohjelman suunnitteluvaiheessa ottaa huomioon päivitys mahdollisuudet ohjelmiston kehityksen jatkumisen kannalta. Mikäli tätä ei ota huomioon päätyy kehittäjä opettelemaan päivittämisen ongelmat kantapään kautta.

5.1 Huomioitavia asioita

Julkaistun sovelluksen jatkokehittämisen kannalta tärkeitä huomioita, joita kannattaa käyttää kun kehittää sovellusta tai peliä. Komponentteja joiden mahdolliset lisätoiminnallisuudet on jätetty suunnittelematta, on mahdollista lisätä käyttöön myös jälkikäteen, mutta vaatii silloin yleensä ylimääräistä kikkailua riippuen tilanteesta. Kyseisestä syystä on helpompaa suosia dynaamista sisältöä alueilla, joihin mahdollisia lisätoiminnallisuuksia on luvassa.

5.1.1 Olemassa oleva käyttäjäkunta

Ohjelman ollessa julkaistuna ja ladattavissa kasvattaa se käyttäjäkuntaansa kokoajan. Ohjelman ladanneet ja käytön aloittaneet käyttäjät olettavat pääasiallisesti ohjelman tukevan jatkossa vähintäänkin aikaisempia ominaisuuksia. Pääsääntöisesti päivitysten avulla pyritään aina hiomaan vanhoja ominaisuuksia paremmaksi, sekä tuomaan täysin uusia ominaisuuksia sovellukseen. Mikäli näin ei ole tulee käyttäjiltä useimmiten negatiivista palautetta.

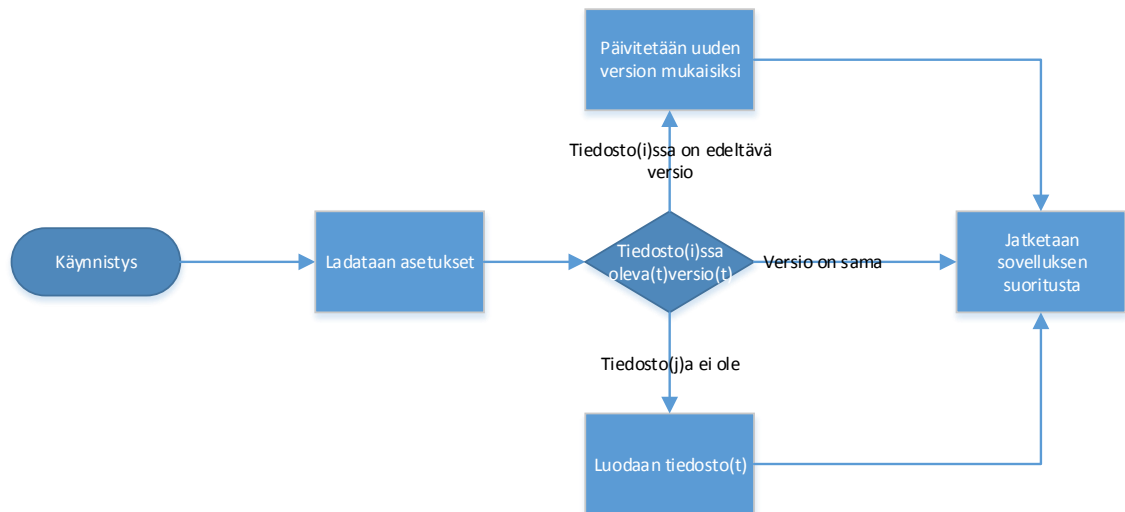
Olemassa olevalta käyttäjäkunnalta voi saada positiivisia vaikutteita, kuten selkeitä ideoita jatkokehittämiseen. Näitä löytää usein sovelluksen arvostelu/palaute sivulta, jolloin näkee myö käyttäjän Windows Live id:n ja pystyy halutessaan kommunikoimaan kommentin jättäneen käyttäjän kanssa

sen kautta. Marketplacen kautta on myös mahdollista ottaa määritelty määrä käyttäjiä beetaversion testaukseen ennen sovelluksen varsinaista julkaisua.

5.1.2 Uudet ominaisuudet

Yhteensopivuusongelmat ovat yleensä välttämättömiä tapauksissa, joissa sovellukseen on tuotu liikaa uusia ominaisuuksia vanhojenkin ollessa vielä epävakaita. Näissä tapauksissa esiintyy usein jatkoseurauksina vanhojen ongelmien moninkertaistuminen ja mahdollisten jo korjattujen ongelmien uudelleen esiintyminen. Yleensä tällaisessa tilanteessa joudutaan eristämään uusi ominaisuus pois ja korjaamaan ensin vanha täysin toimivaan kuntoon. Vasta tämän jälkeen keskitytään uudemman kerran uuden ominaisuuden toiminnallisuuteen ja näiden ominaisuuksien keskinäiseen toiminnallisuuteen.

Ominaisuuksien lisäämisestä syntyviin suurimpiin ongelmiin kuuluvat yleensä rakenteiden muutokset ohjelman kannalta oleellisia tietoja sisältävissä tallennetuissa tiedostoissa. Niitä on helpointa hallinnoida tallentamalla tallennus hetkellä oleva ohjelmaversio käytettävään tietokantaan, XML-tiedostoon tai mikä tallennusmuoto nyt sattuu olemaan käytössä. Tämän avulla on mahdollista tarkistaa ohjelman versionumero käynnistyshetkellä ja verrata tallennetun tiedoston versionumeroon, jonka jälkeen on helpompaa suorittaa muutoksia tallennettavaan dataan kuten kuvassa 9. Mikäli tämä tieto puuttuu tulee tiedoston päivittämisestä huomattavasti hankalampaa.



Kuva 9. Tiedostomuutoksien hallinta.

