

Mobiilipelin suunnitleminen Android-alustalla yritykselle X

Jouni Pärssinen

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2012



Tekijä tai tekijät Jouni Pärssinen	Ryhmätunnus tai aloitusvuosi TIKO06S
Raportin nimi MOBIILIPELIN SUUNNITTELEMINEEN ANDROID-ALUSTALLA YRITYKSELLE X	Sivu- ja liitesivumäärä 50+41
Opettajat tai ohjaajat Juhani Välimäki	
<p>Mobiilipelaamisen lisääntyvä kasvu tarjoaa kiinnostavan markkina-alueen yrityksille. Yritys X on muutaman vuoden ikäinen mobiilisovelluksiin ja internet markkinointiin keskittyvä startup-yritys, joka on päättänyt laajentaa toimintaansa mobiilipeleihin. Yritys tilasi opinnäytetyönä mobiilipelin määrittämis- ja suunnittelutyön, jonka perusteella lopullinen peli toteutetaan. Opinnäytetyöhön ei kuitenkaan kuulunut pelin toteutusosaa tai ohjelmistomäärittäystä. Projekti toteutettiin 2012 kevään ja kesän aikana.</p> <p>Tässä opinnäytetyössä lähdettiin selvittämään mobiilipelin toteuttamiseen vaadittavia tekijöitä. Työ aloitettiin tutkimalla mobiilimarkkinoita ja niillä menestyneitä mobiilialustoja yleisesti. Selvitystä laajennettiin sitten perehtymään erityisesti Googlen Android-alustaan sekä sille saatavilla olevaan AndEngine-pelimoottoriin ja Box2D-fysiikanmallinnukseen.</p> <p>Opinnäytetyön kirjallisina tuotoksina tuotettiin selvityksen lisäksi konsepti- ja määrittelydokumentit sekä tekninen dokumentti. Konseptidokumentti kuvaa pelin teeman ja tarkoituksen, määrittelydokumentti määrittää pelille toiminnallisuudet ja listaa pelin toteuttamiselle tarvittavat elementit kuten hahmot ja äänet. Tekninen dokumentti laajentaa määrittelydokumenttia siten, että siinä lisätään pelin elementeille tekniset määrittelyt kuten kuvien koot ja äänitehosteiden kestot. Työn aikana tuotettiin myös grafiikka- ja äänielementtejä, joita käytetään lopullisessa pelissä.</p> <p>Opinnäytetyön tavoitteena oli selvittää pelin toteutukselle oleelliset tekniikat sekä luoda pelin toiminnallisuuksien ja elementtien määrittelyt. Nämä tavoitteet saavutettiin joten toimeksiantaja voi käyttää opinnäytetyön tuotoksia hyväksi aloittaakseen peliprojektin toteutusvaiheen.</p>	
Asiasanat Mobiilimarkkinat, Mobiilialusta, Mobiilipeli, Android	

Degree Programme in Information Technology

<p>Authors Jouni Pärssinen</p>	<p>Group or year of entry TIKO06S</p>
<p>The title of thesis PRODUCING A MOBILE GAME ON ANDROID PLATFORM FOR COMPANY X</p>	<p>Number of pages and appendices 50+41</p>
<p>Supervisor(s) Juhani Välimäki</p>	
<p>The increasing popularity of mobile gaming introduces an interesting market space for companies. Company X is a few years old startup company that focuses in mobile applications and internet marketing, which has decided to expand to mobile games. The company has ordered a thesis of specification work for a mobile game, which will be used to produce the final game. The implementation of the game or programming specification is not a part of the thesis. The project is conducted during spring and summer of 2012.</p> <p>This thesis began with the goal to clarify what factors are required to realize a mobile game. The work started by studying the mobile markets and the mobile platforms that have prospected in them. The study was expanded specifically to Google's Android platform and the game engine AndEngine and physics modelling Box2D available to it.</p> <p>In addition to the conducted studies the literary results of the thesis included a concept, specification and technical document. The concept document describes the theme and purpose of the game whereas the specification document brings functionality to the game end lists the elements required for the game, like characters and sounds. The technical documents expands the specification document by adding technical specifications for the elements, like image sizes and lengths for audio effects. The project also included producing of graphics and audio elements that will be used in the final game.</p> <p>The goal of the thesis was to clarify the required technologies for the implementation of the game and to create a specification for the functionality and elements. The goals of the project were met so the client can use the results of the thesis to start the implementation phase of the game project.</p>	
<p>Key words Mobile markets, Mobile platform, Mobile game, Android</p>	

Sisällys

SANASTO	1
1 JOHDANTO	3
1.1 TARKOITUS JA RAJAUS	4
1.2 RAKENNE	5
2 MOBIILIALUSTAT	6
2.1 MOBIILIALUSTOJEN MARKKINAOSUUDET	6
2.1.1 MOBIILIALUSTOJEN EROT JA YHTÄLÄISYYDET	8
2.1.2 MOBIILIKEHITYKSEN MAHDOLLISUUDET	9
2.1.3 MOBIILIKEHITYKSEN HAASTEET	9
2.2 SOVELLUSKAUPAT	10
2.2.1 APPLE APP STORE	11
2.2.2 GOOGLE PLAY	11
2.2.3 WINDOWS PHONE MARKETPLACE	12
2.2.4 SOVELLUSKAUPPOJEN EROT	12
3 ANDROID OHJELMISTOALUSTANA	14
3.1 OMINAISUUDET	14
3.2 JÄRJESTELMÄARKKITEHTUURI	15
3.3 OHJELMISTOKEHITYS	16
4 KOLMANSIEN OSAPUOLIEN TEKNOLOGIAT	18
4.1 ANDENGINE	18
4.1.1 KESKEISET KÄSITTEET	19
4.1.2 OPENGL	23
4.1.3 OPENGL ES	23
4.1.4 GLES1 & GLES2	23
4.1.5 LAAJENNUKSET	24
4.2 Box2D	26
4.2.1 KESKEISET KÄSITTEET	27
4.2.2 ANDENGINE JA BOX2D	29
5 PELIPROJEKTI	30
5.1 IDEA	30

5.2	KONSEPTIDOKUMENTTI	31
5.3	MÄÄRITTELYDOKUMENTTI	32
5.4	TEKNINEN DOKUMENTTI	34
5.5	ÄÄNISISÄLTÖ	34
5.6	2D-SISÄLTÖ	35
5.7	3D-SISÄLTÖ	36
6	POHDINTA JA ARVIOINTI	37
6.1	SAAVUTETUT TULOKSET	37
6.2	KUSTANNUKSET	40
6.3	KEHITTÄMISEHDOTUKSET	41
6.4	OPPIMISEN ARVIOINTI	42
7	YHTEENVETO	44
LÄHTEET		45

Sanasto

Android

Googlen kehittämä avoimeen lähdekoodiin perustuva mobiilikäyttöjärjestelmä.

iOS

Applen iPhoneen, iPadin ja iPod Touchin käyttämä mobiilikäyttöjärjestelmä.

Symbian

Nokian käyttämä mobiilikäyttöjärjestelmä.

Windows Phone

Microsoftin kehittämä mobiilikäyttöjärjestelmä.

Blackberry

Kanadalaisen Research in Motionin (RIM) puhelinmallisto.

Sovelluskauppa

Mobiililaitteiden sovellusten kauppapaikka josta käyttäjät voivat ladata maksullista ja ilmaista lisäsisältöä laitteisiinsa.

SDK, Software Development, Kit

Kokoelma ohjelmaluokkia ja työkaluja joiden avulla ohjelmistokehitys tapahtuu kullekin alustalle.

Pelimoottori

Pelimoottori on kokoelma ohjelmistokirjastoja joiden varaan peli kehitetään. Se voi sisältää esimerkiksi kameraohjaukseen ja resurssien hallintaan liittyvät kirjastot jolloin pelikehittäjän ei tarvitse luoda näitä ohjelmia alusta asti itse.

AndEngine

Androidille kehitetty avoimen lähdekoodin 2D-pelimoottori. AndEngine on kokoelma Java-luokkia joiden ympärille pelin Java-projekti rakennetaan.

Box2D

Fysiikanmallinnusta ja törmäyksenhallintaa suorittava moottori.

Sprite

Tietokonegrafiikassa käytetty kaksiulotteinen pinta tai kuva.

Mobiilialusta

Mobiililaitteella käytettävä ohjelmisto tai käyttöjärjestelmä.

1 Johdanto

Vuonna 2007 Yhdysvaltalainen Apple julkaisi iPhoneen, joka aloitti älypuhelimien nousun kuluttajamarkkinoilla. Tätä ennen älypuhelimet olivat pitkälti yrityskäyttäjien suosimia laitteita, jotka yhdistivät sähköpostitoiminnallisuudet ja Internet-selaimen matkapuhelimeen. (Reed, 2010)

Maailman neljästä miljardista matkapuhelimesta neljäsosa on älypuhelimia. Internetin käyttö mobiililaitteilla tulee ohittamaan tietokoneet vuoteen 2014 mennessä. Vuonna 2011 älypuhelimilla oli Suomessa väkilukuun nähden 44 % penetraatio, joka tarkoittaa, lähes joka toisella suomalaisella oli älypuhelin. (Hepburn, 2011; Ahonen, 2011)

Älypuhelimien suosiota on nostanut erityisesti sovelluskaupat, jotka toimivat kauppapaikkana ja jakelukanavana älypuhelimien ja muiden mobiililaitteiden oheissovelluksille. Alun perin Applen iPhoneen avaama Appstore on poikunut vastaavat sovelluspaikat jokaiselle merkittävälle mobiilialustan valmistajalle. Sovelluskauppojen suurin merkitys on jatkuvan lisäarvon tuottaminen laitteen omistajalle erilaisten apu- ja hupisovellusten muodossa. (Rowinski, 2012)

Vuonna 2011 mobiilisovellusten tuottamat tulot sovelluskehittäjille ja muille sidosryhmille olivat arvoltaan 8,5 miljardia yhdysvaltain dollaria ja summan odotetaan olevan moninkertainen muutaman vuoden päästä. (Reisinger, 2012)

Älypuhelistista on tullut perustavanlaatuinen osa arkeamme ja niitä käytetään uutisten lukemiseen, musiikin kuunteluun, navigointiin ja erityisesti pelaamiseen. Yli 60 % älypuhelimien käyttäjistä käyttää laitettaan pelaamiseen, joka on 20 % enemmän kuin esimerkiksi musiikin kuunteluun. Älypuhelimista on täten tullut aina mukana kulkevia alustoja joko vakavampaan tai satunnaisempaan pelaamiseen. (Hepburn, 2011)

Mobiilipelaamisen jatkuva kasvu tarjoaa houkuttelevan uuden markkina-alueen yrityksille. Mobiilipelin tuottaminen on huomattavasti yksinkertaisempaa kuin perinteisen tietokone- tai konsolipelin tuottaminen. Laadukkaan mobiilipelin

tekemiseen saattaa riittää yhden henkilön yritys ja nollabudjetti. Parhaimmassa tapauksessa ainoa kiinteä kustannus saattaa olla pelin julkaisukanava, eli sovelluskauppa, joka on luotu jokaiselle laitealustalle. (Swallow, 2011)

Läsnäolo mobiilikanavissa¹ on muuttumassa vaatimukseksi yrityksen kilpailukyvyyn säilyttämiseksi. Yhä useampi yritys tarjoaa joko mobiilisovelluksia tai vähintään mobiilioptimoituja versioita verkkopalveluistaan. Lisäksi taulutietokoneiden räjähdysmäinen kasvu on asettanut uudet vaatimukset palveluiden ja sovellusten skaalautuvuudelle eri kokoisille mobiililaitteille. (Fishback, 2012)

Yritys X on muutaman vuoden ikäinen mobiilisovelluksiin ja Internet-markkinointiin keskittyvä startup-yritys. Mobiilipelien kehitykseen katseensa suuntaava yritys tarvitsee kokonaisvaltaista kehitystä ohjelmoinnin, grafiikan, animaation ja muun median tuottamiseen. Yrityksen tavoitteena on toteuttaa mobiilipeli Android-alustalle ja julkaista se sovelluskaupassa.

1.1 Tarkoitus ja rajaus

Tämä opinnäytetyö käsittelee mobiilipelin tuottamista Googlen Android-alustalla. Tämän projektin aikana toteutetaan puitteet, jonka pohjalta voidaan toteuttaa mobiilipeli yritykselle X käyttäen Androidia.

Opinnäytetyöhön sisältyy pelin suunnittelu- ja määrittelytyö, jonka perusteella yritys X voi toteuttaa mobiilipelin. Työhön kuuluu erillisten dokumenttien lisäksi peligrafiikan tuottamista, animaatiotyötä, median, kuten äänitehosteiden ja musiikin, tuottamista. Työssä myös tutustutaan ja arvioidaan Android-mobiilialustan soveltuvuutta aloittavan yrityksen järjestelmävalinnaksi. Tätä varten tutustutaan Androidiin kehitysalustana sekä tarkastellaan siihen tarvittavia ohjelmistoympäristöjä ja työkaluja. Tämän lisäksi selvitetään kolmansien osapuolien valmiiden avoimen lähdekoodin ohjelmakirjastojen, kuten fysiikkamallinnuksen, soveltuvuutta projektin pelissä käytettäväksi.

¹ Kirjoittajan oma käännös englanninkielisestä termistä *mobile presence*.

Tähän opinnäytetyöhön ei lukeudu ohjelmakoodin tuottamista tai ohjelmistorakenteiden määrittelyä. Tämän projektin aikana tuotetaan dokumentteja jotka toimivat lähtökohtana ohjelmistosuunnittelulle ja toteutukselle. Lisäksi tuotoksina syntyvää mediaa kuten kuvia ja ääniä tullaan käyttämään lopullisessa pelissä. Projektin määrittelyosuus ja muut projektin myötä tuotetut tiedostot määritellään salaisiksi, eivätkä ne sisälly tähän raporttiin.

1.2 Rakenne

Opinnäytetyö jakautuu rakenteellisesti teoriaosuuteen ja määrittelyosuuteen. Työn teoriaosuus kuvauksen eri mobiilialustoista sekä erityisesti Googlen Androidista kehitysympäristönä. Teoriaosuus kuvaa lisäksi projektin aikana käytettävät kolmansien osapuolien teknologiat. Opinnäytetyön määrittelyosa toteutetaan erillisillä salaisiksi määriteltävillä dokumenteilla joihin sisältyy pelin konsepti- ja määrittelydokumentti sekä tekninen dokumentti.

