



UNITY IPAD-PELIKEHITYKSESSÄ

Essi Suurkuukka

Opinnäytetyö
Joulukuu 2012
Tietojenkäsittelyn koulutus-
ohjelma
Digimedia

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Digimedia

ESSI SUURKUUKKA:
Unity iPad-pelikehityksessä

Opinnäytetyö 48 sivua
Joulukuu 2012

Opinnäytetyön tavoitteena oli tutkia pelikehitysohjelma Unityn soveltuvuutta 3D-pelikehitykseen iPad-laitteelle. Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa Unityllä iPad-laitteelle 3D-peli, joka hyödyntäisi olemassa olevissa peleissä käytettyjä tekniikkoja ja menetelmiä.

Tavoitteen saavuttamiseksi tutkittiin olemassa olevia 3D-pelejä, jotka oli suunniteltu iPadille. Peli suunniteltiin ja toteutettiin ottaen huomioon näissä peleissä käytetyt ratkaisut tehden samalla huomioita Unityn toiminnasta. Työssä esitellään Unityn käyttöä teoriassa ja itse suunnitellun pelin avulla.

Työn aikana Unitystä muodostui hyvä yleiskuva pelikehityksen työkaluna. Unity osoitautui helppokäyttöiseksi, tehokkaaksi ja edistyneeksi ohjelmaksi, joka hoitaa lähes kaiken tarvittavan optimoinnin kohdeympäristöön automaattisesti. Näin pelikehittäjä voi keskittyä olennaiseen, eli pelikokemuksen parantamiseen.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Digital Media

ESSI SUURKUUKKA:
Unity In iPad Game Development

Bachelor's thesis 48 pages
December 2012

The goal of this thesis was to investigate the suitability of the game development program Unity in 3D game development for the iPad device. The objective was to design and implement a 3D game which would make use of techniques and methods used in existing games.

To achieve this goal, existing 3D games which were developed for the iPad device were investigated. While designing the game, the solutions used in these games were taken into account and observations of Unity's operation were made. This thesis explains Unity in theory and by utilizing the game designed during the thesis work process.

During the process a good general view of Unity as a game development tool was made. Unity proved to be easy to use, effective and advanced. It automatically takes care of almost all things necessary to optimize the game to the target environment. Thus, game developers can concentrate on what really matters: improving the game experience.

Key words: unity, ipad, 3d, game development

SISÄLLYS

1	JOHDANTO.....	7
2	MOBIILILAITTEET JA -PELIT	9
2.1	Mobiililaitteiden määritelmä.....	9
2.2	Mobiililaitteiden historia.....	9
2.3	Mobiililaitteiden nykytila	11
2.4	Mobiilipelin määritelmä	13
2.5	Mobiilipelien historia.....	13
2.6	Mobiilipelien nykytila.....	14
3	IPAD.....	16
3.1	Yleistä iPadista	16
3.2	Tekniset ominaisuudet	17
3.3	iPad-ohjelmointi.....	18
4	UNITY.....	19
4.1	Miksi Unity?	19
4.2	Yleistä Unitystä.....	19
4.3	Käyttöliittymä	21
4.4	Assetit ja GameObjectit	22
4.5	Scenet.....	23
4.6	Prefabit.....	24
4.7	Skriptaus	24
5	MENETELMÄT	26
5.1	Työvälineet	29
5.2	Vaatimukset	30
6	PELIN TOTEUTUS	31
6.1	Pelin genre ja idea.....	31
6.2	3D-mallit.....	31
6.2.1	Mallien suunnittelu	31
6.2.2	Valmiiden mallien käyttö.....	32
6.2.3	Tekstuurit ja materiaalit	32
6.2.4	Nivelet.....	33
6.2.5	Animaatiot.....	33
6.2.6	Mallien vienti Unityyn	33
6.3	Pelihahmon ohjaus	34
6.4	Animaatioiden hallinta.....	34
6.5	Pelimaailma	35
6.5.1	Colliderit	36

6.6 Käyttöliittymä	37
6.7 Äänet	39
6.8 Julkaisu	40
7 TULOKSET	41
8 JOHTOPÄÄTÖKSET	43
LÄHTEET	44

ERITYISSANASTO

CSS	Cascading Style Sheets. Tyylitiedostokieli, jonka avulla voidaan määrittää www-sivujen ulkoasu.
FPS	Frames Per Second, kehysnopeus. Kehysnopeudella tarkoitetaan sekunnissa ruudulle piirrettävien kuvien määrää.
HTML	HyperText Markup Language. Www-sivujen kuvauskieli.
mesh	3D-mallin muodostava polygoniverkko.
metodi	Ohjelman itsenäinen suoritettava osa, jota voidaan kutsua toisalta ohjelman sisällä.
olio	Olio-ohjelmoinnissa käytettävä luokan ilmentymä, jolla on omat toimintonsa.
polygoni	Kolmesta pisteestä koostuva alue. 3D-mallit koostuvat toisiinsa kiinnitetyistä polygoneista.
rajapinta	Ohjelmoinnissa tarjottu kokoelma kirjastoja, jonka avulla ohjelma voi kommunikoida toisen ohjelman kanssa.
syöttötapa	Tapa antaa komentoja tietokoneelle tai laitteelle reaali-ilmasta, esimerkiksi hiiren klikkaus, näppäimistö, äänikomennot.
verteksi	Piste, joka muiden pisteiden kanssa koostaa 3D-mallin.

1 JOHDANTO

Mobiililaitteista on tullut viime vuosina hyvin suosittuja kuluttajien keskuudessa ja yhä useampi kuluttaja omistaa esimerkiksi älypuhelimien ja/tai taulutietokoneen. Nykyaikaiset älypuhelimet ja taulutietokoneet ovat varustettu langattomilla yhteyksillä, joiden avulla muun muassa internet-selailu ja sovellusten lataus on helppoa ja vaivatonta. Varsinkin mobiililaitteilla pelaaminen on arkipäiväistänyt ja mobiilipelien tarjonta on lisääntynyt runsaasti.

Mobiililaitteiden suuri kysyntä on tuonut mukanaan runsaasti uusia mobiililaitteita eri valmistajilta. Pelikehityksen näkökulmasta tämä luo haasteita, sillä eri laitevalmistajat tukevat erilaisia käyttöjärjestelmiä, laitetekniikoita ja kehitysympäristöjä. Mikäli sama peli halutaan julkaista erityyppisille laitteille, joudutaan peli yleensä toteuttamaan käytännöllisesti katsoen alusta asti jokaiselle eri alustalle, mikä on aikaavievää ja kallista ja saattaa johtaa jopa julkaisusta luopumiseen sekä tietyn laitteen asiakaskunnan menettämiseen. Laitteiden omat kehitysympäristöt eivät myöskään ole erikoistuneet erityisesti pelikehitykseen vaan niillä voi luoda monentyyppisiä sovelluksia.

Pelikehityksessä olisi hyödyllinen ohjelma, joka veisi pelikehityksen ylemmälle tasolle, jolloin peliä ei tarvitsisi tehdä alusta asti eri alustalle ja joka olisi suunniteltu erityisesti pelituotantoon. Pelien toteutus usealle eri alustalle onnistuu esimerkiksi tässä työssä esittelemälläni ohjelmalla Unity, joka on 3D-pelien toteuttamiseen erikoistunut työkalu. Työn kohdelaitteena toimivan iPadin kehittyneet grafiikkaominaisuudet mahdollistavat 3D:n tukemisen, joten päätin selvittää, kuinka hyvin Unity soveltuu nimenomaisesti 3D-pelien tekemiseen iPadille.

Jokaisella opinnäytetyöllä on tavoite, joka voi olla luonteeltaan kehittävä, tutkiva tai selvittävä. Opinnäytetyöni tavoitteena on tutkia Unityn soveltuvuutta 3D-pelien toteuttamiseen iPad-laitteelle. Toimenpidettä tai toimenpiteitä, joilla tavoitteeseen päästään, kutsutaan tarkoitukseksi. Opinnäytetyöni tarkoituksena on suunnitella ja toteuttaa iPad-laitteelle 3D-peli, jonka avulla esittelen Unityn käyttöä.

Työn teoriaosuudessa tutkitaan aluksi mobiililaitteiden ja -pelien määritelmää, historiaa ja nykypäivää. Sen jälkeen esitellään Unityn toimintaperiaatteita jonka jälkeen kerrotaan

suunnittelemani pelin toteutuksesta. Lopuksi pohditaan Unityn soveltuvuutta 3D-pelien toteuttamiseen iPad-laitteelle.

2 MOBIILILAITTEET JA -PELIT

2.1 Mobiililaitteiden määritelmä

Mobiililaitteiden määritelmästä on esitetty useita eri näkemyksiä, sillä mobiililaitemarkkinat ovat kehittyneet viime vuosina niin nopeasti. Useimmiten mobiililaitteiksi luokitellaan ainakin älypuhelimet, taulutietokoneet, kämmentietokoneet ja kannettavat e-lukulaitteet.

Mobiililaitteille ominaista on niiden pieni koko, helppokäyttöisyys ja paikasta riippumaton yhteys internetiin. Useimmissa mobiililaitteissa on kosketusnäyttö, jonka avulla laitetta ohjataan ja sille annetaan komentoja. Laitteissa on yleensä myös sensori, joka tunnistaa laitteen kääntelyn. Kehittyvä tekniikka on mahdollistanut myös paremman äänikomentojen käytön. Myös korkealaatuiset kamerat ja GPS (paikkatieto) ovat vakio-ominaisuuksia eritoten älypuhelimissa ja taulutietokoneissa.

Laitteiden avulla on mahdollista olla internet-yhteyden avulla yhteydessä toisiin mistä vain. Mobiililaitteet ovat valmiita toimimaan sosiaalisessa mediassa. Laitteiden avulla käyttäjät voivat ladata laitteeseen myös uusia sovelluksia laitteiden omista kauppapaikoista.

2.2 Mobiililaitteiden historia

Motorolan Dynatac 8000x, ensimmäinen matkapuhelin, tuli myyntiin vuonna 1983. (Time 2010) Euroopassa GSM-mobiiliteknologiastandardi virallistettiin vuonna 1987. GSM mahdollisti muun muassa laadukkaat puhelut ja tekstiviestien lähettämisen. Vuonna 1992 Nokia julkaisi ensimmäisen GSM-matkapuhelimensa Nokia 1011:n. (Nokia 2012b)

1990-luvulla monet valmistajat, esimerkkeinä Nokia, Sony, Ericsson ja Motorola julkaisivat lukuisia puhelimia. Matkapuhelimet alkoivat muuttua muotoaan yhä pienemmiksi ja markkinoille tuli myös puhelimia, joissa ei ollut näkyvää antennia.

Kämmentietokoneet alkoivat yleistymään vuonna 1996, kun Palm julkisti Palm Pilot 1000:n. Markkinoilla oli tilaa mobiililaitteelle, joka sijoittuisi jonnekin kannettavan tie-

tokoneen ja matkapuhelimen väliin. (PCWorld 2010) Palm Pilot 1000:ssa oli toimiva käsialantunnistustoiminto ja sen tiedot pystyi synkronoimaan sekä PC:lle että Macille. (Time 2010) Kaksi vuotta myöhemmin julkistettiin myös Nokian 9000 Communicator – puhelin, jossa oli muun muassa internet-selain, faksi ja sähköposti.

