Modulaarinen lähestymistapa 3D-mobiilipelin kenttäsuunnitteluun

Ilkka Siik

Opinnäytetyö
Joulukuu 2013
Viestintä
Vuorovaikutteinen media

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Viestinnän koulutusohjelma
Vuorovaikutteinen media

ILKKA SIIK:
Modulaarinen lähestymistapa 3D-mobiilipelin kenttäsuunnitteluun

Opinnäytetyö 44 sivua, joista liitteitä 1 sivua
Joulukuu 2013

Opinnäytetyön aiheena oli tutkia ja soveltaa modulaarista lähestymistapaa mobiilipelin kenttäsuunnitteluun. Työn tavoitteena on tutkia mobiililaitteita kehitysalustana peligraafikon näkökulmasta sekä muodostaa käsitys mobiilialustan haasteista ja erityispiirteistä. Opinnäytetyön aikana kehitän prosesseja ja käytäntöjä, joiden avulla syntyy modulaarinen pelikenttä Dicework Gamesin ja 10tons Oy:n kaupalliseen fantasiamobiilipeliin nimeltä Trouserheart.

Tutkimusvaiheessa käyn läpi modulaarisen peligrafiikan prosesseja ja käytäntöjä. Lisäksi tutkin esimerkkien kautta modulaariselle kenttäsuunnittelulle ominaisia haasteita sekä tapoja niiden ratkaisemiseen. Pyrin antamaan myös vinkkejä käytännöllisten työkalujen sekä ohjelmistojen valintaan erityisesti indie-pelinkehittäjille.

Opinnäytetyön käytännön ja projektiosuutena käyn läpi mobiilipelikentän toteutusprosessin. Prosessi alkaa modulaaristen osien suunnittelusta sekä pelikentän konseptoinnista, jota seuraavat peliobjektien mallintaminen, teksturointi ja viimeistely. Lopuksi käyn läpi käytännöllisiä optimointiprosesseja sekä tekniikoita, joiden avulla grafiikan suorituskykyä parannetaan merkittävästi optimaalisen pelikokemuksen aikaansaamiseksi. Trouserheart voitti Unity Technologiesin ja Microsoftin järjestämän pelinkehityskilpailun ensimmäisen palkinnon pian pelin julkaisun jälkeen Marraskuussa 2013. Lisätietoja pelistä on osoitteessa http://www.trouserheart.com/.

Asiasanat: mobiilipelit, modulaarisuus, kenttäsuunnittelu, grafiikka

ABSTRACT

Tampere University of Applied Sciences
Degree programme in Media
Interactive Media

ILKKA SIIK:
Modular approach to 3D mobile game environment production

Bachelor's thesis 44 pages, appendices 1 pages
December 2013

The subject of this thesis was researching and applying modular approach to environment design for a mobile game. The objective of this thesis was to research mobile devices as a development platform from a game artist's perspective, and to form an understanding about the challenges and features of mobile platforms. During the thesis I develop processes and practises that are put to use in producing a modular game environment for a commercial fantasy mobile game by Dicework Games and 10tons Ltd.

In the research part of this thesis I analyse the processes and techniques that are used in modular design. I analyse various challenges and features of modular design and different ways to solve them by using practical examples. I also aim to give tips for selecting tools and programs that are useful especially for indie game developers.

In the practical and project phase I go through the development processes of creating a mobile game environment. I start the process by planning and concepting of the environment and modular pieces. Next steps are modeling, texturing and finalizing the game-ready modular assets. Lastly I go through practical optimization processes and techniques that significantly increase the graphical performance to ensure optimal gameplay experience. Soon after the game's release Trouserheart won the first prize in a game development competition arranged by Unity Technologies and Microsoft on November 15th 2013. For more information about the game you can visit http://www.trouserheart.com/