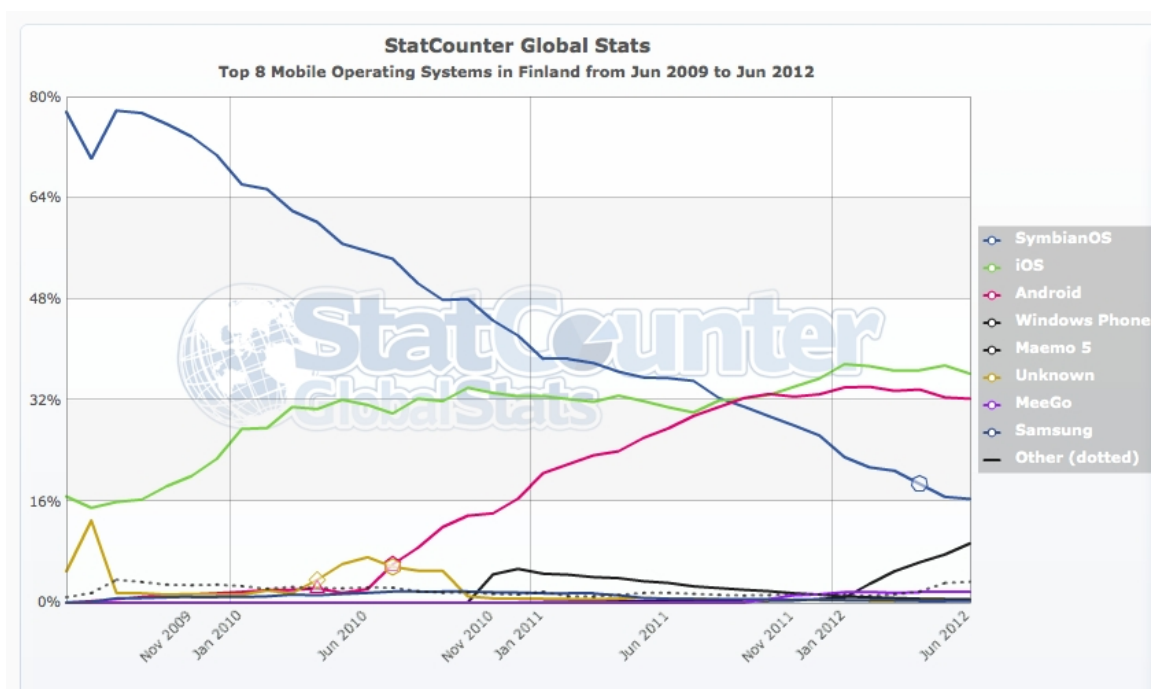
2 Mobiilialustat

Mobiilialusta on yleisnimitys mobiililaitteiden käyttöjärjestelmille. Yhdellä mobiilialustalla voi olla useiden eri valmistajien laitteita useissa eri muodoissa.

Esimerkkejä mobiilialustoista on Googlen Android, Applen iOS, Microsoftin Windows Phone ja Research in Motionin Blackberry.

2.1 Mobiilialustojen markkinaosuudet

Suomessa mobiilikäyttöjärjestelmiä dominoi Applen iOS, Googlen Android sekä Nokian Symbian -käyttöjärjestelmät. Näistä iOS ja Android kamppailevat kärkipaikasta vain muutaman prosenttiyksikön erolla markkinaosuuksista. Etenkin viimeisen kuuden kuukauden aikana molempien osuudet ovat noudattaneet hyvin samanlaisia muutosratoja. Kolmikon häntää pitävä Nokian käyttämä Symbian on menettänyt tasaisesti markkinaosuuttaan vuoden 2009 lähes 80 % osuudestaan nykyiseen 20 % osuuteen. (StatCounter, 2012)



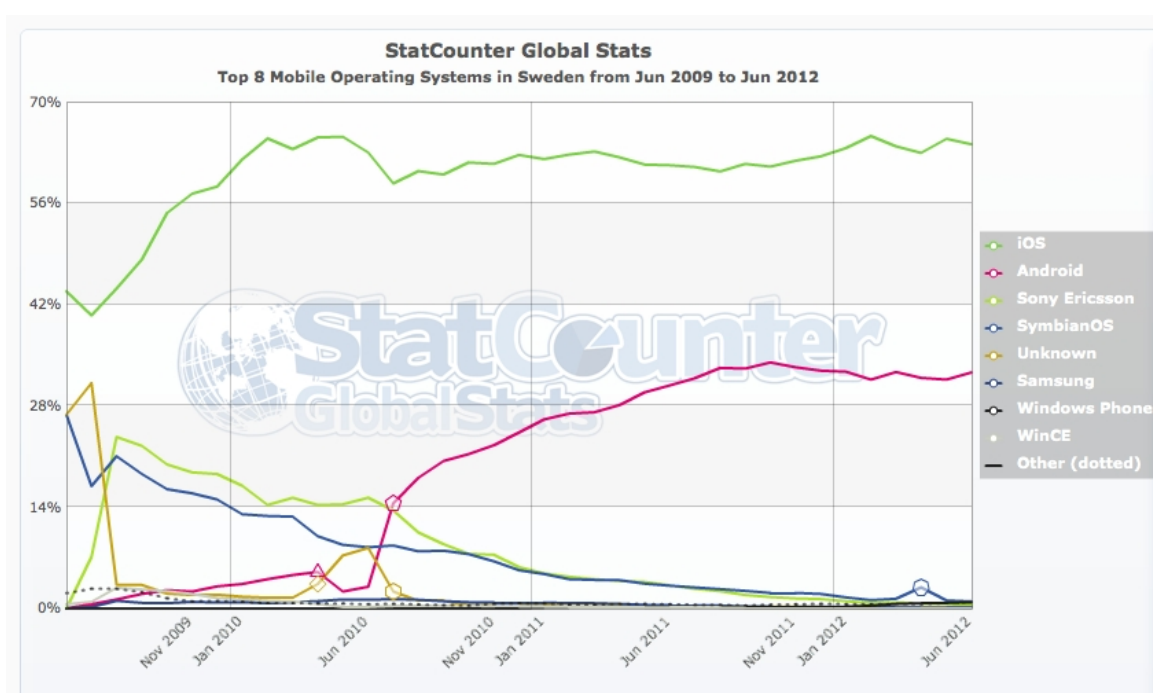
Kuvio 1. Mobiilialustojen markkinaosuudet Suomessa. (StatCounter, 2012)

Tasaisesti muutaman prosentin marginaaliosuuksista kamppailee Samsungin Bada-alusta ja Nokian Maemo ja MeeGo -alustaa käyttävät laitteet. Uutena pelurina

markkinoilla näkyy Microsoftin Windows Phone -alusta, jonka Nokia onkin valinnut sen ikääntyneen Symbianin korvaajaksi. Windows Phonen markkinaosuus Suomessa lähestyy kirjoitushetkellä kymmentä prosenttia jota selittää omalta osaltaan suomalaisten Nokia-uskollisuus (Niemeläinen, 2012)

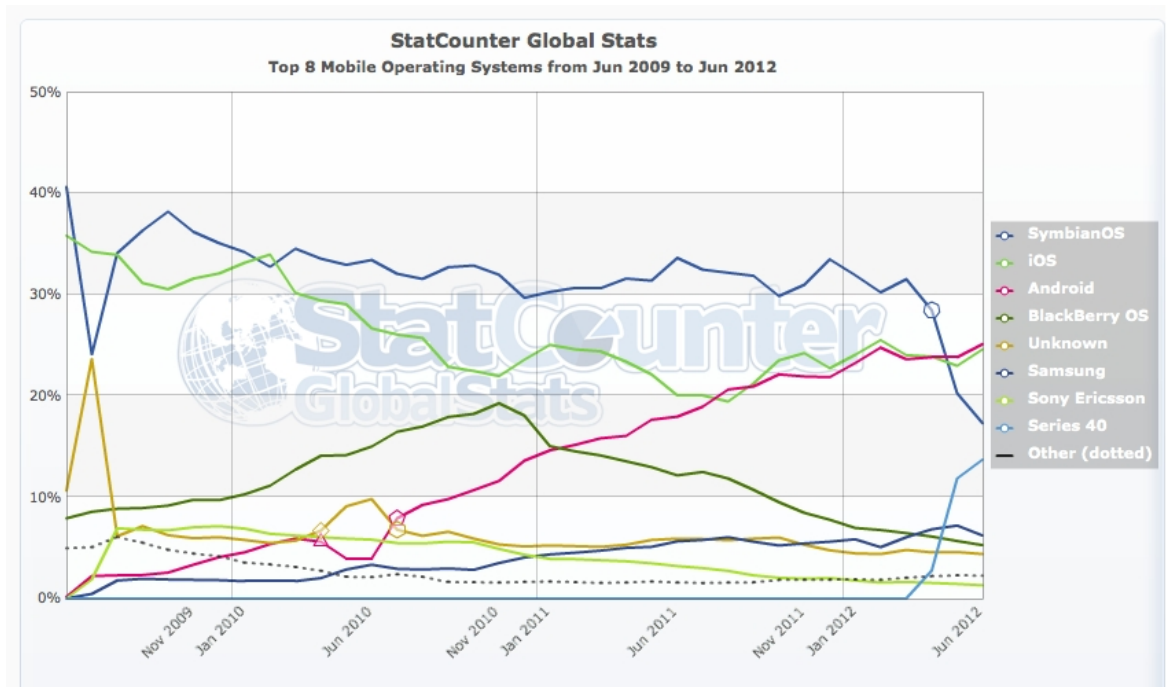
Naapurimaassa Ruotsissa osuudet ovat melko erilaiset kärjen suhteen.

Markkinaosuuksia johtaa Applen iOS yli 60 % osuudella, jota seuraa Googlen Android noin 30 % osuudella. Windows Phonen osuus Ruotsissa on alle prosentin luokkaa, johon myös aikaisemmin melko vahvoilla ollut Symbian on laskenut.



Kuvio 2. Mobiilialustojen markkinaosuudet Ruotsissa (StatCounter, 2012)

Kun tarkastellaan koko maailmanlaajuisia markkinaosuuksia voidaan todeta, että älypuhelinmarkkinoilla Applen iOS ja Googlen Android ovat johtajia. Yksikään muu noususuhdanteessa oleva älypuhelinlusta ei edes erottaudu StatCounterin listalle. Myös Windows Phone loistaa poissaolollaan maailmanlaajuisissa tilastoissa. Maininnan arvoista on, että Nokiolla on yhä suurin markkinaosuus, kun lasketaan yhteen kaikki sen mobiilialustat. (StatCounter, 2012)



Kuvio 3. Mobiilialustojen markkinaosuudet maailmanlaajuisesti. (StatCounter, 2012)

2.1.1 Mobiilialustojen erot ja yhtäläisyydet

Mobiilialustat eroavat toisistaan samassa määrin kuin töppöytäkäytössä oleva Windows-järjestelmä eroaa Linux-järjestelmästä. Niiden ohjelmat ei ole yhteensopivia toistensa kanssa ja ohjelmistokehitys on erilaista. Sovelluskehitys esimerkiksi iOS:n ja Androidin kesken eroaa niin kehitysympäristöltään kuin ohjelmointikieleltäänkin. Seuraavat kappaleet kuvaavat näistä Applen, Googlen ja Microsoftin mobiilialustat. Vertailu keskittyy näihin alustoihin sillä ne edustavat ainoita älypuhelinlustoja joiden markkinaosuudet ovat olleet tasaiset tai noususuhdanteessa viimeisen kuuden kuukauden aikana.

Applen iOS -käyttöjärjestelmä pohjautuu samaisen yhtiön tietokoneille tarkoitettuun OS X:ään. Vuonna 2007 julkistettu alun perin iPhone OS nimellä kulkenut käyttöjärjestelmä nimettiin uudestaan iOS:ksi vuonna 2010 iPadin julkistamisen jälkeen, sillä nämä kaksi laitetta käyttivät samaa versiota käyttöjärjestelmästä. iOS:n sovelluskehitys tehdään OS X:n tapaan objektiiv-C:llä Applen Xcode-kehitystyökalulla. (Apple, 2012)

Googlen Android-alustalle kehitys toteutetaan lähtökohtaisesti Javalla. Järjestelmä kääntää Java-ohjelman laitteen ymmärtämäksi konekielelle² sisäänrakennetun Dalvik-virtuaalikoneen avulla. Kehitykseen yleisimmin käytetty kehitystyökalu on Eclipse, johon ladataan lisäksi Androidin kehitystyökalut eli SDK-ohjelmistopaketti. (Google, 2012)

Windows Phonen osalta kehitys toteutetaan joko XNA- tai Silverlight-ohjelmistokehityksellä.³ Näistä XNA on tarkoitettu pelikehitykseen ja Windows Phone 7:lle mukautettu versio Silverlightista on tarkoitettu sovelluskehitykseen. Kumpaakin tekniikkaa on myös mahdollistaa yhdistää käytettäväksi yhdessä Windows Phone -sovelluksessa. (MSDN, 2012)

2.1.2 Mobiilikehityksen mahdollisuudet

Koska mobiililaitteille suunnatut ohjelmistot ovat toiminnoiltaan ja käyttöliittymältään yksinkertaisempia kuin työpöytäohjelmistot, on niiden toteuttaminen yksinkertaisempaa. Koska mobiiliohjelmistojen kehitys on alana vielä varsin tuore, siellä on lisäksi paljon tilaa innovointiin. Kehittäminen mobiilialustoille onkin ajamassa perinteiseen työpöytäkokoon kehittämisen ohi. (Etisbew, 2012)

2.1.3 Mobiilikehityksen haasteet

Mobiiliohjelmistokehitykseen ryhtyessä törmää väistämättä haasteisiin joita ei ole olemassa työpöytäkokoon ohjelmoitaessa. Ensimmäisenä haasteena on laitteiden koko. Suositujen älypuhelimien koot liikkuvat kolmen ja viiden tuuman alueella, joka johtaa siihen, että käyttöliittymälle on hyvin rajatusti tilaa. Uusien älypuhelimien näyttötarkkuudet riittäisi hyvinkin yksityiskohtaisiin käyttöliittymiin, mutta niiden käyttö ilman stylus-kynää⁴ olisi hankalaa ellei mahdotonta.

² Käännös englanninkielisestä termistä *machine language*.

³ Käännös englanninkielisestä termistä *framework*.

⁴ Mobiililaitteiden näytöille suunniteltu osoitinlaite jolla käyttäjä ohjaa laitetta kuin kynällä.