5.1.3 Sovelluksen tallennustiedostojen tietorakenne

Päivitys ongelman voi ratkaista toisella vähintään yhtä yksinkertaisella tavalla. Nimittäin tekemällä relaatiotietokannan tai XML-tiedoston muuttujanimestä kaksiosaisia, avain (key) ja arvo (value). Tällä tavoin rakennettujen muuttujien avulla voidaan asettaa avaimen paikalle kuvaava muuttujanimi ja arvon kohdalle mitä muuttujan tulisi sisältää. Kuvassa 10 nähdään kuinka kyseinen tapa tallentaa tietoa toimii XML-tiedostomuodossa, vasemmalla puolella struktuurin luonti ja XML-tiedostoon tallennus C# -ohjelmointikielellä, sekä oikealla puolella luotu XML-tiedosto avattuna internetselaimella. Käyttämällä tätä kyseistä tallennus tapaa vältetään xml -tiedostojen ongelmat XmlSerializer -luokan Serialize -metodin tekemän XML-tiedoston parsimisessa takaisin objektiksi Deserialize -metodilla, kun ei tiedetä tarkkaa muotoa (luokka tai struktuuri) johon parsia löydetyt arvot. Tallennettavien arvojen lisääminen onnistuu myös kätevästi vain lisäämällä uusi muuttuja List<Data> -objektiin, joka on samassa muodossa kuin kuvassa 10. Relaatiotietokannassa avain (key) + arvo (value) merkintätapa on melko yleinen, sillä tietokannan taulut itsessään nimetään kuvastamaan riippuvuutta.

```

[Serializable]
public struct Data
{
    public string Key;
    public string Value;
}

class Program
{
    static void Main(string[] args)
    {
        List<Data> SaveData = new List<Data>();

        SaveData.Add(new Data() { Key = "Language", Value = "en" });
        SaveData.Add(new Data() { Key = "Version", Value = "0.1.0.1" });

        Stream saveStream = File.Create("Data.xml");

        XmlSerializer saveXML = new XmlSerializer(SaveData.GetType());
        saveXML.Serialize(saveStream, SaveData);
        saveStream.Close();
    }
}

```

```

<?xml version="1.0"?>
- <ArrayOfData
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Data>
    <Key>Language</Key>
    <Value>en</Value>
  </Data>
  - <Data>
    <Key>Version</Key>
    <Value>0.1.0.1</Value>
  </Data>
</ArrayOfData>

```

Kuva 10. XML-tiedoston tietorakenteen luonti.

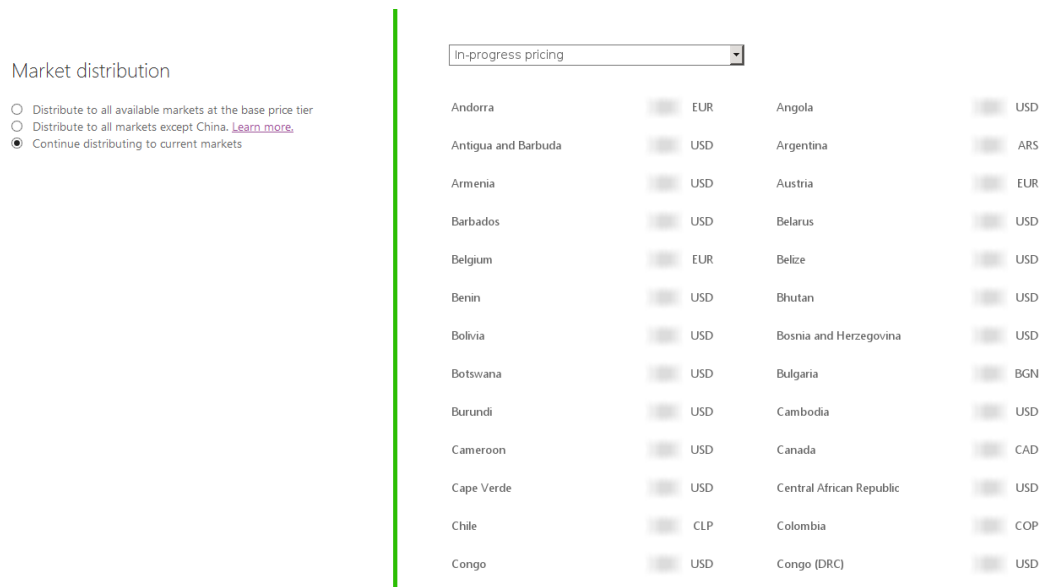
Tallennustiedostojen data muutokset tulee aina ottaa huomioon koodissa, sekä testata päivityksen huomioivan koodin toimivuus kattavasti, sekä unit testeillä ja käytännön testauksella. Esimerkiksi XML tiedostomuodossa olevan datan ladattava objekti muuttuu aina, mikäli siihen lisätään vain muutamakin uusi muuttuja. Yritettäessä ladata dataa väärässä muodossa tapahtuu Exception, joka huomioituna tuottaa jonkinlaisen ilmoituksen käyttäjälle tai merkinnän lokiin, mutta huomioimattomana saattaa jopa pahimmassa tapauksessa kaataa koko ohjelman.

5.1.4 Marketplace alueisiin kohdentaminen

Jo suunnitteluvaiheessa tulisi kohdentaa sovelluksen pääsääntöinen käyttäjäryhmä ja jakelumaat. Kohdentamalla tiettyihin maihin saadaan suoraan selvitettyä, millaisia sertifikaatteja kyseisillä alueilla vaaditaan, jotta pelin tai sovelluksen jakelu Marketplacesissa on sallittua. Kohdennettu alue tulee ottaa huomioon aina sovellusta päivitettäessä, koska muutoin myös ikäluokittelu saattaa muuttua. Ikälohkoon kohdentamista kannattaa miettiä jo suunniteltaessa mm. kielenkäytön ja väkivallan suhteen, sillä ne vaikuttavat suoraan siihen mihin ikäluokkaan pelin tai sovelluksen kohdentaminen sertifikaattien suhteen sijoittuu. Varsinkin kielenkäytön osalta voi hyvin helpolla

epäonnistu mikäli sovellukseen lisätään esimerkiksi yksi uusi string sellaiseen tilanteeseen, jossa se ei tule vastaan kovin usein jälkeinpäin ja saattaa näin ollen jäädä kummittelemaan sovellukseen.