Key words: mobile games, modularity, environment art, graphics

SISÄLLYS

1 JOHDANTO 7
2 MOBIILILAITTEET PELIEN KEHITYSALUSTANA 8
2.1 Mobiilipelien historia 8
2.2 Mobiililaitteet tänä päivänä 12
2.3 Mobiililaitteet peligraafikon näkökulmasta 15
3 MODULAARISUUS 16
3.1 Modulaarisen grafiikan käyttö tämän päivän peleissä 16
3.2 Modulaarisuuden hyödyt pelituotannossa 18
3.3 Modulaarisuuden haasteet 22
4 TYÖKALUT 23
4.1 Mallintaminen 23
4.2 2D-grafiikka 23
4.3 Pelimoottori 24
5 PROJEKTI: TROUSERHEART (iOS / Android) 26
5.1 Suunnittelu 27
5.2 Osien mallintaminen 29
5.3 Teksturointi 31
5.4 Pelikentän kokoaminen pelimoottorissa 33
6 OPTIMOINTI PELIMOOTTORISSA 36
6.1 Geometria 36
6.2 Tekstuurit 37
6.2.1 Alpha 38
6.3 Colliderit 40
7 YHTEENVETO 41
8 LIITTEET 43

LYHENTEET JA TERMIT

Android	Googlen Linuxiin pohjautuva käyttöjärjestelmä
iOS	Applen kehittämä käyttöjärjestelmä, jota käytetään mm. kaikissa Applen mobiililaitteissa.
Mesh	Mallintamalla luotu 3D-kehikko tai pinta.
GPU	Grafiikkaprosessori
Shader	Ohjelma, jolla voidaan muokata GPU:lla prosessoidun grafiikan piirtotapaa.
Polygoni	Monikulmio, joka muodostuu neljästä pisteestä 3D-avaruudessa
Verteksi	Piste jolla on koordinaatit 3D-avaruudessa.
UV mapping	Tekniikka, jonka avulla määritetään tekstuuri 3D-mallin pinnalle.
Alpha	2D bittikartan alpha-kanava. Käytetään useimmiten läpinäkyvyyden toteuttamiseen 3D-objektien tekstuureissa.
Draw Call	Prosessorilta grafiikkaprosessorille lähetetty piirtokutsu.
FPS	Frames per second on arvo, jolla mitataan näytölle sekunnissa piirrettyjen kuvien määrää.
Normal map	Normal map eli normaalikartta on bitmap-kuva jonka avulla muutetaan 3D-grafiikan tekstuurin piirtotapaa suhteessa valaistukseen.

FPS	Frames per second on arvo, jolla mitataan näytölle sekunnissa piirrettyjen kuvien määrää.
Post-processing	Post-processing eli jälkikäsittely on grafiikkaprosessorin kautta toteutettu visuaalinen efekti.
Collider	Pelimoottorin käyttämä objekti jonka avulla mm. kerätään törmäysdataa eri collidereiden välillä.

1 JOHDANTO

Opinnäytetyössäni käsittelen modulaarista lähestymistapaa pelikentän suunnitteluun ja toteuttamiseen mobiilipeliin Unity 3D -pelimoottorissa. Tavoitteenani on tutkia ja analysoida mobiililaitteita kehitysalustana, sekä tuoda esille merkittävimmät modulaarisen kenttäsuunnittelun tekniikat sekä hyödyt. Havaintojen pohjalta syntyy käsitys modulaarisen kenttäsuunnitteluun liittyvistä eduista ja rajoitteista sekä käytännön ohjeistusten ja tekniikoiden hyödyntämisestä optimoidun pelikentän toteuttamisessa mobiilialustoille. Suurin osa esittelemistäni esimerkeistä liittyy Dicework Gamesin Trouserheart -mobiilipeliin, jonka artistina työskentelin.

Opinnäytetyöni projektiosuus käsittelee modulaarisen pelikentän toteutusprosessia 3D-graafikon näkökulmasta. Projektin toteutus alkaa modulaaristen elementtien suunnittelusta, mallintamisesta, testauksesta ja jatkuu tekstuurien sekä lopullisten 3D-objektien viimeistelyyn. Seuraavaksi pelikentän osat tuodaan pelimoottoriin, jossa pelikenttä kootaan, valaistaan ja viimeistellään. Työni viimeisessä osassa käydään läpi projektin graafinen optimointi. Optimointiosassa esittelen käytännöllisiä optimointitekniikoita erityisesti 3D-grafiikkaan sekä Unity3D:n ominaisuuksiin liittyen.