Muita haasteita on esimerkiksi fyysisen näppäimistön puuttuminen, rajallinen suoritinteho ja muistin määrä. Ohjelmistokehitykselle asettaa rajoja myös se, että sovelluskehitys tulee toteuttaa alustavalmistajan määrittelemällä ohjelmointikielellä.

Etenkin älypuhelimille ohjelmoitaessa haasteena on se, että alustana toimii puhelin. Käyttäjät sallivat huomattavasti vähemmän virheitä puhelimessa käytettäville sovelluksille kuin esimerkiksi tietokoneella. Esimerkiksi se, että viallinen mobiiliohjelma aiheuttaisi ongelman puheluun vastaamisessa tai soittamisessa ei ole läheskään niin anteeksiannettavaa kuin viallinen ohjelma tietokoneella. Asiaa voisi verrata mahdolliselle älyautolle tehtyä viihdeohjelmaa, joka päättyisi jotenkin haittaamaan ajoneuvon käyttöä. (Murphy, 2010)

2.2 Sovelluskaupat

Alun perin älypuhelimia käytettiin pääasiassa valmistajan esiasentamilla ohjelmilla. Nuo ohjelmat laajensivat puhelimen perustoiminnallisuuksia esimerkiksi sähköpostilla, verkkoselaimella ja multimediasoittimella. Osaan puhelimista oli mahdollista asentaa kolmansien osapuolien kehittämiä ohjelmia, kuten javaan perustuvia apuohjelmia, jotka jossain määrin laajensivat laitteen käytettävyyttä.

Merkittävä murros tapahtui vuonna 2008, kun Apple lanseerasi mobiiliohjelmien sovelluskaupan, App Storen, sen iPhone-älypuhelimelle. Applen rooli on sinänsä merkittävä, että sen kauppapaikka oli ensimmäinen joka sai kunnollisen jalansijan käyttäjien ja ohjelmistokehittäjien keskuudessa. (Scott, 2009)

Applen sovelluskaupan menestyksen myötä muut valmistajat seurasivat perässä ja julkaisivat omat versionsa. Nokia julkaisi Symbianille Ovi Storen, Google puolestaan Android Marketin omalle Android-alustalleen ja Microsoft taas Marketplacen Windows Mobile -laitteilleen. (Scott, 2009)

Seuraavissa kappalaisessa katsastellaan Applen, Googlen ja Microsoftin sovelluskauppojen historiaa sekä nykytilaa. Lisäksi luodaan katsaus kunkin kolmen alustan kehitystyökaluihin ja ohjelmistokieliin.

2.2.1 Apple App Store

Apple julkaisi App Store -sovelluskaupan heinäkuussa 2008 laajentaakseen suosittua iPhone -älypuhelimensa käyttötarkoituksia. Sovelluskauppa lanseerattiin yhdessä uudistetun iPhone 3G -mallin myötä ja se tarjosi 500 sovelluksen valikoiman sekä ilmaisia että maksullisia mobiilisovelluksia, joita käyttäjät pystyivät ladata älypuhelimensa suoraan puhelimesta käsin.

Applen sovelluskauppa tarjoaa nykyään arviolta 550 000 ohjelmaa iPhonelle, iPod Touchille sekä iPadille. Maaliskuussa 2012 sovelluskauppa saavutti 25 miljardin ladatun ohjelman rajapyykin. Apple on joulukuussa sijoittajilleen pitämän tulospuhelun⁵ mukaan maksanut tähän mennessä sovelluskehittäjille myyntituloista 4 miljardia yhdysvaltain dollaria. (Brian, 2012; Dediu, 2012)

Applen kehitystyökalu Xcode on ilmainen käyttää, mutta jotta sovellus voidaan julkaistaan sovelluskaupoissa kehittäjältä vaaditaan maksullinen kehittäjä tunnus. Kirjoitushetkellä kehittäjä tunnus maksaa 99 yhdysvaltain dollaria vuodessa. (Apple, 2012)

2.2.2 Google Play

Google Play on Googlen aikaisemmin Android Marketina tunnettu sovelluskauppa Android-käyttöjärjestelmää käyttäville mobiililaitteille. Sovelluskauppa avattiin lokakuussa 2008, vain kolme kuukautta Applen oman App Store -sovelluskaupan jälkeen. Alkuvuonna 2012 Google nimesi sovelluskaupan uudelleen Google Playksi, jotta se vastaisi uusien sisältölajien kuten musiikin, kirjojen ja elokuvien mukaantuloa alun perin vain sovelluksille tarkoitettuun kauppapaikkaan. (Liedtke, 2012)

⁵ Kirjoittajan oma käännös englanninkielisestä termistä *earnings call*

Sovelluskaupassa on ladattavissa yli 410 000 sovellusta Androidia käyttäville älypuhelimille, taulutietokoneille sekä Google TV⁶ -televisiojärjestelmälle. Joulukuussa 2011 se saavutti 10 miljardin ladatun sovelluksen rajan. (Isaac, 2011)

Sovellusten julkaisemista varten Google Play -sovelluskaupassa kehittäjä tarvitsee Google Play -kehittäjäprofiilin. Kirjoitushetkellä kehittäjäprofiilin luontia varten tulee maksaa kertaluontoinen 25 yhdysvaltain dollarin summa rekisteröinnin yhteydessä. (Google, 2012)

2.2.3 Windows Phone Marketplace

Vuonna 2010 Microsoft julkaisi Windows Mobile 6 -käyttöjärjestelmälleen Marketplace-sovelluskaupan. Sama termi on otettu käyttöön nykyisen Windows Phone 7:n sovelluskaupalle, josta löytyi maaliskuussa 2012 70 000 sovellusta. Microsoft ei ole vielä toistaiseksi paljastanut sovelluskaupan latausmääriä. (Blandford, 2012)

Sovellukset julkaistaan Microsoftin App Hubin kautta joka maksaa kirjoitushetkellä 99 yhdysvaltain dollaria vuodessa.

2.2.4 Sovelluskauppojen erot

Kaikki listatut sovelluskaupat hoitavat perusasiat samalla mallilla. Kaikista saa ladattua ilmaisia ja maksullisia sovelluksia mobiililaitteeseen. Eroa tulee esimerkiksi sovelluskaupan kautta saavutettavista sisältötyypeistä. Esimerkiksi Google on sisällyttänyt kauppapaikkaan myös sähköiset kirjat, musiikin ja elokuvat. Apple on ratkaissut nämä sisältötyypit ainakin toistaiseksi erillisen iTunes-sovelluksen avulla. (App Store, 2012; Google Play, 2012; Marketplace, 2012)

Kahden suurimman, Applen ja Googlen, välillä esiintyy merkittävämpi ero hinnoittelumallissa. Applen kauppapaikassa pääosa ohjelmista on maksullisia kun

⁶ Googlen Androidiin perustuva ratkaisu jolla voidaan käyttää esimerkiksi verkkopalveluita television kautta

Googlen vastaavassa tilanne on toisinpäin. Eroa voi osaltaan selittää kummankin laitteen käyttäjäkuntien erot siinä minkälaisista sisältöä laitteeseen ladataan. Xyologicin tekemän tutkimuksen mukaan iOS-käyttäjät lataavat laitteeseen enemmän pelejä kuin ohjelmia. Androidilla tilanne on päinvastainen, sillä lataustilastojen perusteella laitetta käytetään pääsääntöisesti erilaisiin hyöty- ja viihdeohjelmiin. (Xyologic, 2011)

3 Android ohjelmistoalustana

Android on Googlen ja sen yhteistyökumppanien kehittämä käyttöjärjestelmä mobiililaitteille. Sen ensimmäinen versio on julkaistu loppuvuonna 2008 tarjoamaan täysin avoin alusta laitevalmistajille ja ohjelmistokehittäjille. Androidia hallinnoi Open Handset Alliance, johon kuuluu muun muassa Google, T-Mobile, Motorola, Intel, sekä muita mobiilialan edustajia ja laitteistovalmistajia. (Rubio, 2011)

Android on ohjelmistopino⁷ johon kuuluu käyttöjärjestelmä, väliohjelmisto⁸ ja perusohjelmat. Ohjelmakehitykseen luotu kehitystyökalu Android SDK⁹ tarjoaa lähtökohdat ohjelmointiin Android-alustalla Java- ohjelmointikielellä. (Android developers, 2012a)

Kirjoitushetkellä Androidin viimeisin virallinen versio on Android 4.0.3, koodinimeltään Ice Cream Sandwich. Laajan yhteensopivuuden saavuttamiseksi tämän projektin aikana kuitenkin käytetään versiota 2.1, jolla saavutetaan yli 98 % yhteensopivuus käytössä oleviin Android-laitteisiin. (Android developers, 2012b)

3.1 Ominaisuudet

Alusta tarjoaa kaikki modernin mobiilikäyttöjärjestelmän ominaisuudet. Järjestelmän perusominaisuuksiin kuuluu seuraavat osat (Android developers, 2012a):

- Ohjelmistorajapinta komponenttien uudelleenkäyttöä varten.
- Dalvik-virtuaalikone ohjelmien ajamiseen.
- WebKit-moottoriin pohjautuva verkkoselain.
- Grafiikkajärjestelmä sekä kaksiulotteisen ja kolmiulotteisen kuvan piirtämiseen.
- SQLite strukturoituun tiedon varastamiseen.
- Tuki yleisille ääni-, video- ja kuvamuodoille.

⁷ Käännös englanninkielisestä termistä *software stack*.

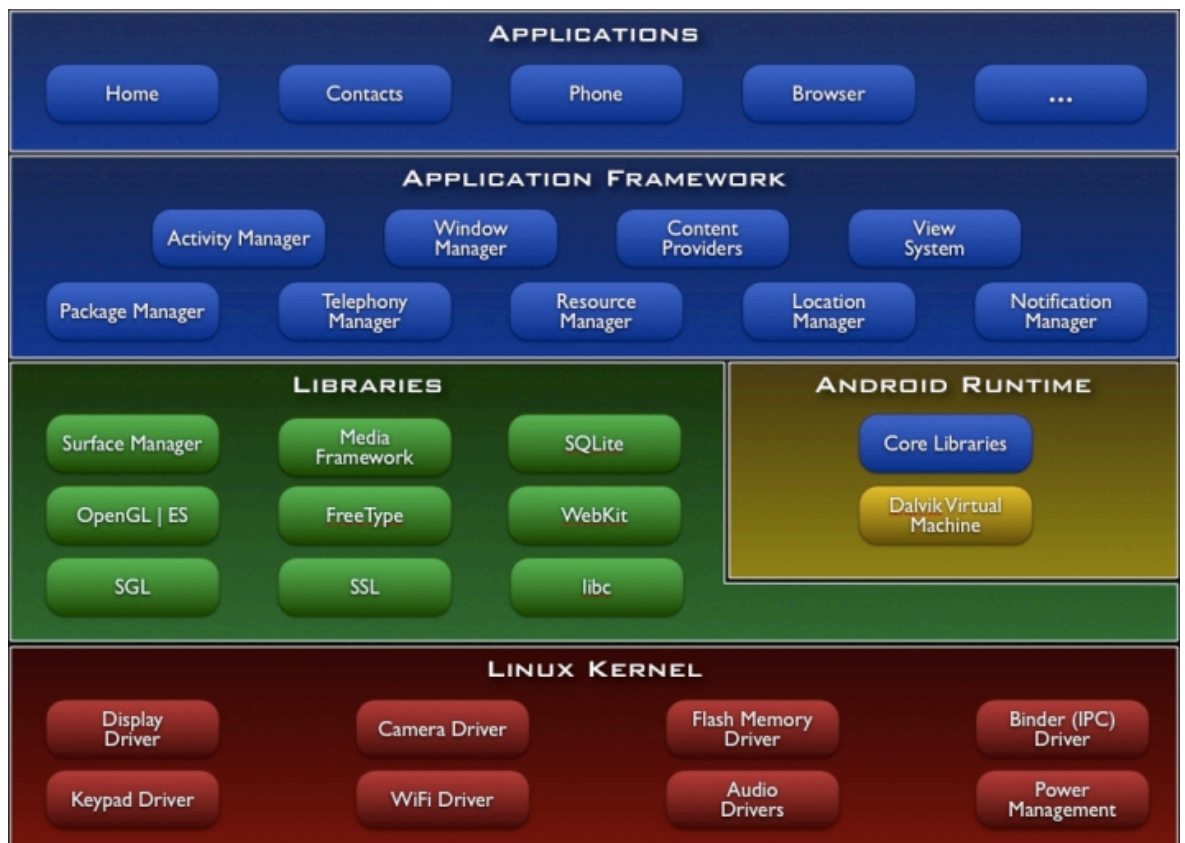
⁸ Kirjoittajan oma käännös englanninkielisestä termistä *middleware*, jolla tarkoitetaan ohjelmaa joka toimii liitoksena esimerkiksi ohjelman ja tietokannan välillä. (Ince, 2009)

⁹ Lyhenne sanoista *software development kit*.

- GSM radio.
- Bluetooth, EDGE, 3D ja WLAN -yhteydet
- Kamera, GPS, kompassi ja kiihtyvyyssanturi.
- Kehitystyökalut, joihin kuuluu muun muassa laite-emulaattori ja virheenetsintätyökalut.

3.2 Järjestelmäarkkitehtuuri

Järjestelmä on jaettu arkkitehtuurillisesti eri tasoihin (kuvio 1). Nämä tasot ovat Linux ydin (Linux Kernel), kirjastot (Libraries), Androidin ajonaikaiset komponentit (Android Runtime), ohjelmistorajapinta (Application Framework) sekä itse ohjelmat (Applications). Tässä opinnäytetyössä keskitytään pääsääntöisesti ohjelmistorajapintaan, jossa itse Java-ohjelmointi tapahtuu. (Android developers, 2012a)



Kuvio 4. Androidin arkkitehtuuri. (Android developers, 2012a)

Androidin avoin arkkitehtuuri tarjoaa ohjelmistokehittäjille mahdollisuuden kehittää ohjelmia, jotka voivat vapaasti käyttää hyväkseen järjestelmän eri osia, kuten

paikannustietoa, mediatoistoa sekä muita perusominaisuuksia. Ohjelmistorajapinnan tärkeimpiä osia ovat muun muassa

- näkymät joilla rakennetaan ohjelmiston kunkin osan käyttöliittymä
- Sisällöntuottajat joilla voidaan käyttää hyväksi laitteeseen tallennettuja tietoja kuten kontakteja
- Resurssihallinta jolla ohjelmistoon voidaan tuoda ulkopuolisia osia kuten kuvia ja ääniä

3.3 Ohjelmistokehitys

Android-ohjelmat kirjoitetaan pääsääntöisesti Java-ohjelmointikielellä.