Kuvan 11 vasemmalla puolella on päivityksen mahdollistamat vaihtoehdot maihin kohdentamisesta ja oikealla puolella parhaillaan käytössä olevat jakelumaat, sekä niiden hinnoittelu (sumennettu). Lähetettäessä täysin uutta sovellusta on mahdollista valita erikseen, mihin maihin haluaa sovellusta jakaa.



Kuva 11. Jakelumaat, Windows Phone Marketplace.

5.1.5 Versionhallinta projektissa

Versionhallinta on erittäin keskeisessä roolissa ominaisuuksien lisääntyessä ja ongelmien alkaessa esiintyä, eikä ole varmuutta siitä minkä muutoksen kautta kyseiset ongelmat ovat syntyneet. Versionhallintaohjelmilla on mahdollista palauttaa tilanne ennen kuin ongelmia, kuten kaatumisia tai ns. ei haluttuja ominaisuuksia, alkaa esiintyä. Varteenotettavia vaihtoehtoja ja yleisimmin käytettyjä ovat Subversion (SVN) ja Git-versionhallintaohjelmistot.

Kehittäjän tarvitsee vain muistaa suorittaa commit-komento versionhallinta clientissa lähettääkseen tekemänsä muutokset repositorioon saatuaan uuden

ominaisuuden tai osakokonaisuuden tehtyä. Tällöin siitä jää merkintä repositorioon ja projekti on mahdollista palauttaa jatkossa tähän tilaan.

Mikäli ominaisuuden toiminnallisuutta halutaan optimoida koodia esimerkiksi muistinkäytön kannalta ja se rikkoo jonkun muun osan koodista tai optimointi koodi ei toimi ylipäänsä on yleensä parempi suorittaa revert-komento versionhallinta-clientissa. Näin päästään aloittamaan ns. puhtaalta pöydältä halutun repositorioversion kohdalta ilman minkäänlaista vauriota kokonaisuuden toiminnallisuuteen.

5.1.6 Testaus

Testausta kannattaa suorittaa sitä myöden kun sovelluksen uusia ominaisuuksia saadaan valmiiksi, jotta saadaan varmistettua ominaisuuden toiminnallisuus. Kehityksen ohessa tehdään toki pienimuotoista testausta, jotta varmistetaan ominaisuuden toiminnallisuuden toteutuminen.

Yksikkötestaus (Unit testing) on usein suureksi avuksi kun joudutaan tekemään muutoksia aikaisempaa koodia koskien. Yksikkötestit (Unit test) kirjoitetaan ominaisuuksien toteutuksen ohessa ja ajetaan useimmiten rakentamisen (build) yhteydessä. Tällöin huomataan välittömästi tapaus, jossa luokan tai metodin toiminnallisuus ei täytä enää sille asetettuja tavoitteita, eli yksikkötesti menee rikki (fail). Yleensä objekti-orientoituneessa ohjelmointikielessä testataan yksikkötesteillä (Unit tests) vähintään konstruktorin (constructor) ja destruktorin (destructor) toiminta.

Kannattaa ottaa tavaksi tietyn commit-määrän täytyessä suorittaa täysimääräinen integraatiotestaus. Integraatiotestauksella saadaan kiinni rajapintojen (API) ja moduulien välillä tapahtuvat virhetilanteet, jotka usein aiheuttavat ohjelman kaatumisen.

Lopuksi ominaisuuksien alkaessa olla kasassa aloitetaan järjestelmätestaus. Järjestelmätestauksessa pyritään suorittamaan mahdollisimman kattavasti koko sovelluksen tai pelin toiminnallisuus. Ohessa pyritään optimoimaan toimintoja

parhaan mukaan. Järjestelmätestauksella pyritään löytämään viimeisetkin mahdolliset ongelmakohdat ennen julkaisua. [14]

Vaihtoehtona ennen varsinaista julkaisua Dev Center sisältää mahdollisuuden jakaa betatestaus oikeuksia kehittäjän asettamalle määrälle latauksia. Tällöin voi saada vielä käyttäjiltä tietoa mahdollisista kriittisistä bugeista, sekä kattavampaa ja ei niin johdonmukaista testausta ennen varsinaista julkaisua. Dev Centerin kautta järjestetty betatestaus toimii luomalla xap-tiedoston lähettämisen ohessa käyttöoikeus lista (Access Control List), joka antaa listassa oleville mahdollisuuden osallistua sovelluksen betatestaukseen. Betatestauksen saa päätettyä lähettämällä lopullisen julkaisun (release) Dev Centeriin.

5.1.7 Sertifiointi ja Microsoftin vaatimusten täyttö

Microsoftin Markeplace sertifiointiin on asetettu liuta erilaisia vaatimuksia, jotka sovelluksen tulee täyttää ennen kuin on mahdollista saada sovellus julkaistua Windows Phone Marketplacessa. Suuri osa vaatimuksista on mahdollista silmäillä yleismääräisesti läpi, perehtymättä sen syvällisemmin niihin, joka helpottaa kehittäjän työtä huomattavasti.