Internetiin yhdistäminen matkapuhelimella jätti aluksi puhelinyhteyden auki, mutta 1990-luvun lopulla julkaistu WAP (Wireless Application Protocol) mahdollisti ensimmäisen kerran edullisen langattomaan internetiin yhdistämisen. WAPin avulla kuluttajat pystyivät siirtämään matkapuhelintensa pelien pistetuloja etäpalvelimille ja tilaamaan uusia soittoääniä. WAPin etuna oli myös se, että siirto perustui kuluttajalle edullisempaan pakettisiirtoon. WAP mahdollisti myös kännyköillä pelattavat moninpelit ja uusien pelien lataamisen palvelimilta. (Business Insider 2012, Tekniikka & Talous 2012, The Entertainment Software Association 2012)

2000-luvulla matkapuhelinten kehitys jatkui. Ensimmäiset väripuhelimet tulivat markkinoille ja puhelimet alkoivat tukemaan myös MP3-äänitiedostoja. Matkapuhelinten muotoilu kehittyi sekä myös ensimmäinen kamerapuhelin julkaistiin vuosituhaten alussa Japanissa. Myös ensimmäinen Symbian-käyttöjärjestelmää käyttävä puhelin Ericsson R380 julkaistiin vuonna 2000. Ensimmäinen massamarkkinoille suunnattu Symbian-käyttöjärjestelmää käyttävä Nokia-puhelin Nokia 7650 julkaistiin vuonna 2001. Vuosina 2001-2006 julkaistiin sata kaupallisille markkinoille suunnattua puhelinmallia ja vuoteen 2006 mennessä oli myyty yli 100 miljoonaa Symbian-käyttöjärjestelmää käyttävää puhelinta. (Nokia 2012c)

Vuonna 2007 julkistettiin Applen iPhone, joka oli edistyksellinen puhelin erityisesti grafiikkaominaisuuksien, käyttöliittymän ja helppokäyttöisyyden osalta. Puhelin oli suuri harppaus edeltäviin matkapuhelimiin ja oli suunnannäyttäjä tuleville älypuhelinmalleille. Ensimmäinen Android-älypuhelin, HTC Dream, julkaistiin vuonna 2008, jota seurasivat lukuisat uudet Android-käyttöjärjestelmää käyttävät älypuhelinmallit eri valmistajilta. (Cnet 2011)

Windows Phone 7 –puhelimet julkaistiin Euroopassa marraskuussa 2010. (Microsoft 2010) Vuoden 2011 alkupuolella Microsoft ja Nokia ilmoittivat yhteistyöstä, jolloin Nokian ensisijainen älypuhelinstrategia olisi Windows Phone. (Microsoft 2011) Tämä tarkoitti myös Nokian luopumista Symbian-käyttöjärjestelmän kehityksestä. (Nokia

2012a) Tämän jälkeen Nokia on julkaissut useita Lumia-puhelimia. Syyskuussa 2012 Nokia ilmoitti uusista tulevista Lumia-malleista, jotka tulevat käyttämään uusinta Windows Phone 8 –käyttöjärjestelmää. (Nokia 2012b)

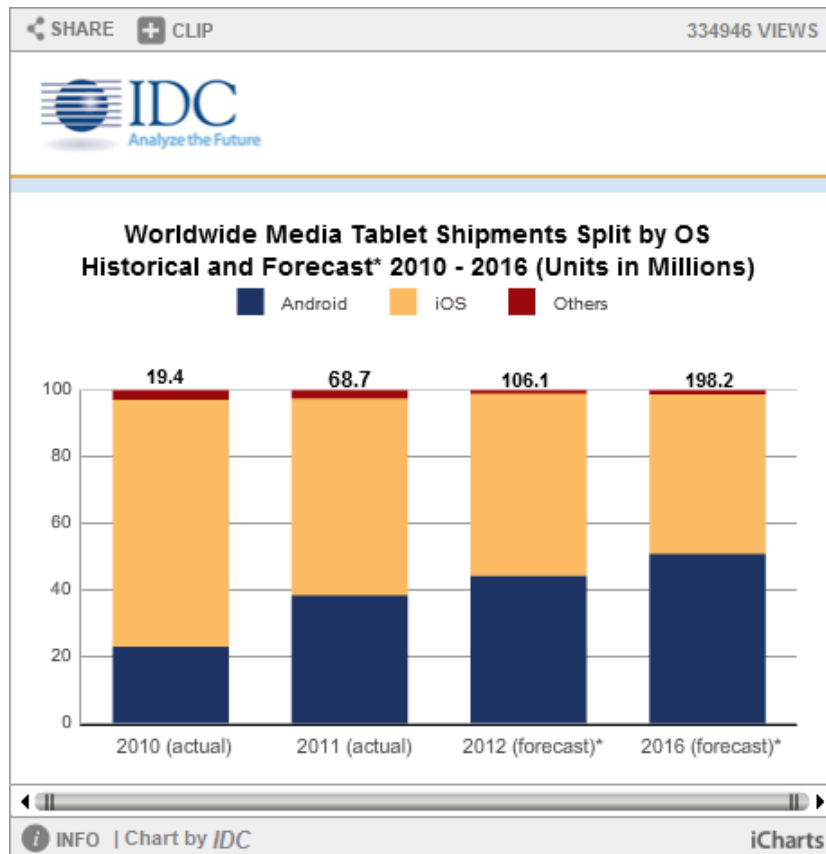
Vuonna 2010 ilmestyivät ensimmäiset taulutietokoneet. Taulutietokoneissa yhdistyi suurempi kosketusnäytön koko, helppokäyttöisyys ja laitteen keveys verrattuna kannettaviin tietokoneisiin. iPad (2010) ja Motorola Xoom (2011) olivat ensimmäisiä taulutietokoneita. Taulutietokoneista ilmestyi useita paranneltuja versioita ja myös Amazonin Kindle –e-lukulaitteen jalanjäljissä julkaistiin vuonna 2011 Android-käyttöjärjestelmää käyttävä Kindle Fire –taulutietokone. (Amazon 2011)

2.3 Mobiililaitteiden nykytila

Yhä useampi matkapuhelimen ostaja valitsee puhelimekseen älypuhelimien. Kodintekniikka-alan yhteistyöfoorumin Kotekin (2012) mukaan Suomessa myytiin tammikuu 2012 – syyskuu 2012 välillä älypuhelimia yli 1,1 miljoonaa. Älypuhelimien osuus kaikista myydyistä matkapuhelimista oli 65 %. Kuluttajat vaativat älypuhelimiltaan parempia ominaisuuksia myös edullisimmissa hintaluokissa, joten älypuhelimien keskihinta on noussut. Älypuhelimien keskihinta on ollut vuonna 2012 tammi-syyskuussa 361 euroa. (Kotek 2012)

Taulutietokoneiden myynti on kasvanut runsaasti. Tammi-syyskuussa vuonna 2012 Suomessa myytiin 160 000 taulutietokonetta. Kotekin tilastojen mukaan taulutietokoneiden myynti on lähes kolminkertaistunut vuonna 2012 verrattuna vuotta aikaisempiin myyntilukuihin. Taulutietokoneiden keskihinta on hieman laskenut 546 euroon. (Kotek 2012)

Maailmanlaajuisesti taulutietokoneita ennustetaan toimitettavan vuonna 2012 yli 100 miljoonaa kappaletta (International Data Corporation 2012). Android-taulutietokoneiden osuuden uskotaan IDC:n ennusteen (2012) mukaan kasvavan lähivuosina jopa puoleen kaikista taulutietokoneista (kuva 1).



KUVA 1. Maailman taulutietokonetoimitukset (toteutuneet ja ennusteet) jaoteltuna käyttöjärjestelmittäin. (IDC, Kuvakaappaus)

Mobiililaitteiden käyttöjärjestelmien ylivoimaisin markkinaosuus oli vuoden 2012 kolmannella vuosineljänneksellä Android-käyttöjärjestelmällä osuuden ollessa 72,4 % (kuva 2). iOS-käyttöjärjestelmän markkinaosuus oli 13,9 %. Symbian-käyttöjärjestelmän myyntiluvut laskivat vuoden 2011 16,9 % osuudesta 2,6 %:iin. (Gartner 2012)

**Worldwide Mobile Device Sales to End Users by Operating System in 3Q12
(Thousands of Units)**

Operating System	3Q12 Units	3Q12 Market Share (%)	3Q11 Units	3Q11 Market Share (%)
Android	122,480.0	72.4	60,490.4	52.5
iOS	23,550.3	13.9	17,295.3	15.0
Research In Motion	8,946.8	5.3	12,701.1	11.0
Bada	5,054.7	3.0	2,478.5	2.2
Symbian	4,404.9	2.6	19,500.1	16.9
Microsoft	4,058.2	2.4	1,701.9	1.5
Others	683.7	0.4	1,018.1	0.9
Total	169,178.6	100.0	115,185.4	100.0

Source: Gartner (November 2012)

KUVA 2. Vuoden 2012 3. vuosineljänneksen maailmanlaajuiset mobiililaitteiden myyntiluvut käyttöjärjestelmittäin jaoteltuna. (Gartner, Kuvakaappaus)

2.4 Mobiilipelin määritelmä

Mobiilipelin voidaan katsoa olevan peli, joka on erityisesti suunniteltu mobiililaitteelle ottaen huomioon laitteen asettamat vaatimukset ja rajoitukset, kuten näytön koon. Kuitenkin on monia pelejä, jotka on suunniteltu jollekin muulle alustalle, mutta joita voi pelata mobiililaitteella. Myös esimerkiksi uudet web-teknologiat kuten HTML5 hämärtävät mobiilipelien ja muiden pelien rajaa.

2.5 Mobiilipelien historia

Ensimmäiset mobiilipelit nähtiin 1990-luvun alkupuolella, kun laskinvalmistajat asensivat Snake-pelin laitteisiinsa. Vuonna 1997 matopelinäkin tunnettu Snake ilmestyi Nokian puhelimiin, mistä lähtien Snake on ollut saatavilla 350 miljoonassa puhelimessa. (Nokia 2012d)

Seuraavat viisi-kuusi vuotta olivat mobiilipelien etsikkoaikaa, kun yritykset tutkivat mobiilipelimarkkinoiden rajoja ja mahdollisuuksia. (Business Insider 2011) Vuosituhannen vaihteessa mobiilipeleihin alettiin sijoittaa yhä enemmän. Matkapuhelimien kehittyivät. Värinäyttöjen saapuminen ja Java ME:n yleistyminen jouduttivat mobiilipelien kehitystä. (The Entertainment Software Association 2012)

Mobiilipelien kehitys saavutti pisteen, jossa suuret pelijulkaisijat päättivät hypätä mukaan. Useat julkaisijat lisenssöivät suosituimmat tietokone- ja konsolipelinsä mobiilipeleiksi. (The Entertainment Software Association 2012)

Kun matkapuhelimeissa alkoi yleistyä 3D-grafiikka, peliyritykset ja -julkaisijat huomasivat pian, että parannellut grafiikkaominaisuudet eivät toimineet kaikissa puhelimissa. Samoja pelejä tehtiin erikseen valikoiduille malleille ja tähän käytettiin runsaasti aikaa ja rahaa. Painotus oli edelleen lisenssipeleissä eikä uusien pelien suunnittelussa. (The Entertainment Software Association 2012)

Mobiilipelien historian käännekohta ja mobiilipelimarkkinoiden elpyminen tapahtui vuonna 2007, kun Applen iPhone julkistettiin (Business Insider 2011). Vuonna 2008 avattu App Store mahdollisti pelien ja sovellusten helpon ja nopean lataamisen iPho-

neen. (The Entertainment Software Association 2012) App Storen avulla kehittäjät pystyivät myymään sovelluksia ja pelejä suoraan kuluttajille ilman välikäsiä.

Samana vuotena aukesi myös Googlen Android Market, josta pystyi nimensä mukaisesti lataamaan sovelluksia Android-laitteille. Vuonna 2012 Google yhdisti Android Marketin, Google Musicin ja Google eBookstoren yhdeksi täysin pilvipalvelupohjaiseksi palveluksi, joka sai nimekseen Google Play. Tuolloin sovelluksia oli saatavilla jo yli 450 000. (Google Official Blog 2012)

Vuonna 2008 ilmestyivät ensimmäiset iPhone-pelit ja Android-pelit. Vuonna 2009 Rovio julkaisi App Storessa ensimmäisen Angry Birds –pelisarjan pelin, josta tuli heti myyntimenestys. Toukokuuhun 2012 mennessä Angry Birds –pelejä oli ladattu jo yli miljardi kertaa eri laitteille. (Kauppalehti 2012, PC World 2012)

2.6 Mobiilipelien nykytila

Älypuhelimien ja taulutietokoneiden kasvanut suoritusaso on mahdollistanut yhä paremmat mobiilipelien grafiikkaominaisuudet. Uusia sovelluksia on hyvin helppo tilata käyttöjärjestelmävalmistajien omista kauppapaikoista (App Store, Google Play). Nopeat 3G- ja 4G-yhteydet ja yleistyneet langattomat verkot ovat mahdollistaneet pelien nopean lataamisen. (Business Insider 2011)

Mobiilipelimarkkinat ovat menestyneet hyvin. Marraskuuhun 2012 mennessä Google Play –palvelussa oli tarjolla jo 700 000 sovellusta (Bloomberg Businessweek 2012) ja syyskuuhun 2012 mennessä sovelluksia oli asennettu jo 25 miljardia (Official Android Blog 2012).

Vaikka suuri osa peleistä on jo itsenäisiä mobiilipelejä, voidaan mainita onnistunut lisenssipelin siirtyminen esimerkiksi The Sims –pelisarjan kohdalla. Elämäsimulaattorista laadittiin The Sims Freeplay –mobiilipeli (kuva 3), joka julkaistiin iPad, iPhone ja iPod Touch –alustoilla vuoden 2011 lopulla (Electronic Arts 2011). Pelistä on julkaistu myös Android- ja Kindle Fire -versio. Pelissä luodaan simejä, joita ohjataan tavalliseen tapaan syömään, peseytymään, harrastuksiin ja vuorovaikutukseen muiden simien kanssa. Pelissä käytetään edelleen simien omaa valuuttaa simoleoneja, mutta pelissä pystyy etenemään myös ansaitsemalla kokemuspisteitä, joiden avulla voi saada Lifestyle-pisteitä.