Ohjelmaprojektin koodit ja resurssit paketoidaan yhdeksi apk-tiedostoksi¹⁰, joka asennetaan laitteeseen joko suoraan tiedostosta tai ohjelmakaupoista. Laitteeseen asennettuja ohjelmia ajetaan niin sanotussa hiekkalaatikossa, jossa ne voivat toimia vain ennalta määriteltujen rajojen sisällä.

Jokainen ohjelma saa käyttöjärjestelmältä yksilöllisen UNIX-käyttäjätunnuksen, joka hallitsee ohjelman käyttöoikeuksia laitteessa. Toimintavaltuudet pyydetään laitteen käyttäjältä ohjelmaa asentaessa, eikä ohjelma tämän jälkeen pysty pääsemään käsiksi määrittelemättömiin järjestelmäresursseihin ilman, että ohjelmasta julkaistaan uusi versio ja käyttäjä hyväksyy uudet käyttöoikeudet.

Tämä arkkitehtuuri toteuttaa niin kutsutun vähimmän etuoikeuden mallia¹¹, jolla sovelluksella on oikeus vain niihin järjestelmäkomponentteihin ja resursseihin joita se tarvitsee toteuttaakseen tarkoituksensa. Toisin sanoen sillä ei ole automaattista pääsyä

¹⁰ Androidin käyttämä päätte tiedostolle, johon ohjelma resurssineen pakataan. Voidaan verrata tietokoneissa käytettyihin ohjelmien asennusapuohjelmiin.

¹¹ Kirjoittajan käännös englanninkielisestä termistä *principle of least privilege*.

toimintoihin, joita ei ole erikseen määritelty sovelluksen niin kutsuttuun manifestitiedostoon¹².

Ohjelmaa asentaessa voidaan kuitenkin käyttäjän niin hyväksyessä sallia lupa esimerkiksi käyttää kameraa, sijaintitietoja tai yhteystietoja. Samalla ohjelmasertifikaatilla allekirjoitetut ohjelmat voivat myös toimia saman UNIX-käyttäjän sisällä, jolloin ne voivat vapaasti jakaa käyttöoikeudet toistensa resursseihin. Tällainen tilanne on esimerkiksi yhden ohjelmistokehittäjän kaksi eri ohjelmaa. (Android developers, 2012c)

¹² Manifestitiedosto on jokaiselle Android-sovellukselle vaadittu tiedosto, johon määritellään muun muassa vaaditut käyttöoikeudet, ohjelmistoversio ja projektin ulkopuoliset ohjelmistokirjastot, kuten Googlen karttasovellusrajapinta . (Android developers, 2012b)

4 Kolmansien osapuolien teknologiat

Tässä osiossa käydään läpi projektin aikana käytettävät kolmansien osapuolien tarjoamat teknologiat. Teknologioilla tarkoitetaan valmiita ohjelmistoja ja rajapintoja, joita voidaan käyttää hyväksi ohjelmistosuunnittelussa. Tämän projektin kannalta oleellimmat teknologiat on pelimoottori ja fysiikanmallinnus. Näistä pelimoottori hoitaa muun muassa pelin näkymät, sekä hahmojen asettelun pelinäkömään. Fysiikanmallinnus vastaa siitä mitä tapahtuu, kun pelielementit kuten hahmot käyttäytyvät, kun ne tulevat kosketukseen toistensa kanssa.

4.1 AndEngine

AndEngine on Nicolas Gramlichin kehittämä avoimeen lähdekoodiin pohjautuva 2D-pelimoottori Android-alustalle. AndEngine on käytännössä kokoelma Java-luokkia, joita pelikehittäjä voi käyttää omissa Android-projekteissaan. Pelimoottorin tarkoitus on helpottaa, nopeuttaa ja yksinkertaistaa pelikehitystä, jotta uuden kehittäjän ei tarvitse luoda yleisiä perustoiminnallisuuksia alusta asti itse.

AndEngine sisältää valmiita luokkia, jolla ohjataan esimerkiksi kameraohjausta, toimintojen suoritusjärjestystä ja pelielementtien käyttöä ohjelmassa. Pelimoottorilla määritellään mitä tapahtuu, kun käyttäjä koskee objektiin ja miltä objektin kuuluu näyttää. Sillä ohjastetaan myös pelin näkymiä jotka määrittelevät sen minkälaiselta pelimaailma näyttää pelaajalle. Tällä tavoin kehittäjä pystyy toteuttamaan suuren osan pelin perustoiminnoista vain muuntelemalla valmiita parametreja ja muuttujia, sekä lisäilemällä ohjelmakoodia tarpeen mukaan.

Helpoin tapa asentaa AndEngine on hakea sen lähdekoodi Eclipsen import-toiminnolla ja ladata uusin versio sen varastosta¹³. Huomioitavaa on, että useimmat AndEnginen ohjeet neuvovat lataamaan lähdekoodin Google Codesta, joka on kuitenkin vanhentunutta tietoa. AndEnginen kehittäjä Nicolas Gramlich on siirtänyt

¹³ Kirjoittajan käännös englanninkielisestä termistä *repository*.

2011 lähdekoodin varaston GitHubiin. Ajankohtaiset asennusohjeet löytää helpoiten AndEnginen omalta keskustelualueelta.

4.1.1 Keskeiset käsitteet

Seuraava osio sisältää Nicolas Gramlichin (Gramlich, 2010) ja Will Shelorin (Shelor, 2011) AndEnginelle listaamia keskeisimpiä käsitteitä ja termejä, jotka kehittäjän tulee sisäistää kyetäkseen käyttämään AndEngine-pelimoottoria tehokkaasti. Lista on koostettu yhdistelemällä tietoa Gramlichin ja Shelorin tekemistä kirjoituksista.

BaseGameActivity
Peliprojektin juuritaso. Kaikki pelin toiminnallisuudet luodaan tämän pääluokan alle. Pelimoottorin toiminnot määritellään tämän luokan alle.
OnLoadEngine
Pelimoottori ladataan tässä luokassa. Siinä määritellään muun muassa pelin orientaatio, kameran aloitussijainti sekä pelinäköymän näyttökoko eli resoluutio. Tässä luokassa luodaan tyhjä näkymä, jonka päälle pelielementit piirretään.
OnLoadResources
Tässä luokassa ladataan pelin käyttämät resurssit. Näihin lukeutuu esimerkiksi taustakuva, pelielementtien pinnat eli tekstuurit sekä äänielementit. Lataamalla resurssit laitteen muistiin pelin käynnistyessä mahdollistaa eri resurssien nopean uusiokäytön ilman, että niitä ladataan uudestaan muistiin tarvittaessa.
OnLoadScene
Tässä luokassa määritellään toiminnot ja käytetään aikaisemmin muistiin ladattuja resursseja. Tällä luokalla asetetaan esimerkiksi pelin taustakuva ja määritellään minkälaisia objekteja ja elementtejä näkymässä on ja miten ne käyttäytyvät.
OnLoadComplete

Tämä luokka käynnistyy kun kaikki on valmista. Tällä luokalla voidaan esimerkiksi käynnistää pelin taustamusiikki ja aloittaa peli.

Engine

Mahdollistaa ohjelman etenemisen asteittain. Tämä luokka käsittelee yleensä yhtä näkymää.

IResolutionPolicy

Määrittelee kuinka pelinäkökäyttö käyttäytyy eri näyttökokojen kanssa. Tällä voidaan määrittellä kuuluuko näkymän skaalautua koko ruudun kokoiseksi.

Camera

Kameralla voidaan määrittellä mikä osio pelialueesta piirretään ruudulle. Aliluokilla voidaan mahdollistaa zoomausta ja sulavia siirtymiä pelialueella.

Scene

Näkymään määritellään kaikki objektit ja elementit jota pelissä näytetään. Näkymää ohjataan tasoilla, joihin voidaan kuhunkin määrittellä kunkin tason elementit. Tällä mahdollistetaan esimerkiksi monitasoinen taustanäkymä ja päällimmäiseksi piirrettävä pelihahmo.

Entity

Olemus käsittää pelin objektit, kuvat ja tekstit. Kaikilla olemuksilla on sijaintiin, kierteeseen, kokoon ja muihin fyysisiin ominaisuuksiin liittyvät määrittelyt.

Texture

Tekstuuri, tai pinta on laitteen grafiikkapiirissä oleva kuva. Pinnoilla määritellään esimerkiksi taustakuva ja peliobjektien graafinen ulkoasu. AndEnginessä pintojen koko tulee olla aina kahden potenssissa, sillä järjestelmä varaa samalla kaavalla pinnoille määriteltävän muistin. Täten pintojen koot tulee olla esimerkiksi 64x64, 128x128, 32x256 ja niin edelleen. Käytettäessä esimerkiksi pintaa, jonka koko on

75x80 järjestelmä varaa aina muistia 128x128 mukaan.

Sprite

Sprite on tietokonegrafiikassa käytettävä kaksiulotteinen pinta, joka asetetaan peliobjektin päälle. AndEnginissä on käytössä kolmenlaisia spritejä:

1. Sprite on oletusarvoisesti muuttumaton pinta. Se voi olla esimerkiksi staattinen pelielementti kuten laatikko tai kivi.
2. TiledSprite eli laattasprite on oletusarvoisesti muuttumaton pinta, mutta sitä voidaan erikseen käskettäessä muuttaa ulkonäköään. Sen ilmeeksi määritellään tietty geometrinen sijainti pintalevyiltä¹⁴ jolloin pinnasta käytetään kerrallaan vain jokin tietty osa. Spriten ilme voidaan muuttaa siirtymällä eri kohtaan pintalevyä.
3. AnimatedSprite eli animoitu sprite on kokoelma pintoja jotka vaihtuvat määritellyllä aikavälillä. Animoitua spriteä voidaan silmukkaa toistamalla¹⁵ käyttää esimerkiksi taustalla näkyvän meren aaltojen animointiin tai puun lehtien heilumiseen tuulessa. Sillä voidaan myös saada esimerkiksi pelihahmosta eläväisemmän ja dynaamisemmän oloinen.

AndEnginen dynaamiset pelielementit koostuvat pitkälti spriteistä joita ladataan pelinäkömään. Spriten vastine kolmiulotteisessa pelimoottorissa on monikulmio eli polygoni.

ITextureSource

Tällä voidaan ladata kuva tietyksi kohdaksi pintaa, jolloin yksittäinen pelin pinta voi koostua useista kuvista.

TextureRegion

¹⁴ Kirjoittajan käännös englanninkielisestä termistä *tile texture*.

¹⁵ Kirjoittajan käännös englanninkielisestä termistä *loop*. Silmukan toistamisella tarkoitetaan ennalta määritetyn työvirran toistamista joko määritellyn iteraation verran tai ikuisesti.

Pintaan määriteltävä suorakulmio, jolla `sprite`¹⁶ määrittelee mitä osiota pinnasta näytetään. Osa niin sanottua pintalevyä, josta voidaan kerrallaan näyttää haluttua osaa.

Scene Touch Events

Scene Touch Eventit, eli näkymän kosketustapahtumat, tarkoittavat tilannetta, jossa käyttäjä käsittelee peliobjekteja laitteen kosketusnäyttöä käyttäen. Kosketusten tunnistaminen määritellään tapahtuman, eli eventin¹⁷ `onTouchEvent()` alle. Sillä määritellään esimerkiksi mitä tapahtuu kun käyttäjä vetää sormeja kosketusnäytöllä ylöspäin.

Item Touch Events

Item Touch Event, tai esineen kosketustapahtuma, on tilanne, jossa seurataan käyttäjän syöttämiä kosketusnäyttötapahtumia jonkun tietyn esineen tai objektin suhteen. Sen tyyppi on niin sanottu `TouchListener`, jolla voidaan seurata käyttäjän tekemiä tapahtumia.

Accelerometer

Accelerometer, eli kiihtyvyyssanturi, on laitteisiin kytketty sensori, joka tunnistaa laitteen kaltevuusasteen X, Y ja Z-akseleilla, eli sivuttais-, pysty- ja syvyys suunnissa. Kiihtyvyyssanturia käyttämällä peliä voi ohjata kääntelemällä laitetta, esimerkiksi sen käyttämiseen rattina rallipelissä.

PhysicsConnector

Mahdollistaa fysiikkamallinnuksen muuttaa `AndEnginen` objekteja. Merkittävää kun projektiin implementoidaan fysiikkamallinnus ja törmäyksen tunnistus.

¹⁶ Sprite on tietokonegraafikassa käytettävä 2D-bittikartta joka piirretään kohteen päälle muuttumattomana. (Microsoft, 2012)

¹⁷ Java-ohjelmoinnissa käytettävä englanninkielinen termi jolla tarkoitetaan tapahtumaa.

4.1.2 OpenGL

OpenGL on varjostinkieli¹⁸ jolla mahdollistetaan 2D- ja 3D-grafiikan esittäminen tietokoneilla. Se on erittäin laajasti käytetty ohjelmointirajapinta sen alustariippumattomuuden vuoksi, joten sen käyttökohteena voi olla normaali PC tai tieteelliseen laskentaan käytettävä supertietokone. (Khronos Group, 2012a)

4.1.3 OpenGL ES

OpenGL ES tai OpenGL for Embedded Systems on nimensä mukaisesti sulautetuille järjestelmille suunniteltu versio OpenGL-standardista. Se on rojaltivapaa, alustariippumaton ohjelmointirajapinta 2D- ja 3D-grafiikan tuottamiseen esimerkiksi pelikonsolleille, puhelimille, viihdelaitteille ja ajoneuvoille. Googlen Android -laitteet tukevat OpenGL ES:ää grafiikan piirtämiseen. (Android developers, 2012d)

OpenGL ES 1.1 on tarkoitettu määrätyn toiminnon laitteille¹⁹ ja vastaa OpenGL 1.5 -määritystä säilyttäen kuitenkin yhteensopivuuden OpenGL 1.0 -määrittelyn kanssa. Versio 1.1 tuo 1.0 versioon nähden laajemman tuen grafiikan laitteistokiihdytykselle.