Windows Phone Marketplace sertifiointiin tarvittavat vaatimukset on jaoteltu viiteen eri osa-alueeseen:

- sovelluskäytännöt (App policies for Windows Phone)
- sisältökäytännöt (Content policies for Windows Phone)
- sovelluksenlähetysvaatimukset (App submission requirements for Windows Phone)
- teknisetsertifiointivaatimukset (Technical certification requirements for Windows Phone)
- muut vaatimukset tietyn tyyppisille sovelluksille (Additional requirements for specific app types for Windows Phone) [15]

Microsoft suorittaa sertifiointitestauksen ennen sovelluksen julkaisua Windows Phone Marketplaceen. Testaus suoritetaan usealla WP-puhelimella, jotta sovelluksen yhteensopivuus saadaan samalla testattua. Testaus kestää parista

päivästä hieman yli viikkoon, riippuen siitä kuinka paljon sovelluksia julkaisujonossa on. Testauksen päätteeksi sovellus joko läpäisee vaatimusten asettamat testit tai ei.

Sovelluksen läpäistessä sertifiointitestit saa kehittäjä viestin ilmoittamaansa sähköpostiin, joka informoi sovelluksen julkaisu valmiudesta. Riippuen valinnasta, jonka kehittäjä teki lähettäessään sovellusta sertifiotavaksi, sovellus laitetaan joko suoraan Marketplaceen testien läpäisyn jälkeen tai jätetään Dev Centeriin odottamaan, että kehittäjä julkaisee sen manuaalisesti Dev Centerin kautta. Microsoft ilmoittaa pyrkivänsä saamaan sertifiointitestauksen suoritettua 5–7 työpäivän aikana. Julkaisun tapahtuessa manuaalisesti tai automaattisesti kestää noin 24 tuntia ennen sovelluksen ilmaantumista Marketplaceen. [16]

Mikäli sovellus ei läpäise sertifiointitestejä, saa kehittäjä pdf-tiedoston sisältävän sähköpostin, josta ilmenee tarkemmin mikä on epäonnistumisen syy. Pdf-tiedosto sisältää selvät toisto-ohjeet (steps to reproduce), jos kyseessä on kaatuminen tai muunlainen toistuva bugi. Vastaava selostus tulee mikäli sovellus ei täytä jotakin asetettua vaatimusta toiminnallisuuden kannalta. Tällöin tulee korjata bugi tai toiminnallisuus vastaamaan vaatimuksia ja lähettää uudemman kerran sertifiotavaksi.

5.2 Miksi halutaan päivittää?

Päivityksiä ilmaantuu erinäisinä sykleinä, joiden perusteella voi päätellä minkä tyyppisestä päivityksestä on kyse. Lyhyellä aikavälillä ja usein ilmaantuvat päivitykset vihjaavat bugeista, joita on päässyt julkaistuun versioon ja edellisen julkaisun bugien korjaus yrityksistä. Päivitysten ilmaantuessa pidemmällä aikavälillä toisistaan on kyseessä yleensä uuden ominaisuuden implementointi eli yleiskielellä suurempi päivitys.

Sovellukseen julkaistaan päivityksiä erinäisten tilanteiden takia, joista yleisimpiä ovat seuraavat:

- uudet ominaisuudet
- optimointi
- bugikorjaukset
- graafiset muutokset

Kaikkein yleisimpiä luetelluista päivittämisen syistä ovat bugikorjaukset ja uusien ominaisuuksien kehittäminen sovellukseen. Optimointi on syytä hoitaa suurempien päivitysten, kuten uusien ominaisuuksien implementoinnin välissä, jotta saadaan valmiiksi toimivat ominaisuudet toimimaan mahdollisimman tehokkaasti ja vakaasti ennen uusien ominaisuuksien tuomia uusia optimointitarpeita.

6 KEY ZONE -PÄIVITYS

Key Zone on XNA framework-pohjainen Windows Phone -peli, jossa ruudulle ilmestyy erinäisiä kuvioita, joita tulee joko ampua tai nille tulee tehdä jotakin muuta riippuen siitä, mikä kuvio on kyseessä. Valittavissa on joko vaikea (hardmode) tai normaali (normalmode) vaikeustaso. Vaikeustasojen erona on mahdollisten ruutujen määrä, joihin kuvioita ilmaantuu, ja nopeus jolla kuvioita ilmaantuu.

Pelissä käytettäviä kuvioita on mahdollista nähdä kuvassa 12 aivan oikean puoleisimmassa kuvankaappauksessa. Pelin käyttöliittymästä on pyritty tekemään mahdollisimman interaktiivinen esimerkiksi huomioimalla käyttäjän asetuksista valitsema ase jopa menun valinnoissa. Kaikki menun otsikot hyödyntävät toisaalta käyttäjän Windows Phone -mobiilikäyttöjärjestelmän käyttämää teemaväriä, kuten kuvassa 12 näkyvissä kolmessa ensimmäisessä kuvankaappauksessa on nähtävissä.

Windows Phone

Hae sovelluksia ja ohjeita


Puhelimet Ominaisuudet Sovellukset + pelit Ohjeet

Selaa
Oma puhelin

Valokeilassa Sovellukset Pelit Ostohistoria

pelit / ongelmanratkaisu- ja tietopelit

Key Zone



Ilmainen ★★★★★
1 arvostelua

asenna


Tykkää 10

Julkaisija
HumusFoodz

Key Zone, a simple and addictive game to play on the move, without focusing too much on it!

The game has 12 possible squares where the figures might appear, but only 9 of them are in use if you're not playing in hardmode. Figures will start appearing pretty fast after you start the game, which you'll have to decide either to hit or not to (check help for info on how each figure behaves). Once you have all of the squares as flags you'll move on to the next level.

Off course there is an highscore table as well, that keeps up to 10 best scores on your phone.



Arvostelut

Ei arvosteluita

Kuva 12. Key Zone 1.0 Windows Phone Marketplace.