Lifestyle-pisteitä voi ansaita pelaamalla tai ostamalla niitä pelin sisäisestä kauppapaikasta.



KUVA 3. The Sims Freeplay (Kuvakaappaus)

Tämä ansaintamalli, jossa itse peli on ilmainen, mutta pelaamista voi jouduttaa ostamalla pelissä käytettävää valuuttaa, on yleistymässä. Perinteisesti ansaintamalli on perustunut pelin ostohinnan maksamiseen. Kuitenkin rinnalle ovat tulleet ilmaiset pelit, joissa käytetään mainontaa, kuten Angry Birds, ja pelin sisäiset ostot, joiden avulla peliin voi lisätä pelissä käytettävää valuuttaa tai ostaa muuta sisältöä, esimerkiksi uusia aseita, kenttiä tai ajoneuvoja. Myös pelien päivitykset, lisäosat ja pelaajien mahdollisuudet tuottaa omaa sisältöä pidentävät pelien elinaikaa. (Forbes 2012; Pierce 2012, 258).

3 IPAD

3.1 Yleistä iPadista

iPad on Apple-yhtiön suunnittelema taulutietokone. iPadin avulla voi muun muassa pelata pelejä, katsella elokuvia, kuunnella musiikkia ja selata internetiä.

Tammikuussa 2010 Steve Jobs julkisti ensimmäisen iPadin San Fransiscossa (Apple 2010). Vain 28 päivän kuluttua kauppojen hyllyille saapumisestaan (4.3.2010) iPadin myynti saavutti miljoonan kappaleen rajapyykin (Apple 2010). Kaikkiaan iPadeja on myyty marraskuuhun 2012 mennessä jo yli 100 miljoonaa kappaletta. (Apple 2010a, 2010b, 2011a-d, 2012a-e)

iPadin käyttöjärjestelmä on iOS. iPadin mukana tulee useita valmiiksi asennettuja ohjelmia kuten Safari-internetselain, Mail, iBooks, Maps, FaceTime ja PhotoBooth. Lisää sovelluksia iPadiin saa asennettua App Store –kauppapaikan kautta. Musiikkia ja elokuvia voi puolestaan ostaa iTunesista. (Apple, 2012f)

iPadin käyttö perustuu suureksi osaksi sormilla kosketusnäytöllä tehtäviin eleisiin, joita ovat muun muassa napautus (tap), vieritys (scroll), vetäminen (drag), pyyhkäisy (swipe), nipistys (pinch) ja kaksoisnapautus (double-tap). (Apple 2012g, Apple 2012h)

3.2 Tekniset ominaisuudet

iPad-malleja on syyskuuhun 2012 mennessä ilmestynyt yhteensä kolme kappaletta (taulukko 1). Kaikista malleista on saatavilla sekä WiFi, että Wifi + 3G-versio. Laite tukee Bluetooth-yhteyttä ja siinä on useita sensoreja: kiihtyvyyssensori, ympäröivän valon mittaussensori, etäisyysensori ja iPad 2:sta lähtien gyroskooppi. Lisäksi laitteessa on digitaalinen kompassi. Apple ilmoittaa iPadin akkukeston olevan jopa 10 tuntia. (Apple, 2012i)

TAULUKKO 1. iPadin mallivertailu (Apple 2012i, Unity 2012b, Wikipedia 2012)

	iPad	iPad 2	3. sukupolven iPad	iPad Mini
Ilmestymisvuosi	2010	2011	2012	2012
Resoluutio	1024 x 768	1024 x 768	2048 x 1536	1024x768
Näytön koko	9.7"	9.7"	9.7"	7.9"
Proessori	1 GHz Apple A4	1 GHz Apple A5	1 GHz Apple A5X	Apple A5
RAM-muisti	256 MB DDR	512 MB DDR2	1 GB DDR2	1 GB DDR2
Flash-muisti	16, 32 tai 64 GB			
Kamera	ei	kyllä, 2 kpl	kyllä, 2 kpl	kyllä, 2 kpl
Gyroskooppi	ei	kyllä	kyllä	kyllä

Piercen (2012, 137) mukaan iPadin näytössä on käytetty kapasitiivista materiaalia, jossa on sähkövaraus. Kun näyttöä kosketetaan, materiaali tunnistaa sähkövarauksen muutoksen kosketuspaikan kohdalla. Tieto välitetään prosessorille, joka määrittää kosketuksen tarkan paikan. Kosketuksia voidaan tunnistaa kerrallaan 1-5.

Fyysiset painikkeet iPadissa ovat kotipainike, äänenvoimakkuudensäädin, äänenvaimennuspainike ja unitila/virtapainike. (Apple, 2012h)

3.3 iPad-ohjelmointi

iPadin käyttöjärjestelmänä toimii iOS. iOS oli alun perin lyhenne iPhoneen käyttöjärjestelmästä (iPhone Operating System). Kun markkinoille tuli muitakin tätä käyttöjärjestelmää käyttäviä laitteita (iPad ja iPod Touch), käyttöjärjestelmän nimi lyhennettiin pelkäksi iOS:ksi. (Macworld 2010)

iOS:lle ohjelmointi onnistuu kahdella tavalla: natiiviohjelmoinnilla ja web-sovellusohjelmoinnilla.

iOS-natiiviohjelmointiin tarvitaan iOS SDK (iOS Software Development Kit), joka on Applen luoma kehitystyökaluympäristö. iOS SDK:n mukana tulee Xcode, jolla voi kirjoittaa ohjelmia iOS:lle ja OS X:lle. Ohjelmointikielenä iOS-ohjelmoinnissa on Objective-C. Xcoden mukana on sisäänrakennettuna Interface Builder, jonka avulla voi suunnitella sovelluksen käyttöliittymän. Kehitystyökalujen mukana tulee myös iOS-simulaattori ja useita muita työkaluja. (Apple 2012j)

Natiiviohjelmointi perustuu MVC-arkkitehtuuriin (Model-View-Controller), jossa toiminnot on eriytetty malliin, näkymiin ja ohjaimiin (Apple 2010e). MVC-arkkitehtuuri määrittelee säännöt ohjelman osien rooleihin ja kommunikointiin. Tämä arkkitehtuurimalli mahdollistaa ohjelmakoodin paremman uudelleenkäytettävyyden ja sovelluksen myöhemmän laajentamisen (Apple 2010e; Apple 2011e).

Jotkut kehitystyökalut, kuten tässä työssä käytettävä Unity, tarjoavat rajapinnan sovellusten tekoon. Varsinainen julkaisu tapahtuu kuitenkin edelleen XCoden kautta kahdessa eri vaiheessa. Ensin luodaan XCode-projekti, jonka jälkeen projekti koostetaan ja asennetaan iPadille. (Unity 2011c)

Web-sovellusohjelmoinnissa käytetään HTML:ää, tyylitiedostoja (CSS) ja JavaScriptiä vuorovaikutteisissa sovelluksissa, jotka sijaitsevat web-palvelimilla ja joita käytetään Safari-selaimessa.

4 UNITY

4.1 Miksi Unity?

Unity on kehitetty erityisesti pelikehitykseen ja se sisältää työkalut, jonka avulla voi luoda nopeasti pelattavan maailman. Unityn käyttöliittymä ja toimintaperiaate ottavat huomioon pelikehityksessä usein tarvittavat ominaisuudet, kuten hahmojen ja esineiden helpon uudelleenkäytettävyyden, graafisen pelimaailman muokkaimen ja nopean testausympäristön. Unity sisältää myös valmiiksi toteutettuja muokattavia komponentteja, joita ovat esimerkiksi fysiikkaominaisuudet ja törmäysalueet.

Unity tukee useimpia pelikehityksessä tarvittavia tiedostoformaatteja (mukaan lukien 15 eri 3D-tiedostoformaattia) ja huolehtii niiden optimoisesta projektiin, jolloin pelikehittäjän ei tarvitse kuluttaa aikaa tiedostojen muuntamiseen oikeaan muotoon. Unity tarjoaa rajapinnan useille eri alustoille julkaisemiseen.

4.2 Yleistä Unitystä

Unity on Unity Technologies –yhtiön luoma pelikehitystyökalu, jolla voi luoda 3D-pelejä. Unity tarjoaa ympäristön, jossa voi nopeasti luoda pelattavaa sisältöä.

Unityllä pystyy julkaisemaan pelejä useille eri alustoille. Huomioitavaa on, että julkaisu eri alustoille onnistuu käyttämällä suurimmaksi osaksi samaa ohjelmakoodia. Yleensä muutokset ovat lähinnä optimointia. Hyöty on huomattava niin ajallisesti kuin rahallistakin kun samaa peliä ei tarvitse tehdä erikseen jokaiselle eri kohdelaitteelle.

Unityn ilmaisiin julkaisuvaihtoehtoihin kuuluvat selaimessa ajettava Web Player, ajettavat ohjelmat PC:llä ja Macillä sekä Linux-versio Unityn 4-versiosta lähtien. Flash-tuki on saatavilla ilmaisessa kehitysversiossa, mutta kaikkia ominaisuuksia ei tueta. (Unity 2012a) Maksullisen lisenssin vaativat iOS, Android, Nintendo Wii, Playstation 3 ja Xbox 360. Unity Pro ja Unity iOS Pro –lisensseistä on saatavilla 30 päivän trial-versiot. (Unity 2012c)

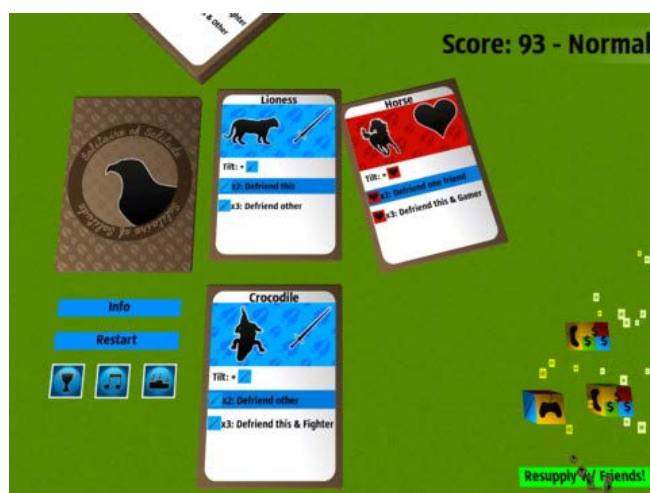
Unity on saatavissa Microsoft Windows ja Mac OS X -käyttöjärjestelmille. Varsinaisen kohdelaitteella ajettavan version koostamiseen iOS-laitteille tarvitsee Mac-tietokoneen,

johon on asennettu Mac OS X –käyttöjärjestelmä ja Xcode-kehitysympäristö sekä Applen provisiolisenssin.

Unityn käyttökelpoisuutta pelikehityksessä parantaa kiistatta sen eri julkaisuvaihtoehdot. Tietokoneiden ja pelikonsolien lisäksi maailmassa on käytössä satoja miljoonia eri mobiililaitteita ja uusia malleja tulee jatkuvasti lisää. On selvää, että mobiililaitteet ovat yksi nopeimmin kasvavista alueista pelikehittäjille. Ottaen huomioon mobiililaitteiden eri käyttöjärjestelmät ja 3D-kiihdyttimet, olisi todella vaikeaa tuottaa sisältöä niin suurille massoille tuottavasti ilman jotakin mekanismia, joka ottaisi huomioon laitteiden erot ylemmällä tasolla. Näin pelikehittäjä voi keskittyä paremmin itse pelikokemuksen suunnitteluun. (Pierce 2012, 8)

Pelikehitystiimissä myös muut kuin ohjelmoijat pystyvät muokkaamaan peliä. Unityn avulla on mahdollista muokata pelin muuttujia ilman, että tarvitsee muuttaa ohjelmakoodia. Tällaisia muuttujia voi olla esimerkiksi hahmon kävelynopeus, ohjuksen tulo-voima, kappaleen massa tai vaadittava pistemäärä jonkin toiminnon suorittamiseen. Muuttujia voi testata myös pelin ollessa käynnissä ja siten nähdä suoraan muutosten vaikutukset peliin. Näin myös pelin testaus nopeutuu ja välistä poistuu yksi välikäsi.

Unityllä on toteutettu runsaasti eri pelejä iPad-laitteelle. Esimerkkinä tässä on Ifelse Median Solitaire of Solitude, joka julkaistiin App Storessa 31.1.2012 (kuva 4).