OpenGL ES 2.0 on tarkoitettu ohjelmoitaville laitteille²⁰ ja vastaa OpenGL 2.0 -määritystä. Se mahdollistaa ohjelmoitavien grafiikkaputken²¹ käytön varjostimien ja ohjelmaobjektien luonnin, sekä vertexien ja varjostinpalasten käytön OpenGL ES varjostinkielellä. (Khronos Group, 2012b)

4.1.4 GLES1 & GLES2

GLES1 ja GLES2 kuvastavat AndEnginessä käytettävän OpenGL ES -rajapinnan versiota. Versio yksi toimii kaikilla laitteilla Android 1.0:sta lähtien. Versio kaksi vaatii vähintään Androidin version 2.2, koodinimeltä Froyo. Näistä kahdesta GLES2 on

¹⁸ Kirjoittajan käännös englanninkielisestä termistä *shading language*.

¹⁹ Kirjoittajan käännös englanninkielisestä termistä *fixed function hardware*.

²⁰ Kirjoittajan käännös englanninkielisestä termistä *programmable hardware*.

²¹ Kirjoittajan käännös englanninkielisestä termistä *graphics pipeline*.

aktiivisen kehityksen alainen ja on myös se versio, jota tämän opinnäytetyön peliprojektissa käytetään.

4.1.5 Laajennukset

AndEngineen on saatavilla useita laajennuksia jotka lisäävät ominaisuuksia pelimoottoriin. Laajennuksien käyttö on tarpeellista, jos haluaa käyttää hyväksi AndEnginen esimerkkiluokkia. AndEnginen laajennukset ovat Java-luokkia, jotka pitävät sisällään jonkun tietyn toimintokokonaisuuden. Esimerkki laajennuksesta on tuki monikosketukselle. Laajennukset asennetaan Java-projektiin samalla tavalla kuin itse AndEngine. (Angel, 2011)

Seuraavassa osio kuvaa AndEnginelle saatavilla olevat laajennukset. Lista on koottu Angelin (Angel, 2011) AndEngine-ohjeista ja Nicolas Gramlichin (Gramlich, 2011a-i) kuvauksista.

AndEngine - Live-Wallpaper Extension
Mahdollistaa Live-taustakuvien ²² luomisen AndEnginellä. (Gramlich, 2011a)
AndEngine - .MOD-Player Extension
Mahdollistaa .MOD-tiedostojen ²³ käytön AndEnginessä hyödyntäen XMP-soitinta ²⁴ . (Gramlich, 2011b)
AndEngine - Multiplayer Extension
Mahdollistaa kahden tai useamman pelaajan yhtäaikaisen pelisession asiakas-isäntä - yhteydellä. (Gramlich, 2011c)
AndEngine - MultiTouch Extension

²² Live-Wallpaper on ominaisuus Androidissa joka mahdollistaa animoitujen taustakuvien käytön laitteella.

²³ .MOD on moduuli-muotoinen alun perin Commodore Amigalla käytetty äänitiedostomuoto. (Chow, 2007)

²⁴ Extended Module Player on moduuli-muotoisten äänitiedostojen komentorivisoitinohjelma. (XMP, 2012)

Mahdollistaa monikosketuksen käytön sitä tukevilla laitteilla. Monikosketuksella mahdollistetaan pelimaailman manipuloinnin kahdella tai useammalla sormella yhtäaikaaisesti. Mahdollistaa esimerkiksi kuvan loitontamisen ja lähentämisen nipistys-
eleellä. (Gramlich, 2011d)

AndEngine - Physics Box2D Extension

Käyttää hyödykseen Erin Catton luomaa Box2D- 2D-fysiikkamallinnusmoottoria. Box2D mahdollistaa fysiikan sekä törmäyksen havainnan ja mallinnuksen AndEnginessä. (Gramlich, 2011e)

AndEngine - Augmented-Reality Extension

Lisätty todellisuus²⁵ -laajennuksella mahdollistetaan AndEnginen näkymän käyttäminen laitteen kameranäkymän päällä. (Gramlich, 2011f)

AndEngine - SVGTextureRegion Extension

Mahdollistaa SVG-tiedostojen²⁶, eli vektorigrafiikan lataamisen käyttöaikaaisesti pelin pinnoiksi tai bittikartoiksi. (Gramlich, 2011g)

AndEngine - Texture Packer Extension

Mahdollistaa sprite-arkkien²⁷ tuonnin TexturePacker²⁸ -työkalusta. (Gramlich, 2011h; Gramlich, 2011i)

²⁵ Käännös englanninkielisestä termistä Augmented Reality.

²⁶ SVG eli Scalable Vector Graphics on vektorigrafiikan tiedostomuoto. Vektorigrafiikka mahdollistaa kuvan tai objektin tarkkuuden säilyttämisen näyttökoosta tai tarkastelusyvyydestä huolimatta. (W3C, 2012)

²⁷ Kirjoittajan käännös englanninkielisestä termistä *spritesheet*.

²⁸ TexturePacker on Code'N'Webin luoma työkalu jolla voidaan muuttaa useita spritejä yheksi kootuksi arkiksi (Löw, 2010)

4.2 Box2D

Box2D on Erin Cattoon luoma kaksiulotteinen fysiikkamallinnusmoottori. Moottori on luotu C++ -ohjelmointikielellä ja on käytössä useissa suosituissa peleissä kuten Crayon Physics Deluxessa ja Angry Birdsissä. Box2D on pyritty tekemään mahdollisimman alustariippumattomaksi, joten se käyttää mahdollisimman vähän C++:n ominaisia toimintoja ja kirjastoja. Tämä mahdollistaa moottorin porttaamisen²⁹ muille alustoille mahdollisimman helposti. (Catto, 2011a; Humpries, 2011)

Peliprojektissa Box2D mahdollistaa törmäyksen ja fysiikan mallintamisen. Sen avulla peliohjelmoijat voivat luoda uskottavasti liikkuvia pelielementtejä ja saavat aikaan vuorovaikutteisemmän pelikokemuksen. Vuonna 2007 julkaistu Box2D on portattu muun muassa C#:lle, Flashille, Javalle, Pythonille, iOS:lle ja JavaScriptille. (Catto, 2011b; Box2D Forums, 2012)

Box2D käyttää liukulukuja, jotka on optimoitu toimimaan parhaiten metrijärjestelmässä, tässä tapauksessa metri-kilo-sekunti, tai MKS-järjestelmässä.³⁰ Ohjelmistokehittäjän tulee varoa käyttämästä Box2D:n yksikköinä peleissä muuten käytettäviä näyttöpisteitä eli pikseleitä. Box2D:n mallintamat yksiköt tulee muuttaa peliprojektiin sopivaksi sopivalla muunnoskaavalla. (Catto, 2011b)

Box2D muodostuu kolmesta eri moduulista, jotka käsittelevät eri toimintoja (Catto, 2011b):

1. Yleinen moduuli sisältää matemaattiset kaavat ja määritelmät.
2. Törmäys-moduuli määrittää muodot ja mahdollistaa törmäyslaskennan.
3. Dynamiikka-moduuli sisältää maailman, kappaleiden, kiinnittimien ja liitosten simuloinnin.

²⁹ Kirjoittajan käännös englanninkielisestä termistä *porting* jolla tarkoitetaan ohjelman muuttamista ohjelmointikieleltä toiselle.

³⁰ Alkuperäinen termi *meter-kilogram-second* eli MKS. (Catto, 2011b)

4.2.1 Keskeiset käsitteet

Box2D:tä käytettäessä ohjelmistosuunnittelijan tulee tuntua fysiikkamallinnusmoottorin keskeiset käsitteet ja terminologiat. Seuraava osio kuvaa niistä merkittävimmät. Listaus on koottu Erin Catton (Catto, 2011b) kirjoittamasta Box2D:n manuaalista, joka sisältää yksityiskohtaisen kuvauksen fysiikkamallinnusmoottorista. Manuaali toimii hyvänä apuna implementoitaessa Box2D:tä peliprojektiin.

Shape
Shape, eli muoto, on kaksiulotteinen objekti, kuten ympyrä tai monikulmio.
Rigid body
Rigid body, eli jäykkä kappale, toimii synonyyminä kappaleelle.
Fixture
Fixture, tai kiinnitin, sitoo muodon ja kappaleen yhteen. Sillä määritellään objekteille materiaaliominaisuus, kuten tiheys, kitka ja palautuvuus ³¹ .
Constraint
Constraint, eli rajoite, mahdollistaa rajoitteiden asettamisen objektien liikkuvuusakseleille. Rajoitteella voidaan esimerkiksi estää objektin liikkuminen sivusuunnassa.
Contact constraint
Contact constraint, eli kontaktirajoite, estää kappaleiden välisen läpäisyn. Tuolloin esimerkiksi jokin objekti ei pysty läpäisemään kappaletta.
Joint
Joint, eli liitos, mahdollistaa kahden tai useamman kappaleen kytkemisen yhteen.

³¹ Materiaalin ominaisuus joka pyrkii palauttamaan sen alkuperäiseen muotoon kun sitä muutetaan.

Joint limit
Joint limit, eli liitosrajoite, asettaa rajoitteita liitoksien liikkuvuudelle. Sillä voidaan estää objektien liikkuvuutta liitoskohdan suhteen. Tämä on verrattavissa esimerkiksi sormen kääntyvyyteen käden suhteen.
Joint motor
Joint motor, eli liitosmoottori, mahdollistaa ennalta määritellyn liikkuvuuden objekteille liitoskohdan suhteen.
World
World, eli maailma, sisältää kaikki objektit, rajoitteet ja materiaalit jotka ovat vuorovaikutuksessa toistensa kanssa.
Solver
Solver, eli ratkaisin, pystyy ratkaisemaan käytettävän maailman kontakti- ja liitosrajoitteet ja mahdollistaa fysiikkamallinnuksen etenemisen.
Tunneling
Tunneling, eli tunnelointi, on tilanne jossa kappale menevät sisäkkäin tai siirtyvät pelialueen ulkopuolelle. Tämä voi tapahtua jos fysiikkamoottori ei pysty laskemaan törmäyksiä tarpeeksi nopeasti. Box2D kiertää tätä ongelmaa laskemalla yhdyntäkohdan ennalta, jonka jälkeen niin kutsuttu aliselvitin laskee reaktion yhdyntäkohdalle.

4.2.2 AndEngine ja Box2D

C++:lla luotu Box2D on portattu Javalle JBox2D -nimisellä projektilla. Sen tarkoitus on kääntää kaikki Box2D:n toiminnot Java-kielelle. AndEnginen Box2D-laajennus käyttää kuitenkin suoraan natiivia C++ Box2D:tä Javan JNI-rajapinnan³² kautta. JNI-rajapinnalla mahdollistetaan C:llä ja C++:lla kirjoitetun ohjelmakoodin käytön Java-projekteissa. Sen etuna on ajantasaisuus viimeisimmän Box2D-version kanssa sekä huomattava nopeusetu Java-koodiin verrattuna. (AndEngine Wiki, 2012; Android developers, 2012e; Liang, 2002, Turner, 2012)

Itsessään AndEngine sisältää fyysisten ominaisuuksien määrittämisen pelielementeille, mutta sillä ei ole mahdollista mallintaa eri elementtien vuorovaikutusta. Box2D:n AndEngine-laajennus on mahdollistettu käyttämällä Android kehitystyökalun, SDK:n, NDK-lisätyökalua³³ ja tuomalla sillä Box2D:n LibGDX-porttauksen³⁴ laajennuksen Java-projektiin. (AndEngine Wiki, 2012)

³² Alkuperäinen termi *Java Native Interface*.

³³ Alkuperäinen termi *Native Development Kit*.

³⁴ LibGDX on Javalla kirjoitettu alustariippumaton kirjasto pelikehityksen avuksi (LibGDX, 2012)

5 Peliprojekti

Opinnäytetyön tavoitteena on toteuttaa projekti, jossa suunnitellaan ja määritellään peli Android-alustalle. Projektin aikana tuotetaan kuvauksien lisäksi grafiikkaa ja ääntä, jota käytetään lopullisessa pelissä. Tuloksena syntyy täten konsepti-, määrittely- ja tekninen dokumentti, kuvia, äänitehosteita, musiikkia sekä 3D-animaatioita. Projektin tuotokset määritellään luottamukselliseksi tätä raporttia lukuun ottamatta.

Tämä opinnäytetyöraportti ei pyri kuvaamaan pelin ideaa yksityiskohtaisesti, vaan nämä kuvaukset toteutetaan erillisillä konsepti-, määrittely- ja teknisellä dokumentilla. Nämä dokumentit toimivat perustana peliprojektille. Dokumentaatiomallina on käytetty Micah Hrehovcsikin luomaa pelisuunnitteluun soveltuvaa mallia, jota on mukautettu soveltumaan tässä peliprojektissa käytettäväksi.

Työn tilaaja yritys X pyrkii julkaisemaan markkinoille Android-pelin kuluvan vuoden aikana ja työn alustus on täten jo aloitettu vuoden vaihteessa. Opinnäytetyön mukana ensimmäisenä tuotettava konseptidokumentti pyrkii jalostamaan aivoriihien myötä syntyneet ideat strukturoituun muotoon.

Projektin tuotokset toimivat tukena ohjelmistosuunnittelussa ja tuotettu audiovisuaalinen sisältö tulee käytettäväksi peliin. Ohjelmistosuunnittelussa tarvitaan erityisesti tietämystä eri teknologioiden käytöstä, kuten pelimoottorin ja fysiikkamallinnuksen osalta.

5.1 Idea

Idea peliin on syntynyt tutkimalla Android sovelluskaupan latauslistoja ja selvittämällä siitä mitkä ovat ne peli-elementit, jotka ovat ominaisia suosituille peleille.

Sovelluskaupan tarjontaan tutustumalla selvisi, että sijoitus latauslistoilla ei välttämättä ole tae laadukkaasta ja pitkäkantoisesta mobiilipelistä. Monet erityisesti ilmaiset suuren suosion saavuttavista peleistä tippuvat alku-suosion jälkeen kärkilistoilta. (Google Play, 2012)

Peli-ideaksi on kiteytynyt kaksiulotteinen tarkkuutta ja ongelmanratkaisua vaativa peli, jossa pelihahmoilla pyritään selvittämään pelikentässä esiintyvät esteet ja etenemään etapeissa kentän loppuun. Pelissä käytetään hyväksi fysiikkamallinnusta, joka mahdollistaa reagoinnin törmäystilanteissa ja massan sekä lentoradan peliobjekteille.