Opinnäytetöprojektini lopputuloksena syntyy modulaarisen kenttäsuunnittelun vahvuuksia hyödyntävä pelikenttä, jonka toteutuksessa on otettu huomioon mobiilialustojen tuomat haasteet sekä suoraviivaiset tuotantoprosessit peligrafiikan optimaaliseen tuottamiseen.

Työni on suunnattu graafikoille, joilla on jo kokemusta peligrafiikan tuottamisesta ja jotka työskentelevät pelituotannossa. Työssäni oletetaan lukijalla olevan tietyn tason ymmärrys 3D-grafiikasta ja siihen liittyvistä yleisistä osa-alueista, kuten tekstuurien tuottamisesta.

Työssäni käsittelen modulaarisen peligrafiikan tuotantoa sekä pyrin esittämään hyödyllisiä tekniikoita optimoidun grafiikan tuottamiseen mobiililaitteille. Työstäni voivat hyötyä pelien kenttäsuunnittelusta kiinnostuneet henkilöt sekä pelialasta kiinnostuneet graafisen alan opiskelijat.

2 MOBIILILAITTEET PELIEN KEHITYSALUSTANA

Mobiililaitteen keskeisimät ominaisuudet ovat kannettava koko sekä langattomuus. Mobiililaitetta voi kuljettaa mukana ja se on helposti käytettävissä. Laitteen fyysiset ominaisuudet, kuten pieni koko ja kevyt paino luovat rajoituksia, joiden seurauksena komponenttien tehokkuus ei välttämättä ole laitteiden, kuten pöytätietokoneiden tai pelikonsolien tasolla. Suurin osa mobiililaitteista on kehitetty ensisijaisesti viestinnällisiin tarkoituksiin, kuten puheluiden ja tekstiviestien välitykseen. Viihdekäyttöön suunnattujen tablet-mallisten mobiililaitteiden merkittävimpiä etuja puhelimiin nähden ovat fyysinen käytettävyys, näytön koko sekä kuvanlaatu.

Mobiililaitteiden rajoitteista johtuen laitteille suunnattujen ohjelmistojen, kuten pelien kehityksessä on otettava erityisesti huomioon grafiikan sekä koodin optimointi. Mobiililaitteiden erityispiirteet, kuten näytön koko sekä laitteen muotoilu voivat usein aiheuttaa käytettävyyteen liittyviä haasteita, jotka pelinkehittäjien täytyy ottaa huomioon. Nykypäivän standardiksi noussut kosketusnäyttöteknologia luo uusia mahdollisuuksia innovatiivisille peleille, mutta voi myös tuoda esiin uusia käytettävyyteen liittyviä haasteita.

2.1 Mobiilipelien historia

Ensimmäiset kannettaville laitteille kehitetyt pelit kehitettiin jo 1970-luvulla. (Skooldays.com. Artikkeli. 1.) Texas Instruments, yritys joka tunnettiin parhaiten taskulaskinten kehittäjänä, julkaisi vuonna 1976 lapsille suunnatun taskuun sopivan laitteen nimeltä Little Professor. Laite ei varsinaisesti ollut taskulaskin vaan matemaattisia ongelmia, kuten kerto- tai jakolaskuja esittävä opetuspeli joka pisteytti vastaukset.