Pelin lähdekoodi on kirjoitettu käyttäen C# -ohjelmointikieltä. Tallennettavat tiedot tallennetaan XML-tiedostoihin. Tallennettaviin tietoihin kuuluvat mm. asetukset ja parhaat tulokset jotka pelissä on saavutettu.

Key Zone ikäluokitus asettui PEGI -luokituksen mukaisesti PEGI 3+ ja ESRB -luokituksen mukaisesti Everyone -luokkaan. Suurimmat syyt miksi luokitus asettui kyseisiin lohkoihin ovat, koska siinä ei esiinny verta saati sitten kiroilua. Kumpaakin luokitusta koskien tuli hakea sertifikaatti internetin välityksellä täytettävällä lomakkeella, kuten 4.3.1 ja 4.3.2 luvuissa opastetaan.

6.1.1 Key Zone 1.0 versio

Koko pelin tärkeimpiä elementtejä on ruudunhallinta (screenmanagement), koska ruutuja on useita erilaisia, joista jokainen omaa oman käyttötarkoituksen. Key Zone ruudunhallinnassa hyödynnetään polymorphismia. Polymorphismin ansiosta voidaan laittaa täysin eri sisällöllisiä ruutuja (screens) pinoon (stack), esimerkiksi alimpana pinossa (stack) on menu ruutu (menu screen) jonka päälle asetetaan itse peli ruudun (game screen) piirto, mutta vain ylimpänä stack:ssa oleva ruutu (screen) piirretään. Tämä ei tarkoita etteikö kaikkia ruutuja voisi päivittää samalla kertaa, osa ruuduista ei vain piirry käyttöliittymään (User interface). Tämä on mahdollista koska polymorfismissa on olemassa ylempi luokka, josta sen johdetut alaluokat perivät pinottaville ruuduille tyypilliset ominaisuudet. Tästä voidaan nähdä esimerkkinä kuvassa 13 esiintyvä AboutScreen -luokka (vasemmalla kuvassa), joka perii MenuScreen -luokan ominaisuudet, joka taas perii GameScreen -luokan ominaisuudet (kuva 13 oikealla). Näin voidaan pinota hyvinkin paljon toisistaan poikkeavia objekteja. Periytyvien ominaisuuksien ansiosta on mahdollista pinota ruutuja, vaikka sisältö ei vastaa toisiaan kuten esimerkiksi integer -tyypin muuttujista koostuva pino.

```
class AboutScreen : MenuScreen
{
    Variables
    public AboutScreen()
        : base("ABOUT.")
    {
    }
    public override void LoadContent()
    {
        ContentManager content = ScreenManager.Game.Content;
        #region Load Fonts + AccentColor
        ThemeColor = WindowsPhoneTheme.getPhoneAccentColour();
        TitleFont = ScreenManager.BoxFont;
        OtherFont = ScreenManager.Font;
        #endregion
    }
}
```

```
/// <summary>
/// Base class for screens that contain a menu of options. The user can
/// move up and down to select an entry, or cancel to back out of the screen.
/// </summary>
abstract class MenuScreen : GameScreen
{
    #region Fields
    // the number of pixels to pad above and below menu entries for touch input
    const int menuEntryPadding = 5;
    List<MenuEntry> menuEntries = new List<MenuEntry>();
    int selectedEntry = 0;
    string menuTitle;
}
```

Kuva 13. Polymorphism.

ScreenManagerin pohjaratkaisuksi on otettu MS-PL (Microsoft Public License) lisenssillä tehty Game State Manager (http://xbox.create.msdn.com/en-US/education/catalog/sample/game_state_management). Lisenssi on määritelty Open Source Initiativen toimesta avoimeksi lähdekoodiksi (open

source). MS-PL -lisenssin vaatimuksena on kyseisen lisenssin liittäminen jaettaessa ohjelmistoa ja alkuperäisten author merkintöjen säilytys, mutta antaa kehittäjälle oikeudet käyttää sovellettuja versioita tästä. [17]

Key Zonessa on käytössä Microsoftin Pubcenter -mainonta API, jonka avulla on mahdollista saada tuloja pelistä. Siihen löytyy valmis pohja MSDN sivustolta, jota voi huoletta käyttää suoraan sovelluksessaan, jossa on oletuksena käytössä Microsoft testi mainos. Ennen julkaisua on syytä muistaa vaihtaa addId ja applicationId muuttujien arvot omiksi Pubcenterista löytyviin ennen julkaisua, jotta saadaan omat Pubcenterissa konfiguroidut mainokset käyttöön, joista kerrotaan luvussa 3.2.

Eräs huomattavimpimmista bugeista 1.0 versiossa on ruudulla näytettävien pisteiden nollaantuminen kentän vaihtumisen jälkeen, koska pisteet lasketaan kenttäkohtaisesti mahdollisten bonuspisteiden vuoksi ja lisätään vasta kentän vaihtumisen aikana kokonaispistemäärään. Bugi johtuu yksinkertaisesti väärän muuttujan käyttämisestä esitettäessä parhaillaan olevien pisteiden määrää.

Pelistä jäi puuttumaan äänet totaalisesti, sillä ääniefektien (sfx) lisäksi oli tarvetta myös taustamusiikille. Ääniefektit ja taustamusiikin toiminnallisuus oli käytännössä jo koodattuna sekä implementoituna Key Zone 1.0 lähdekoodissa, mutta eivät käytössä. Äänet jäivät kokonaisuudessaan pois 1.0 version julkaisusta lähinnä vain taustamusiikin puuttumisen vuoksi, joka puolestaan johtui sopivalla tekijänoikeus lisenssillä varustetun kappaleen löytymättömyydestä.