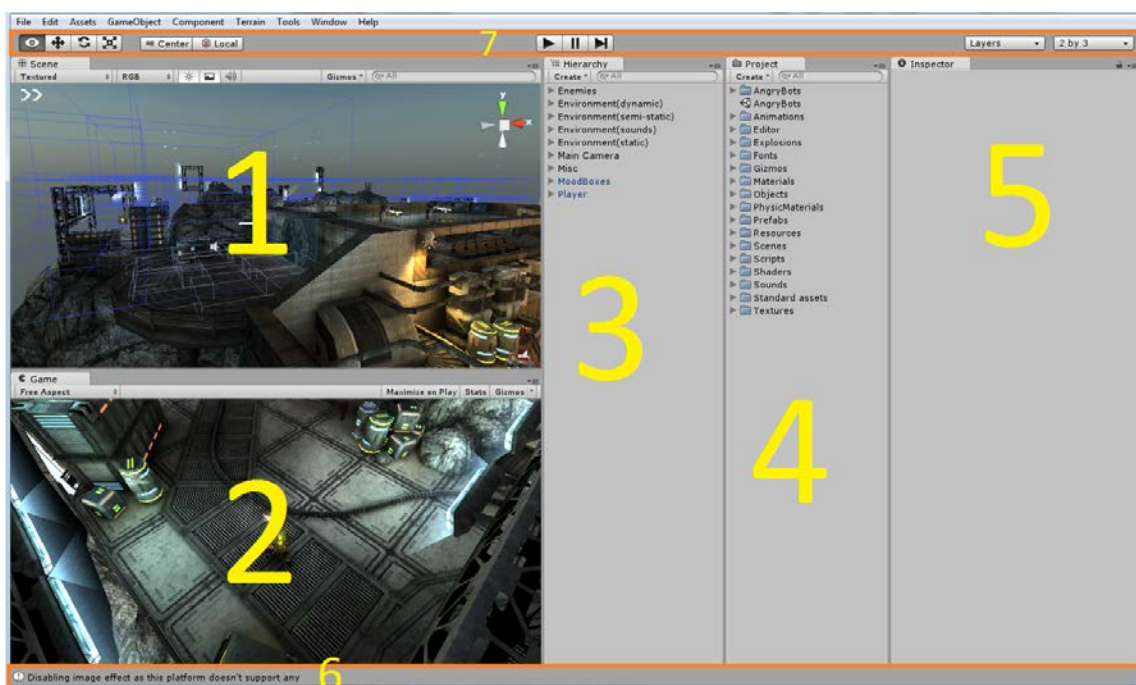


KUVA 4. Ifelse Media Oy:n julkaisema peli Solitaire of Solitude (kuvakaappaus, App Store)

Vaikka Unity on kehitetty ensisijaisesti 3D-pelien tekemiseen, on sille toteutettu sekä ilmaisia että maksullisia lisäosia, joiden avulla voi laatia pelejä, jotka näyttävät 2D:ltä. Ilman lisäosiakin on mahdollista laatia 2D:ltä näyttävä peli, jos syvyysakselia ei käytetä pelin toiminnoissa ja kameran perspektiivinäkymä laitetaan pois päältä (Unity 2011a; Wiebe 2011, 28).

4.3 Käyttöliittymä

Tässä luvussa kerrotaan Unityn käyttöliittymästä. Kuvaan 5 on numeroitu Unityn käyttöliittymän osat. Tämä käyttöliittymän jaottelumalli on 2 by 3 –malli. Osien paikkaa ja jaottelutapaa voi halutessaan muuttaa. Kuvan jälkeen esitellään käyttöliittymän osien tehtävät.



KUVA 5. Unityn käyttöliittymä (Kuvakaappaus)

Scene-näkökulmasta (1) pystyy hallitsemaan pelimaailman objekteja sekä tutkimaan pelimaailmaa eri kuvakulmista. Scene-näkökulman avulla voi peliin sijoittaa paikoilleen esimerkiksi pelihahmoja, kameroita ja valoja.

Game-näkökulman (2) avulla käyttäjä voi käynnistää pelinsä suoraan Unityssä. Peli käynnistetään ja tarvittaessa pysäytetään käyttöliittymän yläosassa sijaitsevista painikkeista.

Hierarchy-näkökulma (3) näyttää pelin scenellä olevat objektit.

Project-näkyvä (4) näyttää projektiin talletetut tiedostot (esimerkiksi kuvat, 3D-mallit, tekstuurit).

Inspector-näkyvä (5) näyttää Hierarchy- tai Project -näkymistä valittujen objektien komponenttien määrittelyt, kuten sijainnin pelikentällä tai objektiin kiinnitetyt skriptit. Inspector-näkymän avulla on mahdollista testata pelin muuttujia pelin ollessa käynnissä Game-näkymässä suoraan pelin ajon aikana.

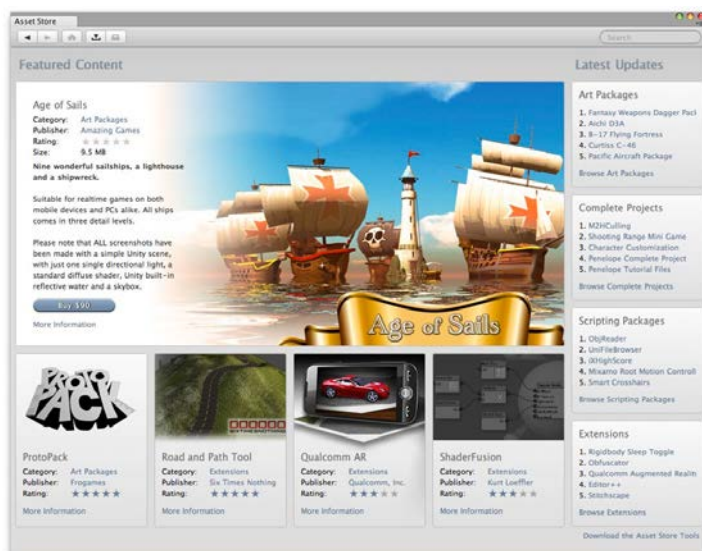
Console-näkyvä (6) näyttää pelistä tulevat viestit, jotka voivat olla virheitä, huomautuksia tai ohjelmakoodista lähetettyjä skriptikomentoja.

Toolbar-näkyvä (7) sisältää vasemmalla scenen objektien valinta, kääntö- ja kiertotyökalut. Keskellä ovat Game-näkymän käynnistys- ja pysäytyspainikkeet. Oikealla on Layers-alasvetovalikko, josta voi valita Scene-näkymässä näytettävät objektit. Oikeimmassa reunassa on Layout-alasvetovalikko, josta voi valita näkymien järjestyksen.

4.4 Assetit ja GameObjectit

Assetit (Assets) ovat pelin osatiedostoja. Asetteja ovat esimerkiksi 3D-mallit, tekstuurit ja ohjelmakoodit. Assetit tallennetaan Unityn projektikansiossa Assets-kansioon. Unityn mukana tulee myös valmiiksi luotuja asetteja.

Asetteja on saatavilla runsaasti Unityn Asset Storesta. Asset Store on Unitystä suoraan aukeava kauppapaikka (kuva 6), josta voi ostaa usein hyvinkin korkealaatuisia valmiita pelin osia. Asset Storesta löytyy myös ilmaisjakelussa olevia tiedostoja. Hyödyllisiä ovat esimerkiksi valmiit 3D-mallit, joissa on animaatiot valmiina. Ladatut tiedostot siirtyvät automaattisesti avoinna olevaan projektiin.



KUVA 6. Asset Store (Unity, Kuvakaappaus)

GameObjectit ovat aluksi tyhjiä objekteja, jotka eivät yksinään tee mitään. *GameObjectien* ensisijainen tehtävä on säilyttää objektin toiminnalliset osat sisältäviä komponentteja, joita hallitaan Inspector-näkymässä (Unity 2012e). Jokaisella *GameObjectilla* on ainakin yksi komponentti, Transform, joka määrittelee objektin paikan, rotaation ja skaalauksen pelimaailmassa (Menard, 2011, 42; Unity 2012e). Jokaisella scenellä olevalla objektilla on määrittely oma paikkansa x, y, ja z -akselilla.

4.5 Scenet

Pelimaailman objektit sijoitellaan pelissä scenelle. Scenet sisältävät kaiken, mitä pelissä halutaan tapahtuvan ja näkyvän mukaan lukien pelimaailman lattian, hahmot, kamerat, valot ja esimerkiksi käyttöliittymään ja pistelaskentaan liittyvät objektit. Scenellä olevia objekteja pystyy tarkastelemaan ja liikuttelemaan haluamansa mukaan scenen käyttöliittymäikkunasta. Kun peli ajetaan Unityssä, sceneen voidaan samanaikaisesti tehdä mitä tahansa muutoksia. Scenelle voidaan esimerkiksi lisätä tai poistaa objekteja tai objektien paikkaa, skaalaa tai rotaatiota voidaan muuttaa. Pelin ollessa Play-tilassa muutokset eivät kuitenkaan jää voimaan, jolloin tarpeettomat muutokset eivät jää vahingossa voimaan

4.6 Prefabit

Prefabit ovat uudelleenkäytettäviä asetteja. Prefabeja säilytetään Project-näkymässä. Prefab luodaan valitsemalla Unityn päävalikosta Assets → Create → Prefab. Tämän jälkeen luotuun prefabiin vedetään haluttu GameObject. Nyt luotua prefabia (instance) voi monistaa pelin scenelle vetämällä luotu prefab pelin scenelle. (Unity 2012d)

Kaikki prefabista monistetut GameObjectit ovat linkitetty alkuperäiseen prefabiin. Tällöin, jos alkuperäistä prefabia muokataan, kaikki GameObjectit perivät kyseiset muutokset. Kuitenkin jokainen prefab on itsenäinen olio pelimaailmassa. Prefabit ovat hyödyllisiä esimerkiksi vihollisten, kerättävien esineiden tai muiden usein toistuvien samanlaisten pelihahmojen tai -esineiden luonnissa.

4.7 Skriptaus

Unityssä on rajapinta C#, JavaScript ja Boo -ohjelmointikielille. Skriptin voi lisätä GameObjectiin raahaamalla skripti GameObjectiin Project-näkymässä. Yhdelle GameObjectille voi liittää useita eri skriptejä. Jos GameObjectia ei ole scenellä, ei skriptin metodeja myöskään kutsuta. Ohjelmointikielien sekoittelu samassa projektissa on mahdollista, mutta skriptien suoritusjärjestyksessä saattaa esiintyä ongelmia, joten sitä ei suositella. Unity ei salli Play-tilaan pääsyä, jos ohjelmakoodissa on virheitä.

(Pierce 2012, 100; Unity 2012f)

Skriptaus tapahtuu liittämällä skripti haluttuun GameObjectiin. Skriptin eri metodeja kutsutaan tiettyjen pelin tapahtumien yhteydessä. Taulukossa 2 esitellään eniten käytetyimpiä Behaviour-tyyppisiä metodeja ja niiden käyttötarkoituksia (Pierce 2012, 101).

TAULUKKO 2. Tärkeimmät skriptausmetodit

Metodi	Käyttötarkoitus
Awake	Kutsutaan kun skriptiobjektia ladataan muiden GameObjectien alustuksen jälkeen.
Start	Kutsutaan vain kerran - kun objektia, johon tämä skripti on liitetty, käytetään ensimmäisen kerran scenellä.
Update	Kutsutaan ennen jokaisen framen renderöintiä.
FixedUpdate	Kutsutaan jokaisella fysiikka-aika-askeleella. Käytetään fysiikkaperusteisiin pelin toimintoihin, jotka eivät ole riippuvaisia kehysnopeudesta (frame rate).
LateUpdate	Kutsutaan Update:n kutsun jälkeen.
”Muut metodit”	Metodien ulkopuolella oleva ohjelmakoodi suoritetaan kun objekti on ladattu. Voidaan käyttää skriptin tilan alustukseen.