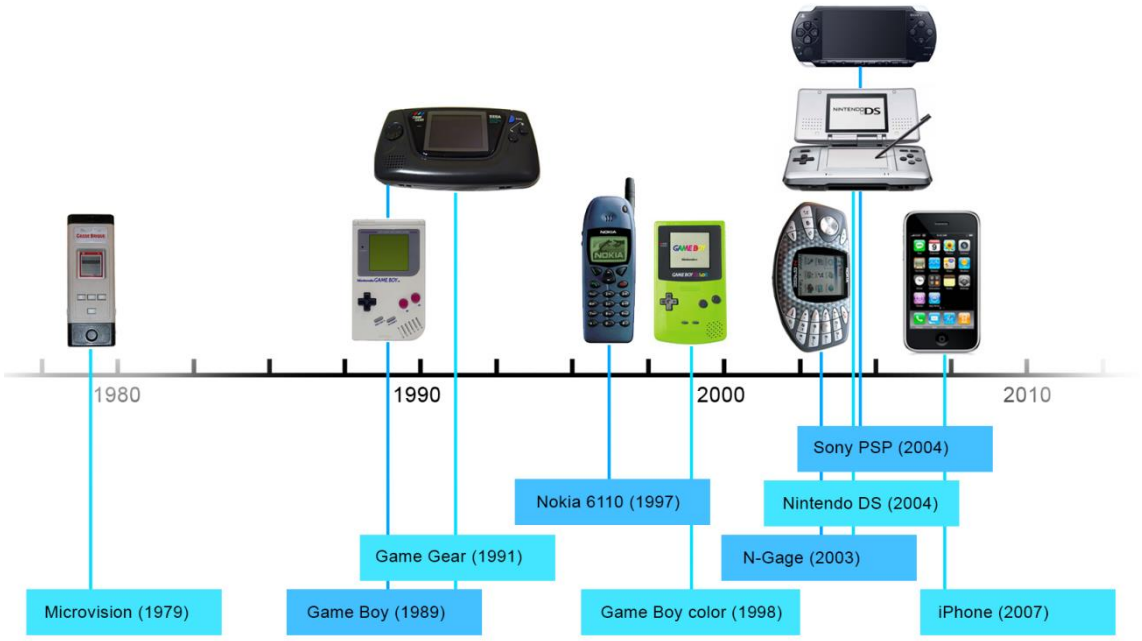
Lähivuosina useat yritykset julkaisivat kannettavia nestekidenäytöillä toimivia pelejä. Myöhemmin 1979 ilmestynyttä kannettavaa Microvisionia voidaan pitää maailman ensimmäisenä mobiilipelikonsolina. Microvision menestyi aluksi hyvin, mutta mielenkiinto pieninäyttöistä ja teknisistä ongelmista kärsivää laitetta kohtaan kuitenkin hiipui ja Microvision -laitteet katosivat markkinoilta vuoteen 1981 mennessä.

Japanilainen elektroniikkayritys Nintendo toi vuonna 1989 markkinoille kannettavan Game Boy -pelikonsolin, jonka suosio nosti kannettavat pelikonsolit suuren yleisön tietoisuuteen. Game Boyn 160 x 144 pikselin 2-värinen LED-näyttö päivittyi myöhemmin vuonna 1998 värinäyttöön, joka pystyi yhtäaikaisesti piirtämään enintään 56 yksittäistä väriä. (McGraw-Hill. BusinessWeek. 2008. 3.) Vuoteen 2008 mennessä Game Boy -laitteita oltiin myyty 118.7 miljoonaa kappaletta.

Kuva 1. Mobiilipelikonsolien alkuhistorian suosituimmat laitteet

Vuonna 1991 Sega Enterprises toi markkinoille uuden kilpailijan, Sega Game Gearin. (McGraw-Hill. 4.) Game Gearin täysin taustavalaistu, 32-värinen näyttö sekä merkittävästi tehokkaampi laitteisto eivät kuitenkaan pystyneet syrjäyttämään markkinajohtajaa Nintendoa Game Boyn suosion ja laajan pelikatalogin takia. Vuonna 2004 Nintendo julkaisi uuden mobiilipelikonsolin nimellä Nintendo DS. Laitteen menestyksestä kertoo erittäin kattava pelikatalogi sekä 50 miljoonaa myytyä konsolia vuoteen 2007 mennessä. (Rivington, James. Artikkeli. 9.) Pian Nintendo DS:n julkaisun jälkeen samana vuonna markkinoille ilmestyi myös kilpaileva mobiilipelikonsoli, Sony PlayStation Portable, josta yleisesti käytetään lyhennettä PSP.

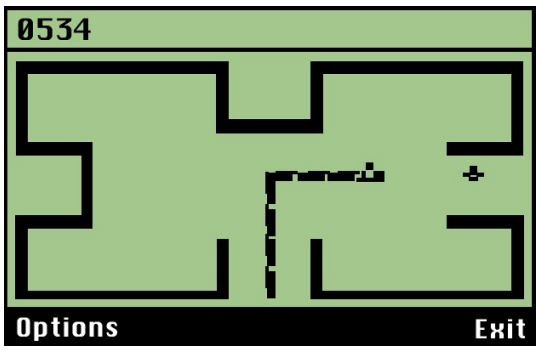
Viimeisen vajaan vuosikymmenen aikana mobiilipelikonsolien markkinoita ovatkin hallinneet lähes poikkeuksetta Sony ja Nintendo.