5.2 Konseptidokumentti

Ensimmäinen peliä varten kehitettävä kirjallinen tuotos on konseptidokumentti (Liite 1: Konseptidokumentti_FB.docx). Konseptidokumentti kuvaa korkealla tasolla pelin perusidean, pelin tunnelman ja ilmeen, sekä arvioi mahdollisia käyttäjäryhmiä ja kaupallistettavuutta.

Pelianalyysissa pyritään kategorisoimaan pelin eri ominaisuudet yleisesti käytetyille määreille. Pelianalyysissa listataan ominaisuudet eri kokonaisuuksien alle. Pelin kuvauksessa luetellaan pelin teema, aihepiiri ja tyyli. Pelin tekniikka listaa pelin mobiilialustan, ja kohdelaitteet. Pelin myynti -osio listaa kohdeyleisön ja julkaisukanavat. Pelianalyysista saa nopean yleiskatsauksen siitä mihin olemassa oleviin kategorioihin peli lukeutuu ja mille laitealustalle se on tarkoitettu.

Pelin tunnelma kuvaa minkäläinen asetelma pelimaailmalla on ja minkäläistä hahmoja pelissä esiintyy. Kullakin pelihahmolla on tietynlainen persoona ja graafinen ilme jonka on tarkoitus helpottaa pelaajan samaistumista hahmoihin. Pelin viholliset jotka uhkaavat pelimaailmaa ja hahmoja kuvataan siten, että niiden läsnäolo pelimaailmassa on välitön ristiriita ja pelaajaa ei tarvitse erikseen opastaa ymmärtämään ketkä ovat vihollisia. Pelin ääniraita on suunniteltu siten, että se kuvastaa pelimaailman teemaa ja luo lisätunnelmaa ja immersiota pelaajalle.

Pelin mekaniikka käyttää hyväkseen fysiikkamallinnusta ja sen mahdollistamia ballistisia lentoratoja. Pelissä ammutaan pelihahmoa tasolta toiselle tuhoten samalla kentässä esiintyviä vihollisia. Laukaisumekanismissa otetaan oppia hyväksi todetuista elementeistä.

Pelissä pelataan kerralla yhtä kolmesta päähahmosta, joista kullakin on omat ominaisuutensa. Kenttien läpäisy parhaalla mahdollisella tavalla vaatii pelihahmojen käyttöä oikeassa järjestyksessä. Pelihahmot pystyvät sortamaan pelimaailman rakennelmia ja kimpoamaan isommista esteistä.

Pelin tarina on suunniteltu jaettavaksi erillisiksi luvuiksi, joka mahdollistaa pelin julkaisuversion pienemmällä määrällä kenttiä ja erillisiä elementtejä. Lukujen lopussa pelaajan on voitettava loppuvastustaja, jonka jälkeen luku merkataan läpäistyksi. Peliin julkaistaan lisää lukuja ohjelmapäivitysten muodossa. Pelin ensimmäiseen lukuun on suunnitteilla 15 kenttää, jotka nekin jakautuvat kenttäkokoelmiksi jotka sijoittuvat johonkin tiettyyn paikkaan pelimaailmasta.

5.3 Määrittelydokumentti

Määrittelydokumentti on tuotettavasti dokumenteista tärkein (Liite 2: Maarittelydokumentti_FB.docx). Se sisältää kaikki pelin toiminnallisuuksiin, hahmoihin, kenttiin, ääniin, grafiikkaan ja juoneen liittyvät määrittelyt. Määrittelydokumenttiin on koottu ja jäsennelty kaikki pelin ideoinnin jäljiltä hyväksytyt sisältö, jonka perusteella varsinaista peliä voidaan lähteä rakentamaan. Määrittelyssä otetaan valmiiksi huomioon esimerkiksi kaikki pelissä tarvittavat kenttäelementit joista kentät koostuvat ja äänet mitä pelissä kuullaan.

Pelin määrittelyssä asetetaan tietyt suunnittelurajoitteet, joiden mukaan pelin graafinen ulkoasu ja äänet tulee suunnitella. Rajauksella määritellään, että pelielementeissä ei tule olla liikaa yksityiskohtia, jotta ne mukailevat paremmin pelin temaa ja muuta visuaalista ilmettä.

Pelin eri elementit on jaettu kolmeen eri ryhmään, pelaajaelementteihin, viholliselementteihin sekä globaaleihin elementteihin. Näistä ensimmäisiin lukeutuu kaikki pelihahmoihin liittyvät määrittelyt. Pelaajaelementeille määritellään tässä hahmojen kuvaus, ominaisuudet sekä graafinen ulkoasu. Samalla kuvataan mitä toimintamahdollisuuksia pelaajalla on suhteessa pelihahmoihin, eli miten pelaaja aloittaa pelin, lataa kentän ja pelaa peliä.

Viholliselementeille kuvataan pelaajaelementtien tapaan hahmot ja niiden ominaisuudet. Tällä osiolla myös muodostetaan lista kaikista pelin aikana esiintyvistä vihollisista. Ominaisuuksien lisäksi viholliselementeille määritellään keinoäly ja sen toiminnallisuuden tilat. Tällä kuvataan miten viholliset käyttäytyvät suhteessa pelaajaan.

Globaalit pelielementit sisältävät kaikki neutraalit elementit, jotka vaikuttavat pelaajaan ja viholliseen vain välillisesti tai ei ollenkaan. Tällaisia elementtejä ovat esimerkiksi pelikentässä olevat tasot, joista pelaaja voi kimmottaa itsensä tai jonka päällä vihollinen voi olla. Välillisesti vaikuttavia elementtejä voi olla edellä mainitut tasot tai muut kentän objektit kuten puurakennelmat kun taas täysin neutraaleja elementtejä on esimerkiksi pelin taustagrafiikka.

Pelin käyttöliittymä näyttää kerrallaan pelaajalle vain osan pelikentästä loppuosan jatkuen näkyvän pelialueen ulkopuolelle. Pelin kamera liikkuu suhteessa pelaajaan ja paljastaa uutta kenttänäkymää sitä mukaa kun pelaaja siirtyy kentässä eteenpäin. Kenttäsuunnittelu toteutetaan siten, että peli aloitetaan aina pelin alareunasta.

Pelin tarina kuvaa mikä pelimaailma on, mikä hahmojen tausta on, miten ne liittyvät juoneen, sekä mitä viholliset ovat ja miksi ne ovat saapuneet pelimaailmaan. Tarina kuvaa miten pelihahmot tapaavat toisensa ja miten heiden seikkailunsa vihollisia vastaan alkaa ja etenee useiden kenttien läpi kohti lopullista vastustajaa.

Pelin aikana pelaajaa ohjeistetaan aputeksteillä jotka kouluttavat pelaajan pelimekaniikkaan, hahmojen valintaan, sekä niiden käyttöön eri tilanteissa. Aputekstit toteutetaan ponnahdusilmoituksilla jotka samalla tauottavat pelin.

Peliä varten tuotetaan lukuisia grafiikka- ja äänielementtejä jotka on myös määritelty dokumenttiin. Äänielementteihin lukeutuu pelin alussa ja päävalikossa soivat taustamusiiikit, sekä pelin aikana kuultavat äänitehosteet. Kullakin pelihahmolla ja vihollisella on omat äänensä, kuten myös kullakin pelimaailman neutraalilla objektilla.

5.4 Tekninen dokumentti

Tekninen dokumentti (Liite 3: Tekninen_dokumentti_FB.docx) sisältää pelin teknisemmän määrittelyt. Dokumentissa listataan ja kuvataan pelin elementit teknisellä tasolla sekä otetaan kantaa pelin toteutuksen muihin teknisiin puoliin ja vaatimuksiin.

Dokumentti kuvaa pelin järjestelmävaatimukset sekä toteutuksen vaatimukset käyttöoikeuksien suhteen. Vaatimuksena mainitaan, että pelin tulee vaatia vain niitä käyttöjärjestelmän oikeuksia jotka on sen toiminnalle välttämättömiä ja että vaaditut erikoisoikeudet ilmoitetaan mahdollisimman läpinäkyvällä tavalla käyttäjälle.

Pelin grafiikka- ja äänielementit listataan tässä dokumentissa siten, että grafiikalle määritellään formaatti, mittasuhteet, dimensiot sekä vaadittujen kuvien määrä per elementti. Näillä saadaan määriteltyä valmiiksi minkä kokoisia kuvia peliprojektia varten tulee luoda ja missä tiedostomuodossa ne tulee tuoda projektiin.

Koska osa elementeistä on animoituja, tulee niille myös määritellä animaation toteutukseen vaadittujen kuvien määrä. Pelin kaksiulotteiset elementit animoidaan siten, että niissä ajetaan toistolla tiettyä kuvasekvenssiä. Käytännössä elementin animaatio siis koostuu useasta erillisestä kuvasta. Esimerkiksi puun animaatio voi koostua kahdesta kuvasta, joista toisessa lehtien asento on hieman erilainen kuin ensimmäisessä. Näitä kahta kuvaa kierrättämällä saadaan animaatio jossa lehdet heiluvat tuulessa.

Pelin äänielementit kuvataan määrittelemällä niille pelin toiminnan kannalta paras tiedostomuoto. Lisäksi kaikki tarvittavat äänet listataan ja niille määritellään kesto. Tekninen dokumentti kuvaa myös pelin tekniset haasteet, sekä ehdottaa niille mahdollisia vaihtoehtoja, joita voidaan ottaa huomioon projektin edetessä.

5.5 Äänisisältö

Pelin äänet on lähtökohtaisesti toteutettu Applen Logic Pro -ohjelmistolla käyttäen hyväksi Native Instrumentsin Komplete 7 -ohjelmistopakettia. Musiikit koostuvat

useasta raidasta joista osa on nauhoitettu live-instrumenteilla käyttäen Logicin ja Komplete 7:n reaaliaikaisia efektejä värittämään raitoja. Muut raidat on nauhoitettu Logicin ja Komplete 7:n ääninäytteisiin perustuvilla virtuaali-instrumenteilla joita ohjailtaan sekä nuotittamalla suoraan Logicin sekvensserillä sekä MIDI-soittimilla.

Esimerkiksi pelin tunnusmusiikki on toteutettu akustisella kitaralla jonka äänet on taltioitu ulkoisella mikrofonilla Logicin raidaksi. Tämän jälkeen raitaan on lisätty erilaisia äänimaailmaa muuntavia efektejä ja suodattimia joilla on saatu aikaan pelin teemaan sopiva ääniteema. Toisena raitana käytetään melodista instrumenttia joka on nauhoitettu MIDI-koskettimistolla joka ohjailee yhtä Komplete 7:n ohjelman virtuaali-instrumenttia. Taustalle on myös nauhoitettu erilaisia teemaan sopivia taustaääniä.

Pelin äänitehosteet koostuvat pitkälti Komplete 7 -paketin äänigeneraattoreilla tuotetuista äänistä ja erilaisista ääninäytteistä koostetuista äänistä. Jotkut äänet jotka vaativat enemmän elollista tuntua on nauhoitettu mikrofonilla ja miksattu jälkikäteen ohjelmistojen työkaluilla.

5.6 2D-sisältö

Pelissä esiintyvät kaksiulotteiset elementit on kaikki pääsääntöisesti tuotettu tai muokattu OS X:lle saatavalle Pixelmator-kuvankäsittelyohjelmalle. Pixelmator soveltuu hyvin kevyeksi korvikkeeksi Adoben Photoshopille ja soveltuu kaikkeen kevyeen kuvanmuokkaukseen.

Graafiset elementit on viety projektiin joko yksittäisinä kuvina tai useista elementeistä koostuvina kuvina, jolloin pelimoottorissa voidaan määritellä mitä kohtaa kuvasta näytetään. Animoituiden elementtien on toteutettu pelimoottorilla näyttämällä tiettyjä kuvia sarjassa.

Esimerkkinä yksi pelin päähahmoista animoidaan siten, että siitä on tuotettu useita versioita joita sitten toistetaan silmukassa tietyssä järjestyksessä. Samalla tavalla toteutetaan esimerkiksi jonkun tietyn pelielementin hajoamisen animaatio, jossa

ensimmäisessä kuvassa se on ehjä, seuraavassa se alkaa hajoamaan osiin ja kolmannessa se on hajonnut osiin.

5.7 3D-sisältö

Pelissä käytettävät erilliset kolmiulotteinen grafiikka ja animaatiot on tuotettu NewTekin LightWave mallinnus- ja animaatio-ohjelmistolla. LightWaven Modeler-ohjelmalla on ensin tehty kolmiulotteisia malleja esimerkiksi hahmoista ja pelimaailmasta erilaisten kaksiulotteisten hahmotelmakuvien pohjalta.

3D-mallit on sitten tuotu LightWaven Layout-ohjelmaan jossa ne asetetaan kolmiulotteiseen maailmaan. Mallit valaistetaan Layoutissa erilaisilla valonlähteillä ja niiden pintoja muokataan sopiviksi Layoutin pintatyökalulla. Malleja kuvataan virtuaalisella kameralla jonka näkymän kautta lopullinen kuva muodostetaan.

Kolmiulotteinen kuva muodostetaan siten, että ohjelma laskee näkymän valaistusasetusten perusteella valosäteiden osumat kolmiulotteisille malleille ja piirtää ne sen ja mallien pinta-asetusten perusteella omaksi kuvaksi, joka voidaan tallentaa esimerkiksi pakatuksi kuvatiedostoksi.

Animaatioiden tapauksessa on kaiken muun lisäksi myös määritelty kolmiulotteisten objektien, valaistuksen ja kameran sijainnin ja arvojen muuttuminen aikajanalle. Tällöin voidaan määritellä esimerkiksi, että kamera liikkuu animaation aikana lineaarisesti pisteestä A pisteeseen B. Tällöin määritellään kameralle tietty sijainti aikajanana ensimmäiseen kuvaan³⁵ ja joku toinen sijainti animaation aikajanana viimeiseen kuvaan. Kun animaatio etenee ensimmäisestä kuvasta viimeiseen aikajanalla, ohjelma määrittelee kameralle vastaavan sijainnin kullekin aikajanana kuvalle. Kun jokainen aikajanana kuva muodostetaan ohjelmasta ja kuvat liitetään yhteen animaatioksi, saadaan käsitys, että kamera liikkuu sulavasti kolmiulotteisessa maailmassa.

³⁵ Käännös englanninkielisestä termistä *frame*.