Key Zonessa on mietitty päivitysvalmiutta muuttujien objekti rakenteen osalta, jolla tarkoitetaan toisiaan vastaavia objekteja eri arvoilla. Tällaisia ovat mm. alueet joihin kuvioita voi ilmaantua ja itse kuviot joita alueisiin voi ilmaantua. Kuvio objektit on laitettu C# spesifiseen List<> -tyypin muuttujaan, joka nimensä mukaisesti on lista. Näin uuden kuvion tekeminen ei vaadi lähdekoodin osalta kuin uuden objektin lisäämisen listan jatkoksi. Tämänlainen menettely helpottaa huomattavasti kehittäjän työtä kun lisätään kyseisiä kuvio objekteja.

6.1.2 Key Zone 1.1 versio

Suuri osasy s päivittämiseen on latausmäärien hiipuminen (katso kuva 2, luvussa 3.1). Julkaisun jälkeisen käyttäjien alkuinnostuksen jälkeen latauskäyrä laski yli kymmenen latauksen päivä tahdilla. Alussa tapahtuneista latausmääristä voidaan päätellä WP puhelimen omistavien käyttäjien olevan kiinnostuneita pelin ideaa kohtaan, mutta toteutuksen olevan huono, koska latausmäärät eivät pysyneet korkeina.

Key Zone 1.0 käytti vielä .NET 4.0 kirjastoja sillä .NET 4.5 ei ollut vielä saatavilla kun Key Zone julkaistiin 26.10.2012. Kirjastoversio vaihtuu uusimman päivityksen myötä.

Key Zone 1.0 julkaisun jälkeen alettiin vasta fokusoimaan pääpainoa musiikin ja äänien hankkimiselle. Taustamusiikin ja ylipäänsä äänien takia tarvitsi tutustua myös tekijänoikeussäntöjä koskeviin seikkoihin musiikin saralla. Tekijänoikeussäntöihin perehtymisen jälkeen koitettiin etsiä muutaman viikon verran sopivaa musiikkia peliin sopivalla tekijänoikeus lisenssillä löytämättä sellaista edelleenkään. Painittaessa taustamusiikki ongelman kanssa mainittiin asiasta erälle DJ:lle, joka innostui ideasta, että saisi musiikkiaan maailmanlaajuiseen levitykseen ja samplasi taustamusiikin peliin. Taustamusiikki ongelman ratketessa tarvitsi enää vain hankkia ääniefektit, joka oli jo huomattavasti helpompaa.

Ääniefektit lisättiin pelissä oleville eri asetyypeille, joita ovat haulikko ja pistooli, eli samat kuin 1.0 versiosta löytyvät kaksi asetusta. Näiden lisäksi laitettiin pommin räjähtämiseen oma ääniefektinsä tehostamaan aikaisempaa välkähtely efektiä. Kaikkien ääniefektien soittamiseen käytetään SoundEffectInstance -luokkaa, jolla saadaan luotua jokaisesta uudesta soitettavasta äänestä oma instanssinsa. Käyttämällä omaa instanssia jokaiselle soitetulle efektille vältetään bugimaiselta kuulostava äänen katkaisu, jos yritettäisiin soittaa uutta ääniefektiä samassa instanssissa.

Ääniefektien ja taustamusiikin lisäämisen lisäksi tuli lisätä asetuksiin mahdollisuus poistaa kumpikin käytöstä, mikäli käyttäjä ei halua ääniä. Asetuksien lisääminen toi muutoksia XML-tiedostoon, koska ei haluta hukata käyttäjän aiemmin valitsemia asetuksia. Asetuksien tallentaminen onnistui kivuttomasti, koska XML-tiedoston tietorakenteen päivittämisen mahdollisuus oli huomioituna jo aiemmassa 1.0 versiossa.

Eräs näkyvämmistä päivityksistä on Key Zone 1.1 version käyttämä Live Tile, eli Windows Phonen "Metro" -käyttöliittymä spesifinen laatikko, jonka kääntöpuolella voidaan esittää tietoa. Tämä toteutettiin käyttämättä BackgroundWorker -luokkaa, koska ei haluta kuluttaa puhelimen akkua ei niin relevanttiin tekemiseen. Live Tile tehtiin esittämään vain parhain saavutettu tulos, joka asetetaan päivittymään aina kun pelaaja sellaisen onnistuu saavuttamaan. Paras tulos ei muutu mikäli pelaaja ei ole peliä välissä pelannut, joten olisi turhaa pyörittää taustalla akkua syövää BackgroundWorker -luokkaa. Eli nykyisellään Key Zonen Live Tile kääntyy kovakoodatun (hardcode) ajan välein ja esittää parhaan puhelimesta saavutetun tuloksen sekä pelaajanimen.

Ensimmäisen Key Zonen julkaisun jälkeen on ilmaantunut 48 uutta maata WP Marketplaceen. Tämä kävi ilmi kun päivittäessä valittiin sovellukselle halutut jakelumaat. Jakelumaista on poisluettuna Kiina ja Brasilia johtuen maissa käytettävistä ikäluokitusjärjestelmistä, joihin ei jaksettu hakea sertifikaatteja tämän päivityksen ohessa.

Ennen 1.1 version lähettämistä huomattiin vielä viimehetkellä bugi, joka ilmaantui aina kentän vaihdon yhteydessä. Se vaikutti pisteiden laskentaan huomattavasti ja on ollut ilmeisesti 1.0 versiosta lähtien olemassa, mutta päässyt luistamaan testeistä. Bugi oli vastaavanlainen kuin esitettyjen pisteiden kanssa eli käytetty yksinkertaisesti väärää muuttujaa operaatiossa. Pisteet oli tarkoitus laskea käyttäen kuvan 14 vasemmalla puolella näkyvää kaavaa, mutta muuttujat Score ja TotalPoints olivat menneet sekaisin kuten kuvassa 14 voidaan huomata. Kuvassa 14 vasemmalla puolella keltaisella merkittynä 1.1 versioon korjattu muuttuja ja oikealla 1.0 versiossa käytetty väärä muuttuja.