Kuva 2. Mobiililaitteiden kehitys – merkittävimmät julkaisut 1979-2008

Game Boy colorin julkaisuun mennessä myös matkapuhelimet olivat kehittyneet vuonna 1983 julkaistusta, 840 gramman painoisesta Motorola DynaTAC 8000X:sta käyttäjäystävällisemmiksi laitteiksi, joille oli mahdollista sisällyttää myös pelejä. (GSMHistory.com. 5.) Ensimmäinen matkapuhelimelle julkaistu peli oli Tetris, joka ilmestyi vuonna 1994 Tanskassa kehitetylle Hagenuk MT-2000:lle.

Kolme vuotta myöhemmin Nokia saavutti seuraavan mobiilipelien rajapyykin (Joel Willans. Artikkeli. 6.) julkaisemalla Nokia 6110-mallin matkapuhelimen. Puhelimeen oli esiasennettuna Snake -niminen mobiilipeli, joka saavutti suuren suosion miljoonien käyttäjien keskuudessa. Seuraavina vuosina Snakea asennettiin yhteensä yli 350 miljoonalle laitteelle. Snaken menestyksen seurauksena mobiilipuhelimet alkoivat vaikuttaa varteenotettavalta alustalta pelinkehittäjille. Myös Tetris teki uuden paluun, tällä kertaa Nokian matkapuhelimille vuonna 2000.



Kuva 3. Mobiilipeli Snake, Nokia 6110

Vuonna 2003 Nokia julkaisi pelikäyttöön suunnatun puhelimen nimeltä N-Gage (Snow, Blake. GamePro. 7.) Laitteen menestys jäi kuitenkin erittäin vähäiseksi. N-Gagen suurimmat haittapuolet olivat korkea julkaisuhinta, käytettävyysongelmat sekä keskinkertainen suunnittelu. Laitetta myytiin lopulta vain 3 miljoonaa yksikköä. Nokia N-Gagea voidaan pitää yhtenä 2000-luvun suurimmista epäonnistumisista mobiiliteollisuuden alalla.

Kuva 4. Nokia N-Gage (Nokia Oyj)

Mobiilipelien aseman yhtenä peliteollisuuden merkittävimmistä sektoreista lujitti lopulta Applen markkinoille vuonna 2007 julkaisema iPhone. iPhone ja seuraavana vuonna julkaistu latauskanava App Store tarjosivat matalan kynnyksen pelinkehittäjille sovellusten kehittämiseen sekä suoran myyntikanavan pelinkehittäjien ja kuluttajien välille. App Storen kautta mobiililaitteille kehitettyjä ohjelmistoja voitiin ensimmäistä kertaa myydä kuluttajille ilman välikäsiä, kuten puhelinoperaattoreita tai julkaisijoita. Jo vuonna 2009 App Storesta tehdyt lataukset ylittivät miljardin rajan. (O'Brien, Terrence. Engadget. 8.) Tähän mennessä App Storen julkaisun jälkeen palvelusta on ladattu yli 50 miljardia sovellusta ja iPhone -mobiilipuhelimia on myyty yli 500 miljoonaa kappaletta.