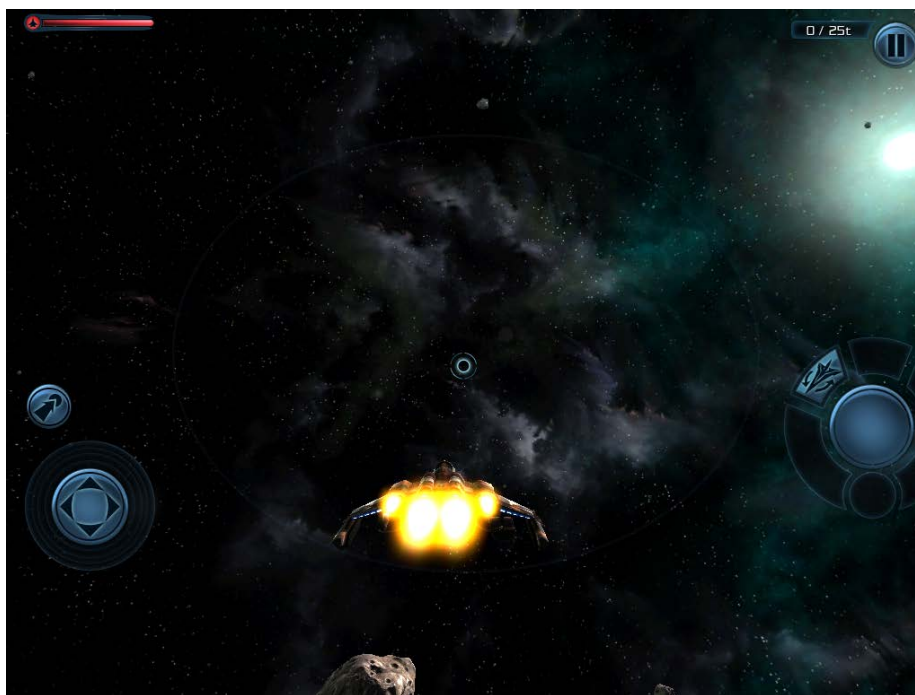
Unityn valmiista aseteista löytyy useita eri valmiita ohjelmakoodeja. Esimerkiksi kolmannen persoonan hahmo-ohjaus on hyvin käyttökelpoinen Unityn opettelussa, kun ei tarvitse aloittaa ohjelmointia ruohonjuuritasolta. Valmiista aseteista löytyvät helpot työkalut, joiden avulla voi luoda melko nopeastikin hahmon ohjausmekaniikkoineen ja pelimaailman.

5 MENETELMÄT

Jotta saataisiin selville, kuinka hyvin Unity soveltuu 3D-pelien tekemiseen iPad-laitteelle, tulisi suunnitella ja toteuttaa peli, joka käyttäisi yleisimpiä 3D-pelien tekniikoita. Päätin ensin tutkia, millaisia 3D-pelejä on toteutettu iPadille, minkälaisia ohjausmekaniikoita ne käyttävät ja mitä muita yhteisiä ominaisuuksia ne sisältävät.

Pelien ohjaustapaa tutkittiin viiden eri 3D-pelin avulla. Tutkittaviksi peleiksi valittiin tunnettuja ja menestyneitä pelejä, jolloin voitaisiin varmistua siitä, että pelien toteutustavat olisivat todennäköisesti toimivia. Tutkittaviksi peleiksi valittiin App Storesta Minecraft – Pocket Edition, Galaxy on Fire 2™ HD, Temple Run, Fruit Ninja HD Lite ja Zen Bound 2.

Useissa iPadin 3D-peleissä ohjausmekaniikka on toteutettu käyttöliittymässä olevien nuolien avulla. Joissakin peleissä, kuten Minecraft – Pocket Edition –pelissä nuolet ovat paikoillaan kun taas esimerkiksi Galaxy on Fire 2™ HD –avaruuslentelyssä ohjaus tapahtuu käyttöliittymässä joystickin tapaan, jolloin liu'uttamalla sormea ohjausalueella liikkuu myös alueella oleva ”joystick”. Useimmiten ohjauksen käyttöliittymäelementit oli sijoitettu pelin käyttöliittymän vasempaan alanurkkaan, ja mahdolliset muut toiminnot oli sijoitettu oikeaan reunaan tai oikeaan alanurkkaan (kuva 7).



KUVA 7. Galaxy on Fire 2™ HD (Kuvakaappaus)

Minecraft – Pocket Edition tuntui luontevammalta sijoittaa pöydälle kuin pitää käsissä. Pelihahmoa ohjataan käyttöliittymän vasemman alalaidan nuolinäppäimillä, joiden keskellä on hyppynäppäin (kuva 8). Oletuksena pelihahmo ei ole lainkaan näkyvässä. Rakentaminen onnistuu koskettamalla ruudulla haluamansa kohtaa.



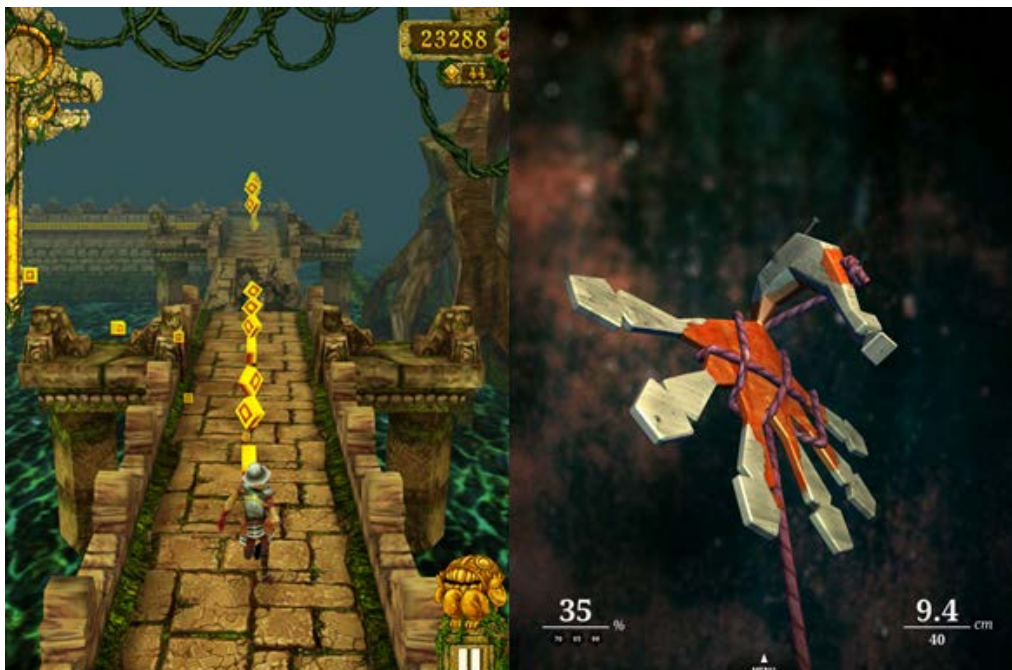
KUVA 8. Minecraft – Pocket Edition (Kuvakaappaus)

Fruit Ninja HD Lite –pelissä (kuva 9) halkaistaan lenteleviä hedelmiä vetämällä sormi ruudun poikki. Hedelmien halkaisu opetetaan jo käyttöliittymässä valitsemalla haluttu kenttä vetämällä sormi kyseisen kenttävalinnan yli. Tässä pelissä iPad oli luontevinta sijoittaa pöydälle, sillä peli on nopeampoinen ja vaatii molemmat kädet käyttöön.



KUVA 9. Fruit Ninja (Kuvakaappaus, Fruitninja.com)

Testiin valittiin myös kaksi pystysuuntaista peliä, Temple Run ja Zen Bound 2 (kuva 10). Joissakin peleissä kuten Temple Runissa käytetään kallistusta ohjaamiseen. Temple Run –pelissä pelihahmo juoksee kapeaa siltarakennelmaa pitkin ja hahmoa ohjataan kallistelemalla laitetta ja vetämällä sormeja ruudulla pelihahmon hyppyjä, käännöksiä ja aliliukuja varten. Toisessa pystysuuntaisessa pelissä, Zen Bound 2:ssa, yritetään kieputtaa köyttä erilaisten leijuvien esineiden ympärille käyttäen köyttä mahdollisimman vähän mutta samalla peittäen koko esineen pinta-alan. Ohjaus tapahtuu koskettamalla esinettä ja pyörittämällä sitä. Peliä ohjataan myös kääntelemällä laitetta, jolloin köyden tulokulma muuttuu.



KUVA 10. Temple Run ja Zen Bound 2 (Kuvakaappaus)

Kaikissa peleissä oli aloitusruutu, joista valitaan haluttu taso tai pelimoodi. Kaikissa peleissä oli käyttöliittymässä pause/menu-painike, jota napauttamalla aukeaa useimmissa peleissä äänen mute-valinta ja paluu päävalikkoon -valinta.

Peleistä paljastui ero, joka oli täysin riippuvainen pelin käyttöliittymästä. Pelin käyttöliittymän sijoittelu ja ohjaustapa vaikuttivat siihen, miten laitetta pidellään – joko kädessä tai pöydällä. Piercen (2012, 119) mukaan nämä erot myös vaikuttavat ohjauksen nopeuteen ja tarkkuuteen. Ensimmäisessä tapauksessa käyttäjä pitelee laitetta käsissään. Tällöin ensisijainen syöttötapa tapahtuu peukaloiden avulla melko hitaasti ja heikolla tarkkuudella. Toisessa tapauksessa käyttäjä asettaa laitteen pöydälle tai syliin, jolloin käytetään yleensä etusormea paremmalla tarkkuudella ja nopeimmin liikkein. Näin myös ohjausmekaniikka ja pelin tempo ovat kytköksissä toisiinsa.

Ohjausmekaniikan valinnan tueksi testattiin Unityn valmiita mobiiliassetteja, jotka sisältävät valmiina ohjausmekaniikat. Valmiista ohjausmekaniikoista toimivimmilta ja luontevimmilta vaikutti sidescroll-mekaniikka, joka sisälsi myös joidenkin testattujen pelien tapaan näkyvät ohjausalueet ruudulla. Peliin päätettiin toteuttaa myös aloitusruutu ja pause-ruutu äänen hiljennystoiminnolla.

5.1 Työvälineet

Työssä käytettävä Unity-versio on 3.5.2. Päävalikon suunnittelu tapahtuu Adobe Photoshop CS4 –ohjelmalla ja pelin muun käyttöliittymän muokkaus Gimp –ohjelmalla. 3D-mallinnukseen käytetään Cinema 4D:n R13-versiota. Kohdelaitteena toimii omistamani iPad 2, jonka käyttöjärjestelmä on iOS 4.3.5. Ohjelmoinnissa käytetään Unityn mukana tulevaa Mono Develop –editoria. Ohjelmointikieleksi valittiin C#, joka tarjoaisi paremman tuen olio-ohjelmointiin.

5.2 Vaatimukset

Pelihahmojen graafiset vaatimukset

Iästään riippuen, iOS-laitteet pystyvät renderöimään 10 000 – 30 000 verteksiä kerralla. Tätä suuremmat polygonimäärät aiheuttavat FPS:n ja pelin tökkimistä. Mobiililaitteille suunniteltaessa 3D-mallin olisi hyvä sisältää 300-1500 polygonia. Hyvä suorituskyky mobiililaitteilla saadaan kun käytetään alle 30 luuta per malli. Sivuosassa oleville hahmoille riittää 10-15 luuta per malli. (Unity 2011b; Wiebe 2011, 69-70)

Materiaaleja ei tulisi olla enempää kuin kolme yhtä 3D-mallia kohden. Renderöinnissä suoritusastoa voidaan parantaa käyttämällä yhteistä tekstuuria useammalle pelihahmolle, joille määritellään UV-kartoituksen avulla omat värinsä käytetystä tekstuurista. (Unity 2011b)

6 PELIN TOTEUTUS

6.1 Pelin genre ja idea

Valmiiden mobiiliassetien perusteella pelin lajityypiksi, ”genreksi”, valittiin tasohyppely. Tasohyppely on luonteeltaan peli, jossa pelihahmo liikkuu useimmiten sivuttaissuunnassa, hyppien eri korkeuksilla oleville tasoille, esteitä ja vihollisia välttämällä. Tasohyppelylle on ominaista epärealistinen fysiikka: pelihahmo voi useissa peleissä muuttaa suuntaansa hypyn aikana ilmassa (Adams 2010, 396). Wieben mukaan (2011, 51-53) tasohyppelyn peruselementtejä ovat erityyppiset kerättävät esineet, arkut, energianpalaajat ja mekaniikat. Mekaniikoista toteutettavaksi valittiin kerättävät esineet ja liikkuva taso, jonka avulla pelaaja voi ylittää rotkon.