Muuttujien sekoittaminen keskenään aiheutti huomattavan suuria tuloksia jo muutaman kentän pelaamisen jälkeen.

```
/// <summary>
/// Values to be added before proceeding with value nulling / saving to the highscores
/// </summary>
public void CountTotalPoints()
{
    _TotalPoints = _Evades * _Score;
}

/// <summary>
/// Values to be added before proceeding with value nulling / saving to the highscores
/// </summary>
public void CountTotalPoints()
{
    _TotalPoints = _Evades * _TotalPoints;
}
```

Kuva 14. Key Zone, pisteiden lasku -bugi.

Peliin tehty päivitys meni Microsoftin sertifiointi prosessista läpi ensi yrittämällä ja vielä luvatus viiden päivän prosessointi ajan sisällä. Sertifioinnin jälkeen päivitys päätettiin julkaista heti ja kestitkin vain kaksi tuntia, jonka jälkeen päivitys olikin jo nähtävillä Windows Phone Marketplacessa, kuten kuvassa 15 nähdään. Itse puhelimelle ilmoitus päivityksestä ilmaantui vasta noin päivää myöhemmin.

Windows Phone

Hae sovelluksia ja ohjeita


Puhelimet Ominaisuudet Sovellukset + pelit Ohjeet

Selaa
Oma puhelin

Valokeilassa Sovellukset Pelit Ostohistoria

pelit / ongelmanratkaisu- ja tietopelit

Key Zone



NOW WITH LIVE TILE & SOUNDS!

Key Zone, a simple and addictive game to play on the move, without focusing too much on it!

The game has 12 possible squares where the figures might appear, but only 9 of them are in use if you're not playing in hardmode. Figures will start appearing pretty fast after you start the game, which you'll have to decide either to hit or not to (check help for info on how each figure behaves). Once you have all of the squares as flags you'll move on to the next level.

[näytä tiedot](#)

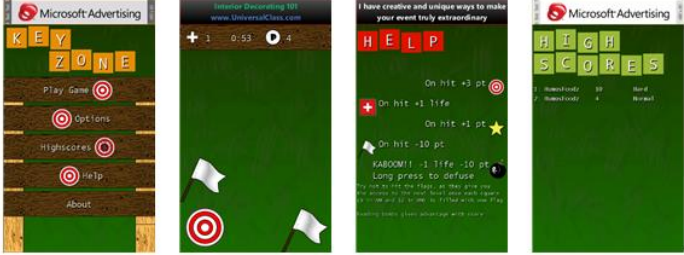
Ilmainen ★★★★★
1 arviolua

asenna

Tykkää 10

Julkaisija
HumusFoodz

Arvostelut



Kuva 15. Key Zone 1.1 Windows Phone Marketplace.

6.1.3 Key Zone jatkokehittäminen

Jatkossa on tiedossa ainakin pelissä esiintyvien kuvioiden eli pääosiltaan graafisia lisäyksiä, jotta saadaan käyttäjille lisää mielenkiintoa peliä kohtaan. Kuvioiden lisäys olisi peliin tuomisen osalta hyvin helppoa, sillä se vaatisi vain kuvio objektien luomisen ja lisäyksen kuvio listaan. Tämä toisi lisää tapahtuma mahdollisuuksia. Mietteissä on ainakin seuraavanlaisia kuvioita:

- kypärä
- kello

Kyseisillä kuvioilla olisi toki täysin uusia ominaisuuksia, joita niillä voi tuoda mukaan peliin. Kypärä toimisi suojana pommia vastaan, jolloin sen saamiseen täytyisi olla vaatimuksena ettei pelaaja ole saanut kypärää vielä entuudestaan. Kello puolestaan hidastaisi aikaa mahdollistaen pisteiden keräämisen tai pommin purkamisen helpottamisen.

Suurin remontti tulisi tehdä kaikkiin ajastimiin, joita pelissä on käytössä useampia. Key Zonen 1.0 ja 1.1 -versiot käyttävät ajastimia, jotka toimivat XNA spesifisen GameTime -muuttujan avulla, eli toisinsanoen synkronisesti. Ajastimet tulisi muuttaa toimimaan asynkronisesti, jotta saataisiin peli pyörinään astetta sujuvammin. Nykyisellään (Key Zone 1.1) voidaan huomata synkronisuus kun uutta kuviota aletaan hakemaan tietyin väliajoin ja toiminta pysähtyy satunnaisgeneraattorin palauttaessa kuvioita, jonka kenttäkohtainen raja on jo ylittynyt. Rajan ylitys ongelma ratkaistaan yksinkertaisesti pyörittämällä samaa kuvion haku looppia kunnes palautetaan kuvio, jonka osalta kentälle asetettu raja ei ole vielä ylittynyt.

Jatkamis (Continue) toiminnallisuuden lisäys on toinen suurempi mietinnässä oleva ominaisuus. Helpoimmillaan sen voisi toteuttaa suorittamalla Serialize -metodi peliruudulle (GameScreen), jolloin saataisiin talteen kyseisen ruudun kaikkien muuttujien sen hetkiset tilat. Tosin tämä vaatisi myös Deserialize -metodin käyttämistä haluttaessa jatkaa tallennetusta tilanteesta. Logiikka uudistuksia mm. uuden pelin aloittamiseksi vaikka on olemassa jatkettavissa oleva peli. Logiikkaa vaadittaisiin myös päinvastaiseen tilanteeseen, eli kun ei ole olemassa peliä jota olisi mahdollista jatkaa, jolloin jatkamiseen tarkoitettu nappula tulisi kytkeä pois päältä (disable).