6 Pohdinta ja arviointi

Opinnäytetyön lähtökohtana oli mobiilipelin toteutus Android-alustalla. Tilaus työlle syntyi Yrityksen X tahtotilasta siirtyä internet-markkinoinnin lisäksi keskittymään vahvemmin mobiilisovelluskehitykseen ja erityisesti mobiilipelikehitykseen. Projektia on alustettu alkaen loppukesästä 2011, jolloin ensimmäiset aivoriihet on järjestetty ja pelin raamit on määritelty. Päätös tehdä opinnäytetyö aiheesta syntyi syksyn 2011 aikana ja aihe-ehdotus työstä jätettiin alkuvuonna 2012.

Projektin motiivina oli perehtyä erityisesti Googlen Androidiin mobiilikehitysalustana nimenomaan pelikehityksessä ja selvittää mitä tekijöitä tulee ottaa huomioon onnistuneen toteuttamisen kannalta. Tätä varten tavoitteena oli selvittää Androidille saatavilla olevien pelimoottoreiden ja fysiikkamallinnusmoottoreiden laatua ja toimintaperiaatetta.

Ennen projektin alkua oli tehty alustavaa selvitystä ja todettu ilmainen AndEngine kiinnostavimmaksi pelimoottorikandidaatiksi ja yhtä lailla ilmainen Box2D -ohjelmisto hyvin projektin tarpeisiin soveltuvaksi fysiikka- ja törmäysmallinnusmoottoriksi. Projektin myötä oli tavoitteena tuottaa pelille määritysdokumentaatiota sekä kuvia, äänitehosteita ja videoita käytettäväksi lopullisessa tuotteessa. Nämä määritysdokumentit ja media-elementit sovittiin määriteltäviksi luottamuksellisiksi.

6.1 Saavutetut tulokset

Projekti aloitettiin projektisuunnitelman laatimisella jossa määriteltiin opinnäytetyöprojektin alustava rajausta, motiivi, aikataulukaus ja listattiin projektin kannalta oleelliset riskit. Työ jaettiin projektisuunnitelmassa siten, että selvittelevä teoriaosuus valmistuu ennen kuin oleellinen määritystyö aloitetaan. Motiivina oli Androidin verrattainen uutuus kehitysalustana kirjoittajalle.

Alkuperäiseen projektisuunnitelmaan sisältyi laajemman suunnitteludokumentin toteuttaminen niin, että sovelluskehitys aloitetaan jo ennen opinnäytetyön valmistumista. Projektin aikana työn painopiste on kuitenkin siirtynyt enemmän

toimintojen ja elementtien määrittelyyn ja lopputuloksena on ohjesäännöt ja rakennuspalikat sovelluskehitystä varten.

Projektin aikana tuotettu konseptidokumentti sisältää pelin idean määritelmän ja pyrkii kategorisoimaan pelin johonkin tunnistettavaan aihepiiriin. Pelin konseptidokumentti tarjoaa nopean markkinoinnillisen katsauksen peliin ja sen potentiaaliseen kohdeyleisöön ja markkina-alueeseen. Konseptidokumentin tärkeä tehtävä on viestiä lukijalle minkälaisesta pelistä on kyse ja minkälaisia tuntemuksia sen on tarkoitus herättää pelaajassa.

Projektin lopputulosten sydän on määrittelydokumentti joka sisältää kaikki pelin toiminnallisuuksien ja eri elementtien määrittelyt. Dokumentti kuvaa pelin rajauksen ja suunnittelun ohjesäännöt sekä määrittelee ominaisuudet ja käyttötarkoitukset pelin pelihahmoille, vihollisille, grafiikoille ja äänille. Määrittely antaa ohjeistuksen siitä minkälaista grafiikkaa peliin tuotetaan ja miten ne on tarkoitus esittää pelaajalle. Määrittely antaa myös merkityksen pelin sisällölle tarjoamalla pelille juonen, hahmoille taustatarinan sekä tarjoaa raamit sovelluskehitykselle ja audiovisuaaliselle tuottamiselle.

Pelin tekninen dokumentti asettaa pelin toteutukselle oleellisille elementeille tekniset rajaukset kuten vaadittavat tiedostomuodot ja koot kuville, pituudet äänitehosteille ja määrittää mitä kolmannen osapuolen tekniikoita pelissä tulee käyttää niin pelimoottorin, fysiikkamallinnuksen kuin audiovisuaalisen sisällön esityksen suhteen.

Projektin lähtökohtainen aikataavoite oli saavuttaa lopullinen hyväksyntävaihe jo toukokuussa joka osoittautui ensimmäisessä seurantalaverissa mahdottomaksi saavuttaa, joten takarajaa siirrettiin elokuuhun. Alkuperäisen suunnitelman mukaan projekti pyrittiin saattamaan valmiiksi noin kahdessa kuukaudessa. Projektin kirjallisesta osuudesta ehdittiin kuitenkin saada valmiiksi noin 80 % toukokuun viimeinen päivä pidettyyn seurantalaveriin mennessä. Tuohon päivään mennessä kuitenkin mitään audiovisuaalista tuotantoa ei oltu vielä ehtiä aloittaa joitakin hahmotelmakuvia lukuun ottamatta.

Projektin aikana tuotetut konsepti- määrittämis- ja tekninen dokumentti mahdollistavat kirjoittajan käsityksen mukaan ohjelmistosuunnittelun ja -kehityksen aloittamisen niin pian kuin mahdollista. Perehtyminen mobiilialustoihin tarkemmin sekä erityisesti Androidiin kehitysalustana lisäsi osaamista alustoista sekä niiden hyödyistä ja haitoista yleisesti. Erityisesti AndEngine-pelimoottoriin ja Box2D-fysiikkamallinnusohjelmistoon perehtyminen tarjosi kullannarvoista tietoa projektin toteutusvaiheen kannalta.

Projektin aikana käytetty teoriatausta pohjautui pitkälti verkkolähteisiin, joista osa erityisesti AndEnginen keskittyvistä teksteistä oli hajautettu pitkälti erinäisiin kirjoituksiin AndEnginen keskustelupalstalla. Tämä teki tiedon keräämisen haastavammaksi, mutta samalla tarjosi mahdollisuuden koota hajallaan oleva tieto yhteen koottuun teokseen. Huomioitavaa on, että tietyistä aihepiireistä saatavilla olevan lähdetiedon vähyyden vuoksi eri suosituksiin tulee suhtautua kriittisesti. Erityisesti lähdetäessä toteuttamaan peliä täytyy huomioida, että malliesimerkkejä ei välttämättä ole riittävästi, jotta niitä tulisi pitää ehdottomina totuuksina tai parhaina menettelytapoina.

Opinnäytetyön aloittaminen tutkivalla osuudella laajensi näkökulmaa mobiilimarkkinoihin yleisesti ja mahdollisti kriittisemmän arvioinnin peliprojektin valintojen, kuten alustan ja käytettävien tekniikoiden suhteen. Suositus jatkaa eteenpäin Android-alustalla pysyy huolimatta muiden alustojen mahdollisista tietyistä eduista. Selvityksen perusteella voidaan todeta, että jokaisella alustalla on tietyt vahvuudet ja heikkoudet ja jokainen tarjoaa puitteet pelin menestymiseen.

Työn selvityksen tuloksista voidaan todeta, että Androidilla menestyäkseen pelin on kannattavaa tukeutua pääsääntöisesti mainostuloihin ja mahdollisiin maksullisiin lisäsisältöihin. Projektin aikana selvinneet markkinaosuudet ja käyttäytymisjakaumat ovat jo tässä vaiheessa vaikuttaneet muun muassa peliprojektin tulomallin ja markkinoinnin suunnitteluun.

Pelin määrittelyosuuteen on käytetty mallina Micah Hrehovcsikin luomaa pelisuunnitteluun erikoistuvaa dokumentaatiomallia, jota on mukautettu soveltumaan tämän opinnäytetyön ja peliprojektin tarpeisiin. Malli mahdollisti pelin määrittelyn jakamiseen kolmeen erilliseen dokumenttiin siten, että kullakin dokumentilla on oma käyttötarkoitus ja ne toimivat pitkälti erillään toisistaan.

Toimeksiantaja on tyytyväinen projektin tähänastisiin tuotoksiin ja on jo käyttänyt syntyneitä dokumentteja apunaan peliprojektin eteenpäin viemisessä. Projektin myötä tuotetut graafikka- ja äänielementit on todettu kelpaavan käytettäväksi lopullisessa pelissä kun sen kehitys saadaan käyttöön. Toimeksiantaja on tyytyväinen opinnäytetyöprosessin läpivientiin ja ohjaukseen, sekä kokee, että opinnäyte on yrityksen kannalta hyödyllinen.

6.2 Kustannukset

Projekti on toteutettu opinnäytetöille varatun 400 tunnin puitteissa, eikä projektin aikana esiintynyt mitään muita kuluja. On kuitenkin huomioitavaa, että projektin audiovisuaaliseen tuotantoon käytettävät ohjelmistot, laitteet ja instrumentit olisivat tuottaneet mittavan kulun projektille, ellei projektin tekijällä olisi ollut näille valmiutta ennestään.

Osa ohjelmistoista olisi ollut mahdollista korvata ilmaisohjelmilla³⁶ tai avoimen lähdekoodin ohjelmilla, mutta tämä olisi vaatinut perehtymisen aikaisemmin tuntemattomampiin ohjelmistoihin. Esimerkiksi 2D-kuvankäsittely olisi voitu toteuttaa Gimpillä ja 3D-mallinnus sekä -animaatio Blenderillä. Myös äänituotantoon on saatavilla ilmaisia ohjelmistoratkaisuja, mutta erityisesti laadukkaiden ääninäytteisiin perustuvien virtuaali-instrumenttien löytäminen olisi saattanut olla haastavampaa.

³⁶ Käännön englanninkielisestä termistä *freeware* joka on ilmaisesti jaettava ohjelma jonka mukana ei kuitenkaan toimiteta lähdekoodia (vrt. Avoimen lähdekoodin ohjelma)

6.3 Kehittämisehdotukset

Opinnäytetyöprojektin päättämisen jälkeen kaikki dokumentaatio luovutetaan Yrityksen X käytettäväksi ohjelmistosuunnittelussa. Tällöin yritykselle on tarpeellista saada turvattua ohjelmistosuunnittelulle tarpeelliset henkilöresurssit, jotta kehitys saadaan käyntiin. Vaatimuksena toteutukselle Android-alustalla on vähintään vahva Java-ohjelmoinnin kokemus sekä ymmärrys Androidin SDK:hon.

Projektin aikana on myös selvitelty erikseen muita mobiilialustoja mahdollisina kehitysympäristöinä, erityisesti Microsoftin Windows Phone ja Applen iOS -alustaa. Windows Phonen kiinnostavuus pohjautuu alustan tuoreuteen, sekä vielä toistaiseksi melko väljäksi markkina-alueeksi. Tämän myötä esimerkiksi pelin saaminen näkyville muiden alustan pelien joukkoon saattaisi olla helpompaa, kuin jo useamman vuoden ikäisellä Androidilla ja iOS:llä.

Applen iOS:n kiinnostavuus perustuu projektin teoriaosuudessa esille tulleeseen markkinatutkimukseen, jonka mukaan Applen iOS-laitteen omistajat käyttävät laitetta erityisen paljon pelaamiseen suhteessa muihin alustoihin. Lisäksi iOS-käyttäjät ostavat markkinatutkimuksen mukaan maksullisia ohjelmistoja esimerkiksi huomattavasti Android-käyttäjiä herkemmin. Applen johtava asema tabletti-markkinoilla tarjoaisi potentiaalisen suurempaan laitekokoon soveltuvan version pelistä. Projektin aikana Android on kuitenkin pysynyt vahvimpana suosikkina alustavalinnaksi, joten myöskään tämän opinnäytetyön Android-keskeisyyttä ei nähty tarpeelliseksi muuttua.

Androidin kanssa etenemisessä on kuitenkin hyvä ottaa huomioon tämän opinnäytetyön aikana selvitetty tekijät markkinaosuuksista ja käyttäjäkunnasta. Selvitettyjen asioiden myötä suositellaan Android-pelin toteuttamista ilmaisversiona ja rakentamalla tulomalli mainostuloihin pohjautuvaksi. Harkinnan alaisena voidaan pitää myös freemium-mallia, jolla pelin perusversio on ilmainen ja lisää sisältöä on ladattavissa pieniä maksuja vastaan. Näitä maksullisia lisäsisältöjä voi olla esimerkiksi lisäkentät, hahmot ja hahmojen erikoisominaisuudet.

Mikäli tämän opinnäytetyön myötä käynnistyvässä ohjelmistoprojektissa kuitenkin valitaan joku toinen mobiilialusta, on suuri osa tämän raportin selvitystyöstä kuitenkin merkityksellistä myös Windows Phonelle ja iOS:lle. Esimerkiksi jo varhaisessa vaiheessa fysiikka- ja törmäysmallinnukseksi valittu Box2D on käytettävissä myös näille kahdelle mobiilialustalle. Lisäksi pelin toteutuksen kannalta oleelliset konsepti-määrittely- ja tekninen dokumentti pätee myös muilla alustoilla tai on helposti mukautettavissa näille alustoille.

6.4 Oppimisen arviointi

Tämä opinnäytetyö on toteutettu HAAGA-HELIAN opinnäytetyöprosessin mukaisesti ja on sisältänyt kaikki tämän prosessin osat. Projekti on ollut kirjoittajalle toistaiseksi vaativin itsenäisesti toteutettu projekti.

Motiivi projektin toteuttamiseksi oli erittäin hyvä, sillä ideaa mobiilipelin toteuttamisesta oli opinnäytetyöprojektin aloittamishetkellä käsitelty jo useita kuukausia ja alustavia ideoita oli kirjattu ylös. Tällöin oli selkeä valinta tehdä opinnäytetyö tästä aiheesta, sillä peliprojekti oli siinä vaiheessa, että se vaati perustavanlaatuista selvitystä alustavalinnasta, sekä jo siinä vaiheessa potentiaalisiksi todettujen kolmansien osapuolien teknologioiden, tässä tapauksessa Andenginen ja Box2D:n, toimintaperiaatteesta. Peliprojekti tarvitsi lisäksi strukturoidun määrittelytyön toteuttamista, jotta alustetut ideat saatiin jalostettua toimimaan valituilla teknisillä ratkaisuilla.