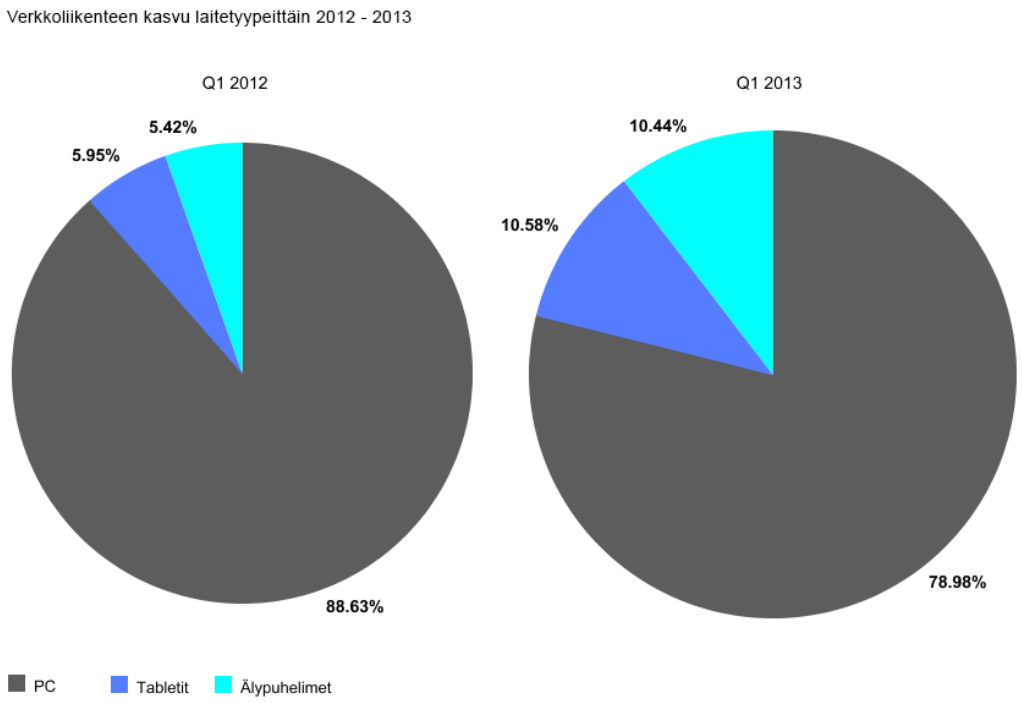
Kuva 5. Apple iPhone, 2007 (Apple Inc)

Applen iOS -mobiililaitteiden suurimmaksi kilpailijaksi nousivat lopulta Android -käyttöjärjestelmällä varustetut älypuhelimet. Android 1.0 julkaistiin Open Handset Alliancen toimesta vuonna 2008. Open Handset Alliance on ryhmä jonka tarkoitus on mobiiliteknologian kehittäminen ja johon kuuluu tänä päivänä 84 teknologia-alan yritystä. Google on yksi Open Handset Alliancen tunnetuimmista vaikuttajista. Ensimmäinen Android -käyttöjärjestelmällä julkaistu mobiililaite HTC Dream sai enimmäkseen positiivisen vastaanoton, mutta ei pystynyt vielä kilpailemaan iPhonen kanssa. Samaan aikaan Google julkaisi Android Marketin, joka toimii App Storen tapaan Android -käyttöjärjestelmälle kehitettyjen sovellusten myynti- ja latauskanavana. Myöhemmin vuonna 2010 Google julkaisi yrityksen oman Android -tuoteperheen nimellä Nexus.

2.2 Mobiililaitteet tänä päivänä

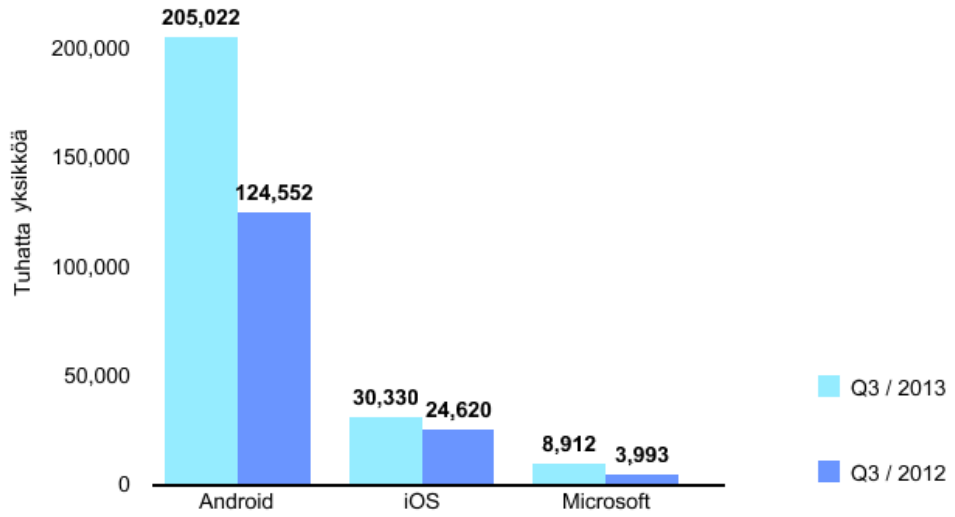
Viimeisen vuosikymmenen aikana mobiililaitteet ovat kehittyneet aktiiviseksi osaksi ihmisten päivittäistä elämää. Varsinkin 2000-luvun ensimmäisen vuosikymmenen jälkeen yhä useammat yritykset ovat alkaneet reagoida mobiilialan vahvaan kasvuun.