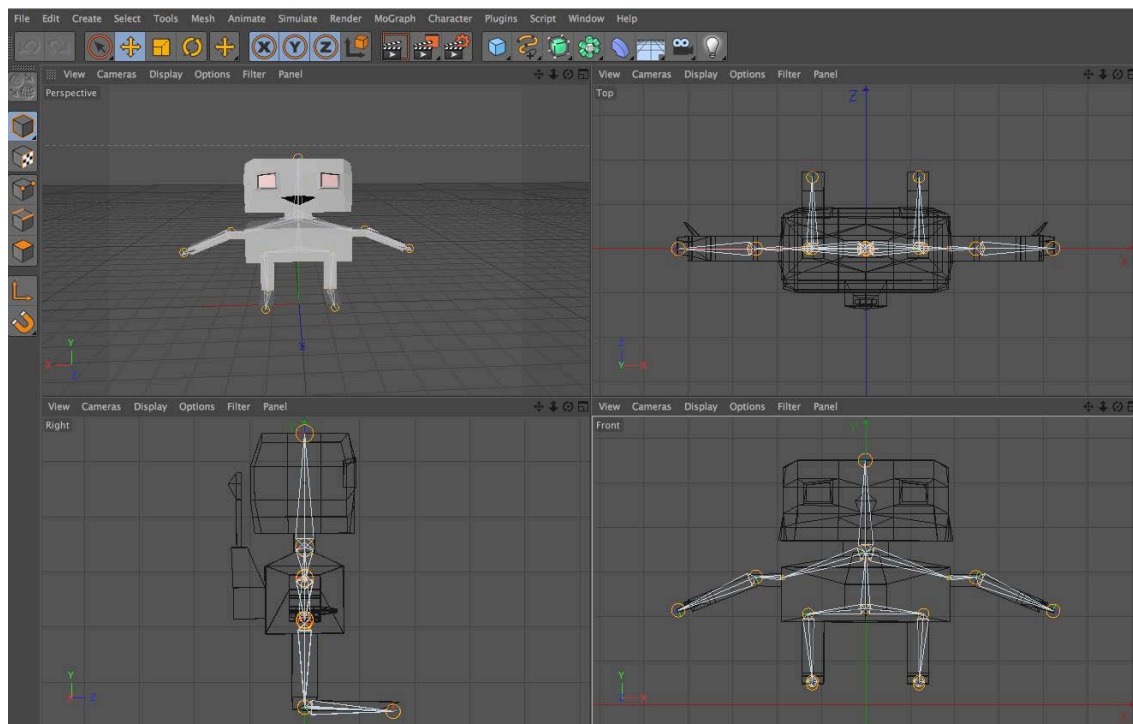
Pelin ideana on edetä avaruudessa robotti-pelihahmolla hyppimällä painovoimatasolta toiselle ja kerätä jalokiviä. Pelihahmoa ohjataan ruudun alalaidan ohjausalueiden avulla. Hahmo kuolee jos se putoaa painovoimatasolta pois. Pelin voittaa, jos pääsee kentän loppuun.

6.2 3D-mallit

6.2.1 Mallien suunnittelu

Tähtäimessä oli tarpeeksi alhainen polygonimäärä eli noin 300-1500 polygonia yhtä mallia kohden. Polygonimäärän seuraamista varten avasin Cinema 4D:ssä Viewport-valikon. Valitsemalla HUD-välilehdeltä ”Total Polygons” -laatikon valituksi pystyin seuraamaan valitun mallin polygonimäärää samaan aikaan, kun muokkasin mallia.

Jotta malli olisi varmasti symmetrinen, asetettiin alkuperäinen polygoniverkko Symmetry-objektin sisään. Tällöin muuttamalla polygoniverkon pisteitä x-akselilla polygoniverkko peilautuu automaattisesti symmetriseksi malliksi. Lopuksi Symmetry-objekti ja sen lapsiobjektit yhdistettiin yhdeksi objektiksi (kuva 11).



KUVA 11. Pelihahmon mallinnus Cinema 4D:ssä (Kuvakaappaus)

6.2.2 Valmiiden mallien käyttö

Pelissä käytettiin myös Unityn Asset Storesta ladattuja ilmaisia 3D-malleja pelin maailmaan luomaan tunnelmaa. Kaikkien mallien kohdalla ei oltu mainittu polygonimäärää, joten joidenkin mallien kohdalla jouduttiin tyytymään silmämääräiseen arvioon ja tarkistamaan polygonien määrä jälkikäteen Cinema 4D:ssä. Kaikki mallit eivät soveltuneet käytettäväksi pelissä niiden sisältämän korkean polygonimäärän vuoksi. Yleisesti ottaen valmiiden mallien käyttö säästi kuitenkin runsaasti aikaa ja vähensi työmäärää huomattavasti sekä paransi pelin ulkonäköä.

6.2.3 Tekstuurit ja materiaalit

Pelihahmon päävärialueet rajattiin polygonivalitsemistyökalulla. Väreille asetettiin omat valinta-alueet, joille taas määrättiin omat materiaalit. Materiaaleissa pitäydettiin Cinema 4D:n perusasetuksissa muutoin kuin värien osalta. Pelihahmolla ei ollut tarvetta varsinaisiin tekstuureihin.

Pelimaailman tasojen tekstuureina käytettiin Unity Asset Storesta ladattavia ilmaistekstuureja. Pelin painovoimatasojen teksturointiin käytettiin ilmaistekstuureja.

6.2.4 Nivelet

Nivelten teko aloitettiin Cinema 4D:ssä nivelytyökalun avulla (joints). Nivelet mallinnettiin ensin pelkästään mallin oikealta puolelta, jonka jälkeen luustosta luotiin peilikuva mirror-työkalun avulla.

Tämän jälkeen malli yhdistettiin niveliin Cinema 4D:n bind-työkalun avulla.

6.2.5 Animaatiot

Animaatiot nauhoitettiin Cinema 4D:ssä ns. keyframe-tekniikalla. Sekunnissa määrättiin olevan 30 framea. Animaatioita tuli yhteensä kolme: idle-, kävely- ja hyppyanimaatio.

Keyframet ovat animaatioframeja, jotka ovat ns. määrääviä frameja. Luiden asentoja muutettiin valituissa frameissa ja nauhoitettiin näiltä keyframet. Cinema 4D päättelee väliin tulevan informaation, jolloin saadaan aikaan sulava animaatio. Aikajanalla katsottuna tarvitsee päättää vain aloittava ja lopettava frame, jolloin Cinema 4D osaa päätellä niiden väliin tulevan informaation.

6.2.6 Mallien vienti Unityyn

Ennen Unityyn viemistä mallit käsiteltiin Cinema 4D:n Triangulate-työkalulla, joka muuttaa kaikki hahmon polygonit kolmioiksi. Mahdolliset muunmuotoiset polygonit voivat aiheuttaa malliin reikiä tai valaistusongelmia Unityssä. (Wiebe, 2011, 239)

Unityyn Cinema 4D-tiedostojen vienti pitäisi toimia suoraan raahaamalla Unityyn, mutta tässä koettiin pieniä ongelmia mahdollisesti versioyhteensopivuusongelman takia. Mallit vietiin Cinema 4D:stä FBX-muodossa, jotka vietiin Unityyn.

6.3 Pelihahmon ohjaus

Unity Pron mukana tulee valmiita eri ohjausmenetelmiä sisältäviä asetteja mobiililaitteille. Testasin näitä kaikkia iPadilla ja valitsin tämän perusteella sivusta ohjattavan pelimekaniikan, joka vaikutti tässä testausvaiheessa luontevimmalta vaihtoehdolta. Valitsemani sidescroll-ohjausmekaniikka sisälsi pelin lattian, ohjaus- ja hyppymekaniikan sekä valmiiksi ohjattavan pelihahmon kapseliprimitiivinä.

Pelihahmoa liikutellaan sivuttaissuunnassa liikuttamalla sormea vasemman alareunan ohjausaluetta. Oikean alareunan napautuksesta pelihahmo hyppää.

Valmiissa sidescroll-asetissa nämä alueet näkyvät vain, jos pelaaja painaa kyseiseltä alueelta. Ohjaus muutettiin heti niin, että alueet näkyvät jatkuvasti. Kun annoin pelin testattavaksi tuttavilleni, ongelmaksi tuli että he eivät siltikään tienneet, mistä pelihahmoa olisi pitänyt ohjata.

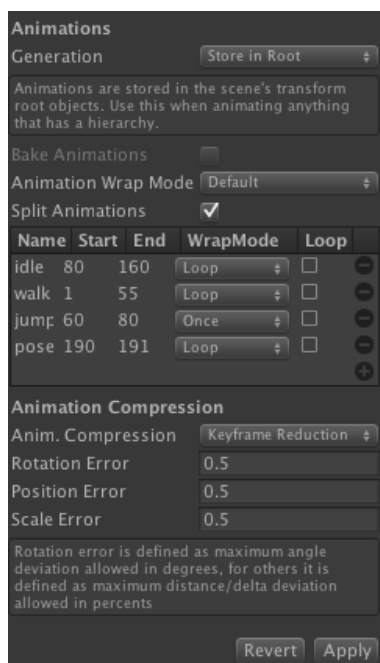
Saadun palautteen perusteella ohjausalueita muutettiin niin, alueiden keskellä on nuolet ja niiden taustalla näkyy vasemmalla sivunuolet ja oikealla nuoli ylös, joka tarkoittaa hyppyä (kuva 12).



KUVA 12. Ohjausalueet ennen muokkausta (ylempi) ja muokkauksen jälkeen (alempi).

6.4 Animaatioiden hallinta

Unityyn viennin jälkeen hahmon animaatiot jaettiin osiin Import-asetusten avulla. Jokaiselle animaatiolle asetettiin nimi ja alku- sekä loppuframe. Hyppyanimaatio asetettiin toistumaan vain kerran. Muut animaatiot asetettiin toistumaan yhä uudestaan (loop) (kuva 13).



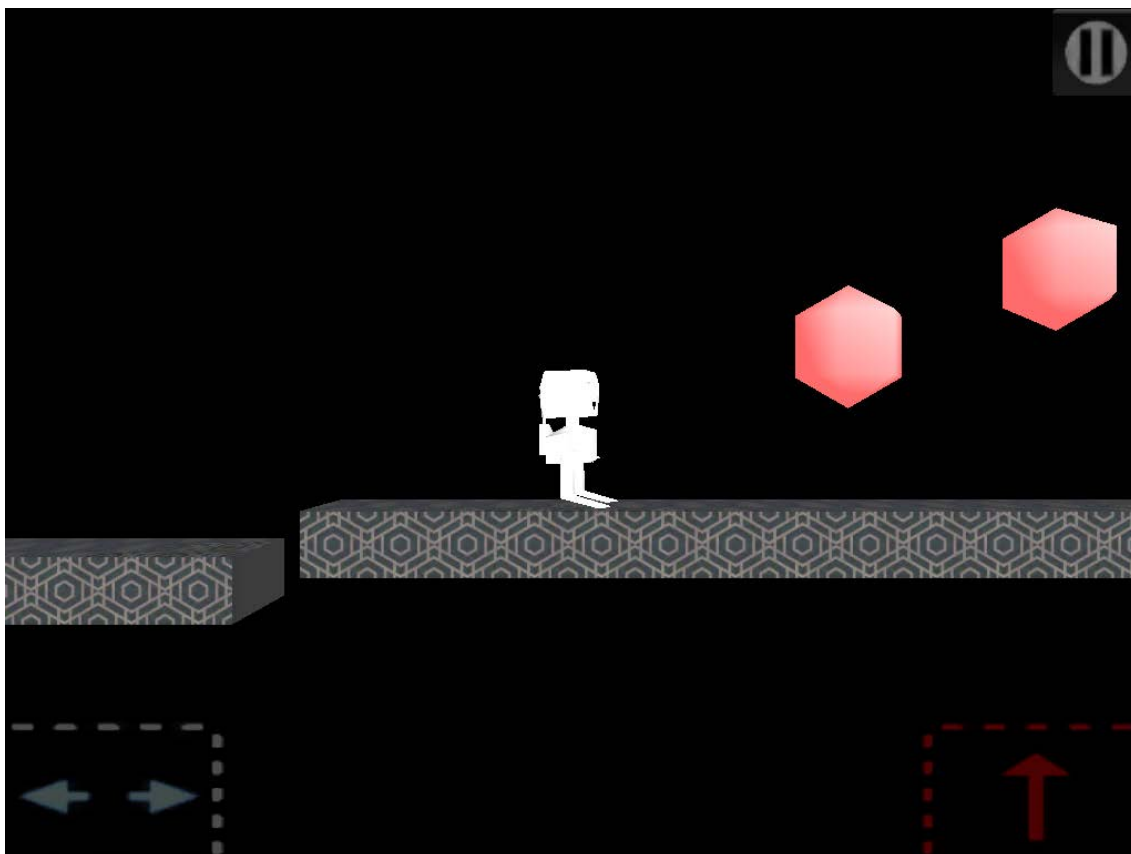
KUVA 13. Pelihahmon animaatioasetukset Unityssä

Animaatiot kiinnitettiin hahmon ohjaukseen, jolloin hyppääminen aktivoi hyppyanimaation. Oletusanimaatioksi asetettiin idle-animaatio, joka näytettäisiin kun hahmoa ei liikuteta.

6.5 Pelimaailma

Pelimaailma sijoittuu erilaisille painovoimatasoille. Jos pelaaja putoaa painovoimatasolta pois, pelihahmo kuolee ja taso alkaa alusta.