7 YHTEENVETO

Opinnäytetyön tarkoituksena oli perehtyä sovelluksen päivittämisen tuomiin ongelmiin ja kuinka päivittämiseen liittyviä ongelmia parhaiten vältetään. Alustaksi päädyttiin valitsemaan Windows Phone, koska viimeisin julkaisemani sovellus oli tehty sille ja sovelluksessa oli tiedossa olevia puutteita. Puutteiden vuoksi oli tarpeen saada aikaiseksi päivitys sovellukseen.

Suurin ongelmani liittyi opiskelijaksi tunnistautumiseen Dev Centerissa, jonka olin tehnyt jo kertaalleen vuoden 2012 puolella. Se ei onnistunut lisenssin mennessä umpeen, vaikka samaisen Dev Center -sivuston mukaan opiskelijuus tulisi vain varmentaa vuoden välein. Kuukausi lisenssin umpeutumisen jälkeen poistetaan sovellukset Marketplacesta ja sovelluksen päivittäminen olisi ollut mahdotonta, joten päädyin maksamaan 75 euroa kehittäjälisenssistäni.

Tehdessäni päivitystä, joka menisi Windows Phone Marketplaceen, ei tarvinnut miettiä yhteensopivuusongelmia XML-tiedostojen tiedostorakenteiden suhteen. Tämä oli mahdollista, koska ladattaessa tallennettuja tietoja tiedostoista tarkistetaan tiedoston tallennusajankohtana ollut versio ja mikäli se ei täsmää nykyiseen, tehdään muutokset XML-tiedostoihin.

Päivitystapahtuma onnistui hyvin Key Zonen osalta johtuen suurimmilta osin juuri päivittämisen huomioidin ansiosta, kun peliä alkujaan suunniteltiin. Pyrittiin saamaan päivitys mahdollisimman vaivattomaksi ja juuri tähän tulokseen päästiin. Ainoastaan ajastimien uusiminen olisi haluttu toteuttaa lisänä tähän päivitykseen, mutta se olisi kasvattanut koodaukseen kuluvaan aikaa huomattavasti johtuen ajastimien käytöstä useassa eri objektissa. Objektien luokkarakenne olisi tarvinnut suunnitella täysin uudelleen parhaan tuloksen saavuttamiseksi. Testauksen määrä olisi kasvanut myös huomattavasti, koska laajan vaikutusalueen vuoksi olisi tarvinnut testata melkein kaikki pelin toiminnallisuudet uudestaan.

LÄHTEET

- [1] [www-dokumentti] Windows Phone, Wikipedia, viitattu 13.04.2013, http://en.wikipedia.org/wiki/Windows_Phone
- [2] [www-kuva] Windows family tree, Wikipedia, viitattu 13.04.2013, http://en.wikipedia.org/wiki/File:Windows_Updated_Family_Tree.png
- [3] [www-dokumentti] Native code on Windows Phone 8, MSDN, viitattu 15.04.2013, [http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj681687\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj681687(v=vs.105).aspx)
- [4] [www-dokumentti] Cross Platform, open source .NET development framework, Project Mono, viitattu 13.04.2013, http://www.mono-project.com/Main_Page
- [5] [www-dokumentti] What's new in the .NET Framework 4.5, MSDN, viitattu 14.04.2013, <http://msdn.microsoft.com/library/vstudio/ms171868>
- [6] [pdf-dokumentti] Maaranen Timo, XNA-peliohjelmointiympäristö, Helsingin yliopisto, 2008, viitattu 22.03.2013, http://www.cs.helsinki.fi/u/vihavain/k08/sem/korjatut/maaranen_timo_XNA_peliohjelmointiymparisto.pdf
- [7] [www-dokumentti] Microsoft Xna, Wikipedia, viitattu 20.04.2013, http://en.wikipedia.org/wiki/Microsoft_XNA
- [8] Lecrenski, N., Holland D., Sanders A., Ashley K., Professional Windows 8 Programming; Application development with C# and XAML, John Wiley & Sons, Inc, 2013
- [9] [www-kuva] Integrating the AdControl using the API, MSDN, viitattu 07.05.2013, [http://msdn.microsoft.com/en-us/library/advertising-mobile-windows-phone-8-adcontrol-walkthrough-api\(v=msads.20\).aspx](http://msdn.microsoft.com/en-us/library/advertising-mobile-windows-phone-8-adcontrol-walkthrough-api(v=msads.20).aspx)
- [10] [www-dokumentti] Pubcenter glossary, Pubcenter, viitattu 10.05.2013, http://help.bingads.microsoft.com/Help.aspx?market=en-US&project=adCenter_Pub_RTW_ss&querytype=topic&query=PUB_REF_Glossary.htm
- [11] [www-dokumentti] Game ratings, MSDN, viitattu 19.04.2013, [http://msdn.microsoft.com/en-us/library/windowsphone/help/jj206734\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/help/jj206734(v=vs.105).aspx)
- [12] [www-dokumentti] Entertainment Software Rating Board rating system, ESRB, viitattu 09.04.2013, <http://www.esrb.org/index-js.jsp>
- [13] [www-dokumentti] Pan European Game information ikäluokitusmerkinnät, PEGI, viitattu 14.04.2013, <http://www.pegi.info/fi/index/id/201/>
- [14] [www-dokumentti] Software testing, Software testing help, viitattu 26.04.2013, <http://www.softwaretestinghelp.com/types-of-software-testing/>
- [15] [www-dokumentti] App certification requirements for Windows Phone, MSDN, viitattu 24.04.2013, [http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184843\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184843(v=vs.105).aspx)
- [16] [www-dokumentti] Understanding app and in-app product submission, MSDN, viitattu 20.05.2013, [http://msdn.microsoft.com/en-us/library/windowsphone/help/jj206729\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/help/jj206729(v=vs.105).aspx)

[17] [www-dokumentti] MS-PL lisenssi, Open Source Initiative, viitattu 04.05.2013, <http://opensource.org/licenses/MS-PL>