Mobiililaitteet näkyvät kaikkialla päivittäisessä elämässämme. Virvoitusjuomapulloissa voi nähdä mobiililaitteilla kuvattavia QR-kampanjakoodeja sekä linkkejä mobiilipeleihin. App Storesta voi löytää tuttuihin tuotemerkkeihin liittyviä viihdesovelluksia. Ihmiset päivittävät Facebook-tilansa ja tarkistavat sähköpostin vaikkapa odotellessaan bussia. Mobiililaitteille on saatavissa miljoonia sovelluksia lukemattomiin käyttötarkoituksiin. Applen viimeisimmän päivityksen mukaan (Artikkeli. Apple. Q3 2013. 12.) App Storessa on yli 900,000 sovellusta, joista 375,000 on suunnattu tableteille. Seuraavassa kuviossa (Kuvio 1) voidaan selvästi huomata huomattava kasvu mobiililaitteiden globaalissa verkkoliikenteessä yhden vuoden aikana.



Kuvio 1. Verkkoliikenteen kasvu laitetyypeittäin 2012-2013. Lähde: Gartner.

Tämän hetken merkittävin kasvusuunta on Android -käyttöjärjestelmää hyödyntävissä mobiililaitteissa. Syinä Androidin menestykseen ovat mielestäni käyttöjärjestelmän avoimuus, laitteiden monipuolisuus sekä hintaerot verrattuna suurimman kilpailijan Applen tuotteisiin. Kun otetaan huomioon nopeasti kehittyvät maat, joissa on erittäin paljon potentiaalisia ostajia huokeammille sekä kehitysalustaltaan avoimille mobiililaitteille, voidaan olettaa Android -laitteiden menestyksen kasvavan myös tulevina vuosina. Näkemystäni puoltavat myös Gartnerin viimeisimmät tulokset, (Gartner Inc. Analyysi 2013. 11.) joissa Android -käyttöjärjestelmää käyttävän Samsungin markkinaosuus Aasiassa on jo yli kaksi kertaa suurempi kuin Applen.



Kuvio 2. Suosituimpien mobiilikäyttöjärjestelmien myynti Q3/2012 – Q3/2013. Lähde: Gartner.

Tämän päivän suurimmat mobiililaitteiden valmistajat ovat markkinaosuudeltaan Samsung (30.4%) sekä Apple (19.4%). Hallitseva mobiilikäyttöjärjestelmä on poikkeuksetta Android jota käyttää 66.8% vuonna 2012 myydyistä mobiililaitteista. (International Data Corporation. Analyysi. 10.)

Kuva 6. Tablet-malliset laitteet; seuraava askel mobiiliteknologian kehityksessä.

Tablet -malliset mobiililaitteet ovat myös viime vuosien aikana nousseet merkittävään asemaan puhelimien rinnalle. Tabletteja käytetään useimmiten kotona ja yksittäiset käyttöajat ovat suhteessa pidempiä kuin pienemmillä mobiililaitteilla. Yhä useampi älypuhelimen käyttäjä omistaa nykyään myös tabletin, joka hankitaan yleisimmin viihdekäyttöön. Suurempi kosketusnäyttö on fyysisesti helppokäyttöisempi ja antaa paremman käyttökokemuksen verrattaessa älypuhelimiin. Erityisesti Applen iPad -tuotesarjan uusimmissa tableteissa näytön tarkkuus on vähintään keskihintaisen pöytätietokoneen LCD-näytön tasolla.

2.3 Mobiililaitteet peligraafikon näkökulmasta

Verrattaessa mobiililaitteita pelikonsoleihin tai pöytätietokoneisiin suurimmat erot alustojen välillä ovat koko, suorituskyky sekä käyttötapa.