Ensimmäisen tason alussa on kuoleminen lähes mahdotonta, jotta pelaaja tottuisi pelimekanismiin. Pelimaailmaan asetettiin kerättäviä esineitä kuten jalokiviä (kuva 14). Pelin pääsee läpi kun pääsee ensimmäisen kentän loppuun.



KUVA 14. Pelimaailma (kuvakaappaus)

6.5.1 Colliderit

Colliderit ovat kehitteitä, joiden tehtävänä on viestiä pelimoottorille törmäyksestä (collision) Collidereita on Unityssä erimuotoisia, usein käytetään collidereina laatikkoa (Box Collider) ja kapselia (Capsule Collider). Mesh Collider käyttää objektin omaa polygoniverkkoa colliderina ja se soveltuu collideriksi monimutkaisille muotoisille objekteilte. Kehikon suuruutta ja sijaintia suhteessa objektiin voi muuttaa. (Unity Script Reference)

Jos colliderilla on sen komponenteissa Is Trigger –ominaisuus päällä, ei collideria käytetä fyysisenä esteenä vaan viestimenä toisen colliderin saapumisesta trigger-osualueelle. Tällöin osuvalla colliderilla tulee olla myös Rigidbody-fysiikkakomponentti. Käytettäviä viestejä ovat OnTriggerEnter, OnTriggerExit ja OnTriggerStay. (Unity 2009)

Pelihahmolla käytettiin Capsule Collideria, jonka korkeutta ja sijaintia muokattiin niin, että hahmo seisoi tukevasti tasojen päällä.

Kerättävät esineet asetettiin triggereiksi. Esineistä tehtiin prefabit ja niille liitettiin `CollectibleDestroyer.cs` -ohjelmakoodi, jonka tuhoamiskäske sijoitettiin `OnTriggerEnter`-metodin sisään, jolloin kerättävät esineet tuhoutuvat kun pelihahmon collider saapuu esineen osuma-alueelle:

```
void OnTriggerEnter(Collider other) {  
    AudioSource.PlayClipAtPoint(clip, new Vector3(3,5,1));  
    Destroy(gameObject);  
}
```

Painovoimatasojen alle asetettiin triggerinä tyhjä Box Collider, jonka tehtävänä oli toimia ns. kuolemisalueena. Ensin pelihahmolle asetettiin `GameManager.cs`-skriptiin boolean-tyyppinen `characterIsAlive`-muuttuja. Kuolemisalueelle asetettiin `CharacterKiller.cs`-skripti, jonka tehtävänä on vaihtaa `GameManager.cs`-skriptin `characterIsAlive`-muuttuja `OnTriggerEnter`-metodin avulla trueksi. Skriptin yhteys toisen skriptin muuttujiin avattiin alustamalla olio, johon vedettiin `GameManager.cs`-luokka Inspector-näkymässä.

Jos ohjelmakoodissa oli virheitä, ei Unity päästänyt pelitilaan lainkaan vaan ilmoitti virheestä Game-näkymässä. Console-näkymässä oli lueteltu ohjelmakoodin virheet ja rivinumerot, jonka avulla virheet oli helppo paikantaa.

6.6 Käyttöliittymä

Pelin aloitusruudun graafinen ilme toteutettiin Adobe Photoshop CS4 -ohjelmalla. Peli-hahmo liitettiin Cinema 4D:llä renderöidystä kuvasta ruutuun (kuva 15).

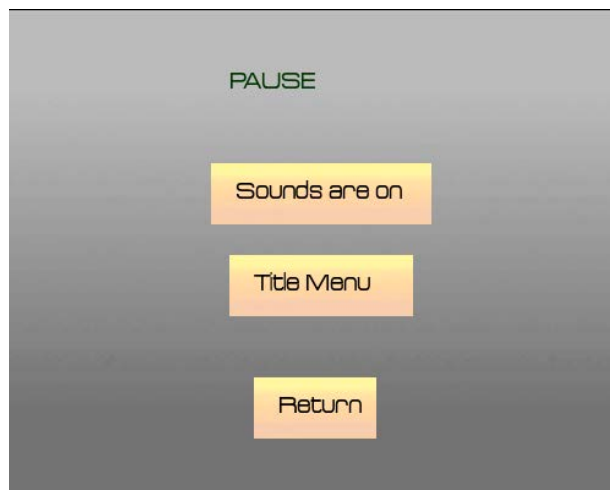


KUVA 15. Pelin aloitusruutu (Kuvakaappaus)

Aloitusruutuun liitettiin ohjelmakoodi, jonka avulla Play-painikkeen napauttaminen aloittaa ensimmäisen levelin:

```
void OnGUI () {
    if (GUI.Button(new Rect(Screen.width/2-100, Screen.height/2-50,
        250, 80), "", guiStyle)) {
        Application.LoadLevel(1);
    }
}
```

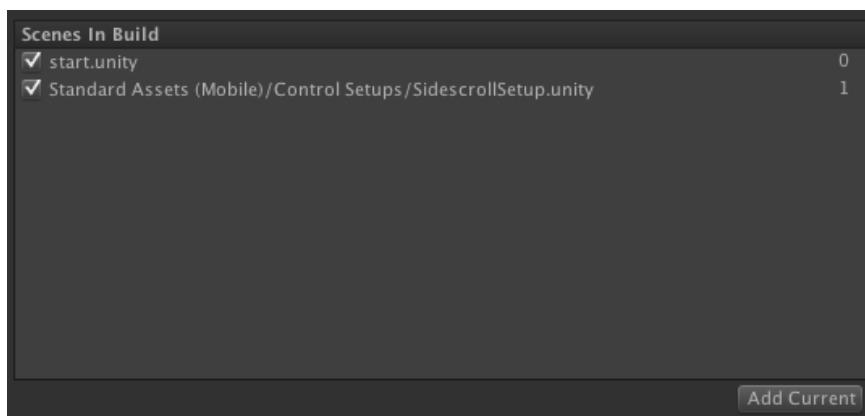
Pelin käyttöliittymään lisättiin pause-painike. Painikkeesta avautuu ruudun keskelle pause-valikko (kuva 16), josta voi säätää äänenvoimakkuutta, palata alkuruutuun ja palata peliin. Pause-valikon aktivoiminen poistaa myös ohjausalueet näkyvistä.



KUVA 16. Pause-valikko

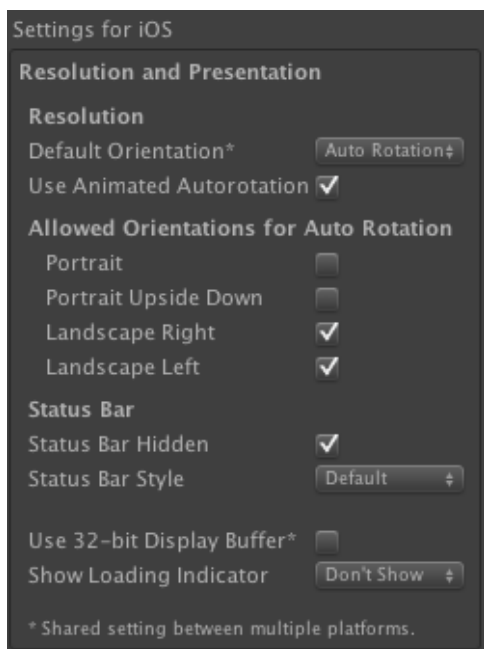
6.8 Julkaisu

Unityn julkaisuasetuksissa valittiin koostettavaan peliin tulevat scenet vetämällä scenet Project-näkymästä. Pelin aloittava start-scene asetettiin ensimmäiseksi, joten se sai indeksinumeron 0. Levelien lataaminen ohjelmakoodissa tapahtuisi helposti käyttämällä pelin indeksinumeroita (kuva 17).



KUVA 17. Julkaisuasetusten scenevalinta (Kuvakaappaus)

Player Settings -asetuksista valittiin iOS-asetuksista automaattinen orientaation käännös ja valittiin sallituiksi ruudun orientaatioiksi Landscape Right ja Landscape Left (kuva 18). Tilarivi piilotettiin.



KUVA 18. iOS-asetusten resoluutio- ja näyttöasetukset (Kuvakaappaus)

7 TULOKSET

Pelin toteuttamisessa ei ilmennyt juurikaan vaikeuksia. Unityn saumaton 3D-mallien käsittely ja animaatioiden hallinta helpottivat pelin toteutusta huomattavasti. Unityn Asset Storesta löytyi laadukkaita ilmaisia pelin osia.

Ohjelmointi oli niin ikään melko helppoa kattavan dokumentaation ja Unityn hyvän toimintalogiikan ansiosta. Unityn valmiit ohjausmekaniikat tarjosivat avun alkuun pääsyssä ja havainnollistivat osaltaan kuinka nopeasti Unityssä voi päästä alkuun pelin toteutuksessa.

Työn edetessä ilmeni, että tasohyppely ei ollut paras valinta pelin genreksi. Parempi vaihtoehto olisi ollut jäljitellä jotakin testattujen pelien ohjausmekaniikkaa. Peliä oli hieman kömpelö pelata ja tasohyppelypelien nopeatempoisuus oli ristiriidassa pelin kankean ohjausmekaniikan vuoksi. Kuitenkin peliobjektien hallinta, ohjelmointi, julkaisu ja muiden Unityn ydintoimintojen käyttö tuli työn aikana tutuksi.

Unityssä oli joitakin puutteita. Unitystä ei löytynyt valmista pääsyä eleihin, mikä olisi ollut todella hyödyllistä. Toteuttamani pelin ohjausmekaniikasta johtuen peliä ei voinut testata Unityssä juuri lainkaan, joten peli jouduttiin koostamaan aina iPadille, missä vierähti aina pieni tovi. Tällöin ilmeni myös joskus virheitä, jotka kaatoivat koko pelin. Pelin pystyi pysäyttämään XCodessa tai poistumaan iPadin Koti-painikkeen avulla kotiruutuun. Itse iPad ei koskaan kaatunut täysin. Testauksessa kului kuitenkin yllättävänkin suuri osa pelin toteutusajasta.

Koin Unityn olevan kehitysympäristö, jonka avulla pystyin erityisesti keskittymään itse pelikokemuksen suunnitteluun ja parantamiseen. Koska Unity on suunniteltu erityisesti pelikehitykseen, onnistui monien ominaisuuksien toteuttaminen valmiiden komponenttien avulla, joita tarvitaan lähes jokaisessa pelissä. Kuitenkin nämä komponentit ovat vain työkaluja eikä niiden avulla luoda vielä erinomaista peliä. Pelin onnistuminen vaatii myös perehtymystä peleihin ja pelikehitykseen syvemminkin. Tämä pätee varsinkin mobiilipelikehitykseen, jossa on erityisen tärkeää ymmärtää kohdelaitteiden käyttäjien tarpeita.

Ansaintanäkökulmasta houkuttelevaa on, että halutessani voisin julkaista valmiin pelin esimerkiksi Android-taulutietokoneille käyttäen samaa ohjelmakoodia. Myös hyvälaatuiset assetit voisi peliprojektin päätteeksi julkaista Asset Storessa, minkä avulla voisi potentiaalisesti luoda lisätuloja.

8 JOHTOPÄÄTÖKSET

Opinnäytetyöni tavoitteena oli tutkia, onko Unity soveltuva 3D-pelien tekemiseen iPad-laitteelle. Omasta mielestäni saavutin tavoitteen hyvin, vaikka parantamisen varaa tietenkin on.

Menetelmän valinnassa riitti pohdittavaa. Eräs tapa saada tavoitteeseen vastaus olisi ollut vertailla iPad-natiiviohjelmointia ja Unityä, mutta sellaisesta opinnäytetyöstä olisi todennäköisesti tullut liian laaja eikä minulla ollut lainkaan kokemusta natiiviohjelmoinnista. Päädyin tutkimaan aihetta yleisesti käytössä olleiden pelikäytäntöjen kautta.

Aloitin aiheen ideoimisen vuoden 2011 lopulla. Opinnäytetyön aihepiiri oli melko uusi, joten kirjallisuutta oli saatavilla vielä melko vähän. Saatavilla olleet kirjalähteet olivat korkeintaan noin kaksi vuotta vanhoja. Aihepiirin kirjoja ilmestyi jopa työn edetessä. Samoin vuoden 2012 aikana, jolloin tein opinnäytetyötäni, ilmestyi kaksi uutta iPad-mallia ja kaksi uutta iOS-päivitystä, mikä osaltaan kertoo mobiililaitemarkkinoiden nopeista liikkeistä.