Opinnäytetyöprojektin mielenkiintoisin osuus oli määrittelytyön toteuttaminen, joka on sisällytetty erillisiin määrittelydokumentteihin, eikä sisälly tähän opinnäytetyöraporttiin. Määrittelytyön aikana tuotettiin peliä varten konsepti- ja määrittelydokumentti, sekä tekninen dokumentti, jotka on toteutettu Micah Hrehovcsikin luoman dokumenttimallia mukauttamalla. Dokumenttimalli toimi hyvin pelikehitykseen soveltumaan kevyempään dokumentointityyliin. Malli ei kaikilta osin vastaa HAAGA-HELIAssa aikaisemmilla kursseilla käytettyjä vaatimus-, määrittely- ja suunnitteludokumentteja, joskin kokemus niiden parissa työskentelystä tuki tämän projektin toteuttamista.

Projektin aikana perehdyttiin erityisen paljon Androidiin kehitysympäristönä ja sille saatavilla oleviin ohjelmistoratkaisuihin. Todennäköistä oli, että Androidin Java-sidonnaisuus teki alustasta huomattavasti muita helpommin lähestyttävän johtuen HAAGA-HELIAssa saadun Java-kokemuksen myötä. Osa selvitystyöstä vaati esimerkiksi Java-pakettien tarkastelua Eclipsessä, joka oli entuudestaan tuttu, eikä vaatinut uuden kehitysympäristön käyttöliittymän ja toimintojen opettelua.

Epäilemättä suurimpana hyötynä projektin toteutumisen kannalta oli opintojen aikana toteutettu aikaisempi määrittäytyö, joka oli jo vuosia sitten luonut pohjan ohjelmistojen toteutukselle vaadittavalle työlle. Tätä kokemusta on sittemmin harjaannutettu työelämässä, jossa työtehtävät pyörivät pitkälti vastaavanlaisten määrittäytöiden ympärillä sisältäen uusien palvelujen ja järjestelmien toteutusta ja läpivientä. Tämän opinnäytetyöprojektin myötä tämänlainen työ sai uudenlaista näkemystä.

Haasteellista projektin toteuttamisessa oli sen lähtökohtainen vaatimus saada koko työ toteutettua vuoden 2012 ensimmäisen puoliskon puolella. Aihe-ehdotusta ja projektisuunnitelmaa alustaessa tavoite oli päästä aloittamaan projekti jo heti alkuvuodesta, jolloin alkuperäinen aikataulu olisi ollut realistinen. Aloituksen myöhästymisen myötä projektilla oli kuitenkin yhä paine saada työ valmiiksi kevään aikana, vaikka käytettävä aika olikin vähentynyt kahdella kuukaudella. Määräajan siirtäminen toisessa seurantalaverissa kesälle antoi projektille muutaman kuukauden lisäaikaa ja mahdollisti täten sen valmistumisen halutulla laadulla.

Projektin myötä on lisäksi saavutettu hyötyjä, jotka ei ollut alun perin tavoitteissa, kuten huomattavasti parempi yleiskäsitys mobiilialustoista ja niiden ympärille rakentuneista markkinoista. Tietolähteenä toimineet lukuisat verkkojulkaisut ja blogit tarjosivat mielenkiintoista luettavaa projektin teoriaosuuden aikana.

7 Yhteenveto

Yhteenvetona voidaan todeta, että projekti toteutti ne merkittävät osa-alueet mitä oli tarkoituskin. Projektia aloittaessa tavoitteena oli myös laajempi ohjelmistosuunnittelu ja ohjelmateknisten toiminnallisuuden määrittely, mutta rajausta muutettiin projektin alkupuoliskolla, kun oli huomattavissa, että työ venyisi muutoin liian pitkäksi.

Projektin myötä voidaan todeta, että Android soveltuu hyvin alustana mobiilipelikehitykselle ja pitää sisällään aktiivisen kehittäjäkunnan, joka jakaa avoimesti ideoita ja neuvoja hyvinkin spesifisten tilanteiden ratkaisemiseksi. Voidaan myös todeta, että kaikilla mobiilialustoilla on jokin etu kilpailijoihinsa nähden, mutta lopullinen alustavalinta riippuu lähinnä projektin rajauksesta ja sen parissa työskentelevien ohjelmistosuunnittelijoiden alustapreferenssistä. Ovet ovat täten avoimena tehdä mobiilipelikehitystä millä tahansa suosituista alustoista.

Lähteet

Ahonen, T. 2011. Smartphone Penetration Rates by Country! We Have Good Data (finally). Luettavissa: <http://communities-dominate.blogs.com/brands/2011/12/smartphone-penetration-rates-by-country-we-have-good-data-finally.html>. Luettu 29.5.2012.

AndEngine Wiki, 2012. AndEnginePhysicsBox2DExtension. Luettavissa: <http://wiki.andengine.org/AndEnginePhysicsBox2DExtension>. Luettu 15.4.2012.

Android developers, 2012a. What is Android?. Luettavissa: <http://developer.android.com/guide/basics/what-is-android.html>. Luettu 6.2.2012.

Android developers, 2012b. Application fundamentals. Luettavissa: <http://developer.android.com/guide/topics/fundamentals.html>. Luettu 7.2.2012.

Android developers, 2012c. Platform versions. Luettavissa: <http://developer.android.com/resources/dashboard/platform-versions.html>. Luettu 7.2.2012.

Android developers, 2012d. OpenGL. Luettavissa: <http://developer.android.com/guide/topics/graphics/opengl.html>. Luettu 11.4.2012.

Android developers, 2012e. JNI Tips. Luettavissa: <http://developer.android.com/guide/practices/design/jni.html>. Luettu 15.4.2012.

Android Market, 2012. Luettavissa: <https://market.android.com>. Luettu 7.2.2012.

Angel, 2011. AndEngine Quick Start Guide. Luettavissa: <http://andenginefromscratch.blogspot.com>. Luettu 14.4.2012.

Apple, 2012. Developer. Luettavissa: <https://developer.apple.com/>. Luettu 12.7.2012.

Blandford, R. 2012. Windows Phone Marketplace passes 70,000 apps. Luettavissa: http://allaboutwindowsphone.com/news/item/14316_Windows_Marketplace_passes_700.php. Luettu 25.3.2012.

Box2D Forums, 2012. Box2D Ports. Luettavissa: <http://box2d.org/forum>. Luettu 15.4.2012.

Brian, M. 2012. Apple hits 25 billion App Store downloads. Luettavissa: <http://www.itpro.co.uk/612598/timeline-the-apple-app-stores-first-year>. Luettu 25.3.2012.

Catto, E. 2011a. FAQ. Luettavissa: <http://code.google.com/p/box2d/wiki/FAQ>. Luettu: 15.4.2012.

Catto, E. 2011b. Box2D v2.2.0 User Manual. Luettavissa: <http://box2d.org/manual.pdf>. Luettu 15.4.2012.

Chow, Kevin. 2007. Quick Introduction to Modules. Luettavissa: <http://modarchive.org/index.php?article-modules>. Luettu 14.4.2012.

Dediu, H. 2012. App developers receive \$12 for each iOS device sold. Luettavissa: <http://www.asymco.com/2012/02/19/app-developers-get-12-for-each-ios-device-sold>. Luettu 25.3. 2012.

Etisbew, 2012. Mobile Software Development Overtaking Desktop Software Development. Luettavissa: <http://www.etisbew.com/articles/index.php/mobile-software-development-overtaking-desktop-software-development/>. Luettu 17.7.2012.

Fishback, B. 2012. 8 Mobile Trends For Small Businesses To Watch In 2012. Luettavissa:

http://www.huffingtonpost.com/bo-fishback/mobile-trends-small-businesses-b_1234524.html?ref=technology&ir=Technology. Luettu 31.1.2012.

Google, 2012. Android Developer Console. Luettavissa:

<https://play.google.com/apps/publish/signup>. Luettu 12.7.2012.

Gramlich, N. 2010. AndEngine – Core-Terminology. Luettavissa:

<http://www.andengine.org/forums/tutorials/andengine-core-terminology-t316.html>.

Luettu 2.4.2012.

Gramlich, N. 2011a. AndEngine - Live-Wallpaper Extension. Luettavissa:

<http://code.google.com/p/andenginelivewallpaperextension>. Luettu 14.4.2012.

Gramlich, N. 2011b. AndEngine - .MOD-Player Extension. Luettavissa:

<http://code.google.com/p/andenginemodplayerextension>. Luettu 14.4.2012.

Gramlich, N. 2011c. AndEngine - Multiplayer Extension. Luettavissa:

<http://code.google.com/p/andenginemultiplayerextension>. Luettu 14.4.2012.

Gramlich, N. 2011d. AndEngine - MultiTouch Extension. Luettavissa:

<http://code.google.com/p/andenginemultitouchextension>. Luettu 14.4.2012.

Gramlich, N. 2011e. AndEngine - Physics Box2D Extension. Luettavissa:

<http://code.google.com/p/andenginephysicsbox2dextension>. Luettu 14.4.2012.

Gramlich, N. 2011f. AndEngine - Augmented-Reality Extension. Luettavissa:

<http://code.google.com/p/andengineaugmentedrealityextension>. Luettu 14.4.2012.

Gramlich, N. 2011g. AndEngine - SVGTextureRegion Extension. Luettavissa:

<http://code.google.com/p/andenginesvgtextureregionextension>. Luettu 14.4.2012.

Gramlich, N. 2011h. AndEngine - Texture Packer Extension. Luettavissa: <http://code.google.com/p/andenginertexturepackerextension>. Luettu 14.4.2012.

Gramlich, N. 2011i. AndEngineTexturePackerExtension. Luettavissa: <http://www.andengine.org/forums/updates/andenginertexturepackerextension-t4334.html>. Luettu 14.4.2012.

Hepburn, A. 2011. Infographic: Mobile Statistics, Stats & Facts 2011. Luettavissa: <http://www.digitalbuzzblog.com/2011-mobile-statistics-stats-facts-marketing-infographic/>. Luettu 29.5.2012.

Humpries, M. 2011. Box2D creator asks Rovio for Angry Birds credit at GDC. Luettavissa: <http://www.geek.com/articles/mobile/box2d-creator-asks-rovio-for-angry-birds-credit-at-gdc-2011032>. Luettu 15.4.2012.

Ince, D. 2009. A Dictionary of the Internet. Luettavissa: <http://www.oxfordreference.com/views/ENTRY.html?subview=Main&entry=t12.e2117>. Luettu 7.2.2012.

Isaac, M. 2011. Android Market Hits 10 Billion Downloads, Kicks Off App Sale. Luettavissa: <http://www.wired.com/gadgetlab/2011/12/android-market-downloads>. Luettu 25.3.2012.

Khronos Group, 2012a. OpenGL - The Industry's Foundation for High Performance Graphics. Luettavissa: <http://www.khronos.org/opengl>. Luettu 11.4.2012.

Khronos Group, 2012b. OpenGL ES - The Standard for Embedded Accelerated 3D Graphics. Luettavissa: <http://www.khronos.org/OpenGL>. Luettu 11.4.2012.

Liang, S. 2002. The Java™ Native Interface Programmer's Guide and Specification. Luettavissa: <http://java.sun.com/docs/books/jni/html/titlepage.html>. Luettu 15.4.2012.

- LibGDX, 2012. Luettavissa: <http://code.google.com/p/libgdx>. Luettu 15.4.2012.
- Liedtke, M. 2012. Google Play Store: Android Market Rebranded, Redesigned. Luettavissa: http://www.huffingtonpost.com/2012/03/06/google-play-store-android-market_n_1324528.html. Luettu 25.3.2012.
- Löw, A. TexturePacker. Luettavissa: <http://www.texturepacker.com>. Luettu 14.4.2012.
- Microsoft. 2012. What Is a Sprite? Luettavissa: <http://msdn.microsoft.com/en-us/library/bb203919.aspx>. Luettu 14.4.2012.
- MSDN, 2012. Application Platform Overview for Windows Phone. Luettavissa: [http://msdn.microsoft.com/en-us/library/ff402531\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/ff402531(v=vs.92).aspx). Luettu 17.7.2012.
- Murphy, M. 2010. Beginning Android 2. Luettu 1.4.2012.
- Niemeläinen, J. 2012. Taloussanomat: Suomi löysi taas Nokian älypuhelimet. Luettavissa: <http://www.taloussanomat.fi/yrittaja/2012/05/26/suomi-loysi-taas-nokian-alypuhelimet/201230161/137>. Luettu 16.7.2012.
- Reed, B. 2010. A Brief History of Smartphones. Luettavissa: http://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html. Luettu 29.5.2012.
- Reisinger, D. 2012. Mobile app revenue set to soar to \$46 billion in 2016. Luettavissa: [http://news.cnet.com/8301-13506_3-57379364-17/mobile-app-revenue-set-to-soar-to-\\$46-billion-in-2016/](http://news.cnet.com/8301-13506_3-57379364-17/mobile-app-revenue-set-to-soar-to-$46-billion-in-2016/). Luettu 12.7.2012.
- Rowinski, D. 2012. [Infographic] History of Mobile App Stores. Luettavissa: http://www.readwriteweb.com/archives/infographic_history_of_mobile_app_stores.php. Luettu 29.5.2012.

- Rubio, J. 2011. The History of Android. Luettavissa: <http://au.ign.com/articles/2011/12/22/the-history-of-android>. Luettu 6.2.2012.
- Scott, J. 2009. Timeline: The Apple App Store's first year. Luettavissa: <http://www.itpro.co.uk/612598/timeline-the-apple-app-stores-first-year>. Luettu 25.3.2012.
- Shelor, W. 2011. Getting Started with AndEngine. Luettavissa: <http://andengineguides.wordpress.com/category/getting-started-with-andengine>. Luettu 14.4.2012.
- StatCounter, 2012. Luettavissa: <http://gs.statcounter.com>. Luettu 25.3.2012.
- Swallow, E. 2011. 5 Trends Shaping the Mobile Gaming Industry. Luettavissa: <http://mashable.com/2011/07/13/mobile-gaming-trends>. Luettu 1.2.2012.
- Turner, A. 2012. Extreme JNI Performance. Luettavissa: <http://nerds-central.blogspot.fi/2012/04/extreme-jni-performance.html>. Luettu 19.8.2012.
- W3C. 2012. What is SVG? Luettavissa: <http://www.w3.org/Graphics/SVG>. Luettu 14.4.2014.
- XMP. 2012. Extended Module Player. Luettavissa: <http://xmp.sourceforge.net>. Luettu 14.4.2012.
- Xyologic, 2011. The Top 25 iPhone and Android App Publishers in 2011 – iPhone is for Games, Android is for Apps. Luettavissa: <http://www.xyologic.com/blog/the-top-25-iphone-and-android-app-publishers-in-2011-iphone-is-for-games-android-is-for-apps>. Luettu 25.3.2012.