Unityn www-sivut osoittautuivat hyväksi tietolähteeksi. Työn alussa niiden tieto oli kattavaa, mutta paikoin tiedot olivat vielä hieman hajanaisia ja puutteellisia. Vuoden 2012 aikana niiden tieto päivittyi ja täydentyi.

Sain yleisellä tasolla käsityksen Unityn soveltuvuudesta 3D-pelien tekoon iPadille. Jatkokehitysideana voisi aihetta tutkia vertailemalla natiiviohjelmointia Unity-ohjelmointiin.

LÄHTEET

Adams, E. 2010. Fundamentals of Game Design. 2. painos. Berkeley (Calif.) : New Riders.

Amazon. 2011. Lehdistöiedote. Julkaistu 8.11.2011. Luettu 11.11.2012.
<http://phx.corporate-ir.net>

Android Official Blog. 2012. Päivitetty 26.9.2012. Luettu 31.10.2012.
<http://officialandroid.blogspot.fi/2012/09/google-play-hits-25-billion-downloads.html>

Apple. 2010a. Apple Launches iPad. Julkaistu 27.1.2010. Luettu 23.8.2012.
<http://www.apple.com/pr/library/2010/01/27Apple-Launches-iPad.html>

Apple. 2010b. Apple Reports Third Quarter Results. Julkaistu 20.7.2010. Luettu 27.8.2012.
<http://www.apple.com/fi/pr/library/2010/07/20Apple-Reports-Third-Quarter-Results.html>.

Apple. 2010c. Apple Reports Fourth Quarter Results. Julkaistu 18.10.2010. Luettu 27.8.2012.
<http://www.apple.com/fi/pr/library/2010/10/18Apple-Reports-Fourth-Quarter-Results.html>

Apple. 2010d. Apple Sells One Million iPads. Julkaistu 3.5.2010. Luettu 27.8.2012.
<http://www.apple.com/pr/library/2010/05/03Apple-Sells-One-Million-iPads.html>

Apple. 2010e. iOS App Programming Guide. Luettu 27.8.2012.
<http://developer.apple.com/>

Apple. 2011a. Apple Reports First Quarter Results. Julkaistu 18.1.2011. Luettu 27.8.2012.
<http://www.apple.com/fi/pr/library/2011/01/18Apple-Reports-First-Quarter-Results.html>

Apple. 2011b. Apple Reports Fourth Quarter Results. Julkaistu 18.10.2011. Luettu 27.8.2012.
<http://www.apple.com/fi/pr/library/2011/10/18Apple-Reports-Fourth-Quarter-Results.html>

Apple. 2011c. Apple Reports Second Quarter Results. Julkaistu 20.4.2011. Luettu 27.8.2012.
<http://www.apple.com/fi/pr/library/2011/04/20Apple-Reports-Second-Quarter-Results.html>

Apple. 2011d. Apple Reports Third Quarter Results. Julkaistu 19.7.2011. Luettu 27.8.2012.
<http://www.apple.com/fi/pr/library/2011/07/19Apple-Reports-Third-Quarter-Results.html>

Apple. 2011e. iOS Technology Overview. Päivitetty 12.10.2011. Luettu 31.8.2012.
<http://developer.apple.com/>

Apple. 2012a. Apple Reports First Quarter Results. Julkaistu 24.1.2012. Luettu 27.8.2012.
<http://www.apple.com/pr/library/2012/01/24Apple-Reports-First-Quarter-Results.html>

Apple. 2012b. Apple Reports Fourth Quarter Results. Julkaistu 25.10.2012. Luettu 5.11.2012. <https://www.apple.com/pr/library/2012/10/25Apple-Reports-Fourth-Quarter-Results.html>

Apple. 2012c. Apple Reports Second Quarter Results. Julkaistu 24.4.2012. Luettu 27.8.2012.
<http://www.apple.com/pr/library/2012/04/24Apple-Reports-Second-Quarter-Results.html>

Apple. 2012d. Apple Reports Third Quarter Results. Julkaistu 24.7.2012. Luettu 27.8.2012.
<http://www.apple.com/pr/library/2012/07/24Apple-Reports-Third-Quarter-Results.html>

Apple. 2012e. Apple Sells Three Million iPads in Three Days. Julkaistu 5.11.2012. Luettu 5.11.2012.
<https://www.apple.com/pr/library/2012/11/05Apple-Sells-Three-Million-iPads-in-Three-Days.html>

Apple. 2012f. Built-in Apps. Luettu 27.8.2012.
<http://www.apple.com/ipad/built-in-apps/>

Apple. 2012g. iPad käyttöopas. Päivitetty 15.3.2012. Luettu 2.10.2012.
http://manuals.info.apple.com/fi_FI/ipad_kayttoopas.pdf

Apple. 2012h. iPad User Guide. Päivitetty 15.3.2012. Luettu 2.10.2012.
http://manuals.info.apple.com/en_US/ipad_user_guide.pdf

Apple. 2012i. iPad technical specifications. Luettu 29.10.2012.
<http://www.apple.com/ipad/specs/>

Apple. 2012j. What's New in Xcode 4. Luettu 27.8.2012.
<https://developer.apple.com/technologies/tools/whats-new.html>

Bloomberg Businessweek. 2012. Google Says 700,000 Applications Available for Android. Päivitetty 29.3.2012. Luettu 3.12.2012.
<http://www.businessweek.com/news/2012-10-29/google-says-700-000-applications-available-for-android-devices>

Business Insider. 2011. A Brief History of Mobile Gaming. Päivitetty 21.8.2012. Luettu 28.8.2012.
http://articles.businessinsider.com/2011-08-21/tech/30035291_1_mobile-gaming-n-gage-competitive-online-gaming

Cnet. 2011. A brief history of Android phones. Julkaistu 2.8.2011. Luettu 8.11.2012.
http://reviews.cnet.com/8301-19736_7-20016542-251/a-brief-history-of-android-phones/

Electronic Arts. 2011. EA Releases The Sims FreePlay App on iPad, iPhone and iPod Touch for Free. Julkaistu 16.12.2011. Luettu 5.11.2012.
<http://investor.ea.com/releasedetail.cfm?ReleaseID=633878>

Forbes. 2012. Mobile Game Revenue Math. Julkaistu 14.2.2012. Luettu 25.9.2012.
<http://www.forbes.com/sites/terokuittinen/2012/02/14/mobile-game-revenue-math/>

Gartner. 2012. Lehdistötiedote. Julkaistu 14.11.2012. Luettu 3.12.2012.
<http://www.gartner.com/it/page.jsp?id=2237315>

Google Official Blog. 2012. Introducing Google Play: All your entertainment, anywhere you go. Julkaistu 6.3.2012. Luettu 31.10.2012.
<http://googleblog.blogspot.fi/2012/03/introducing-google-play-all-your.html>

International Data Corporation. 2012. Julkaistu 13.3.2012. Luettu 6.12.2012.
<http://www.idc.com/getdoc.jsp?containerId=prUS23371312>

Kauppalehti. 2012. Raja meni rikki: Angry Birdsillä yli miljardi latausta. Päivitetty 9.5.2012. Luettu 23.8.2012.
<http://www.kauppalehti.fi/omayritys/raja+meni+rikki+angry+birdsilla+yli+miljardi+latausta/201205172219>

Kotek. 2012. Luettu 6.11.2012.
<http://www.kotek.fi/>

Macworld. 2010. iPhone OS gets new name, video calling. Päivitetty 7.6.2010. Luettu 27.8.2012.
http://www.macworld.com/article/1151812/iphone_os_4_wwdc.html

Menard, Michelle. 2011. Game Development with Unity. Boston. E-kirja.

Microsoft. 2010. Windows Phone 7: A Fresh Start for the Smartphone. Julkaistu 11.10.2010.
<http://www.microsoft.com/en-us/news/features/2010/oct10/10-11wp7main.aspx>

Microsoft. 2011. Nokia and Microsoft Announce Plans for a Broad Strategic Partnership to Build a New Global Mobile Ecosystem. Julkaistu 10.2.2011. Luettu 6.11.2012.
<http://www.microsoft.com/en-us/news/press/2011/feb11/02-11partnership.aspx>

Microsoft. 2012. Previewing "Windows 8". Julkaistu 1.6.2011. Luettu 2.12.2012.
<http://www.microsoft.com/en-us/news/features/2011/jun11/06-01corpnews.aspx>

Nokia. 2012a. Lehdistötiedote. Julkaistu 22.6.2011. Luettu 6.11.2012.
<http://press.nokia.com/2011/06/22/nokia-and-accenture-finalize-symbian-software-development-and-support-services-outsourcing-agreement/>

- Nokia. 2012b. Lehdistötiedote. Julkaistu 5.9.2012. Luettu 8.11.2012.
<http://press.nokia.fi/2012/09/05/uudet-nokia-lumia-alypuhelimet-tuovat-markkinoille-viimeisimmat-pureview-kuvaustoiminnot-uusia-paikkatietopalveluita-ja-langattoman-latausmahdollisuuden-windows-phone-8-alustalla/>
- Nokia. 2012c. Symbian OS. Päivitetty 26.7.2012. Luettu 2.12.2012.
http://www.developer.nokia.com/Community/Wiki/Symbian_OS
- Nokia. 2012d. The mobile revolution. Luettu 6.11.2012.
<http://www.nokia.com/global/about-nokia/about-us/story/the-mobile-revolution/>
- PC World. 2012. Angry Birds Reaches 1 Billion Downloads. Päivitetty 9.5.2012. Luettu 23.8.2012.
http://www.pcworld.com/article/255337/angry_birds_reaches_1_billion_downloads.html
- Pierce, G. 2012. Unity iOS Game Development : develop iOS game from concept to cash flow using Unity. Birmingham: Packt Publishing.
- Tekniikka & Talous. Jo on aikakin: operaattorit ajavat wap-palveluja alas. Päivitetty 31.10.2012. Luettu 8.11.2012.
<http://www.tekniikkatalous.fi/ict/jo+on+aikakin+operaattorit+ajavat+wappalveluja+alas/a852190>
- The Entertainment Software Association. 2012. The Evolution of Mobile Games. Luettu 28.8.2012.
<http://www.theesa.com/games-improving-what-matters/mobile-games.asp>
- Time. 2010. Julkaistu 25.10.2010. Luettu 6.11.2012.
<http://www.time.com/>
- Unity. Script Reference. AudioSource.PlayClipAtPoint. Luettu 1.10.2012.
<http://docs.unity3d.com/Documentation/ScriptReference/AudioSource.PlayClipAtPoint.html>
- Unity. Script Reference. Collider. Luettu 27.8.2012.
<http://docs.unity3d.com/Documentation/ScriptReference/Collider.html>
- Unity. Script Reference. MonoBehaviour.OnGUI. Luettu 23.8.2012.
<http://docs.unity3d.com/Documentation/ScriptReference/MonoBehaviour.OnGUI.html>
- Unity. 2009. Reference Manual. Mesh Collider. Päivitetty 16.7.2009. Luettu 23.8.2012.
<http://docs.unity3d.com/Documentation/Components/class-MeshCollider.html>
- Unity. 2011a. Camera. Päivitetty 10.11.2011. Luettu 31.8.2012.
<http://docs.unity3d.com/Documentation/Manual/Cameras.html>
- Unity. 2011b. Optimizing Performance in iOS. Luettu 17.4.2012.
<http://unity3d.com/support/documentation/Manual/iphone-performance.html>

Unity. 2011c. Unity Manual. Publishing Builds. Päivitetty 31.10.2011. Luettu 10.9.2012.

<http://docs.unity3d.com/Documentation/Manual/PublishingBuilds.html>

Unity. 2012a. Flash. Luettu 23.8.2012.

<http://unity3d.com/unity/publishing/flash>

Unity. 2012b. iOS Hardware Guide. Päivitetty 11.7.2012. Luettu 28.8.2012.

<http://docs.unity3d.com/Documentation/Manual/iphone-Hardware.html>

Unity. 2012c. License comparisons. Luettu 23.8.2012. <http://unity3d.com/unity/licenses>

Unity. 2012d. Unity Manual. Prefabs. Päivitetty 20.1.2012. Luettu 23.8.2012.

<http://docs.unity3d.com/Documentation/Manual/Prefabs.html>

Unity. 2012e. Unity Manual. Using Components. Päivitetty 19.6.2012. Luettu 28.8.2012.

<http://docs.unity3d.com/Documentation/Manual/UsingComponents.html>

Unity. 2012f. Unity Manual. Using Scripts. Päivitetty 28.6.2012. Luettu 28.8.2012.

<http://docs.unity3d.com/Documentation/Manual/Scripting.html>

Wiebe, R. 2011. Unity iOS Essentials. Birmingham: Packt Publishing.

Wikipedia. 2012. iPad. Päivitetty 30.10.2012. Luettu 30.10.2012.

<http://en.wikipedia.org/wiki/Ipad>