Mobiililaitteet kehittyvät nopeasti ja ero konsoleihin sekä PC:hin kaventuu jatkuvasti. On kuitenkin oletettavaa, ettei mobiililaitteiden suorituskyky tule nykytekniikalla olemaan suosituimpien pelikonsolien tasolla mobiililaitteiden komponentteihin kohdistuvien erityisvaatimusten takia. Mobiililaitteiden ja pelikonsolien ominaisuudet sekä tekniset ominaisuudet, kuten kehittyneemmät piirtotekniikat ovat kuitenkin jo tänä päivänä hyvin lähellä toisiaan. Toisaalta tekniikoiden kuten normaalikarttojen tai koko ruudun jälkikäsittelyefektien käyttäminen mobiilialustoilla on useimmissa tapauksissa epäkäytännöllistä ruudun päivitysnopeuden (FPS) laskiessa murto-osaan optimaalisesta 60:sta.

Peligraafikon täytyy ottaa erityisesti huomioon haasteet kohdealustan suorituskyvyssä, sekä pyrkiä käyttämään laitteen vahvuuksia mahdollisuuksien mukaan. Esimerkiksi älypuhelinten, kuten iPhone 4:n GPU:n fillrate saattaa muodostua pullonkaulaksi, mikäli käytetään monimutkaisia shadereita tai liiallista alpha-piirtoa. Mobiilialustalle kehitettäessä erityisesti grafiikan optimoinnin merkitys on suuri. Mobiililaitteet asettavat graafikolle myös käytettävyyteen liittyviä haasteita kuten kosketusnäytön ominaisuuksien sekä käyttötapojen huomioon ottaminen.

3 MODULAARISUUS

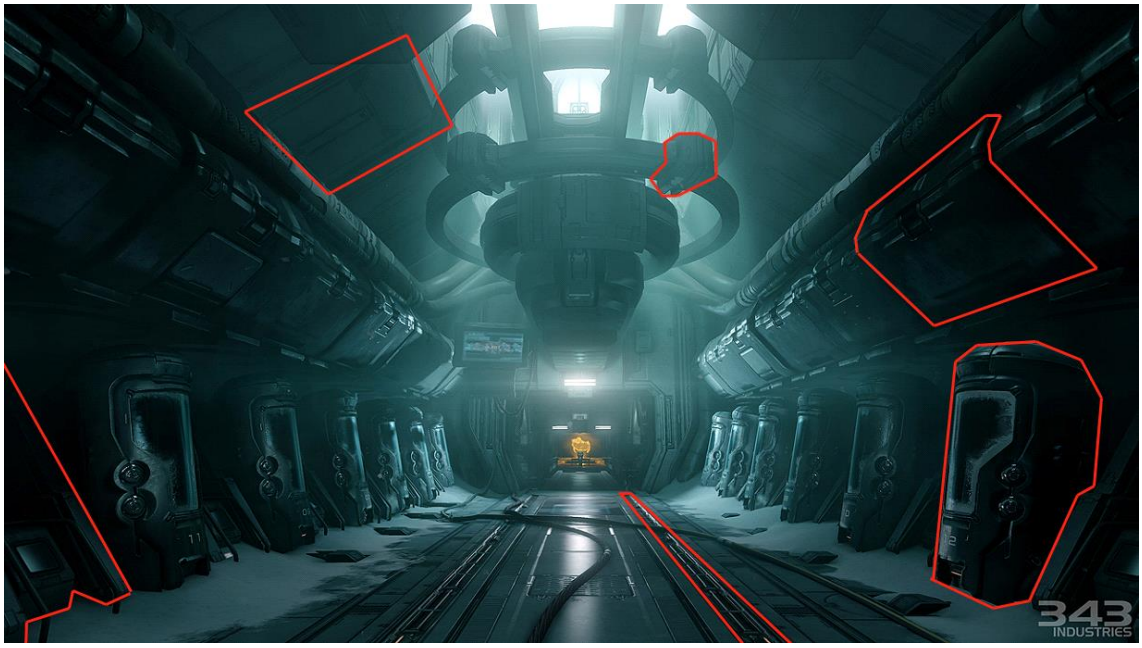
Epic Games määrittelee modulaarisuuden seuraavasti:

"Modular design is concerned with making lots of high-quality chunks of levels and reusing those chunks intelligently" joka voidaan vapaasti suomentaa: *"Modulaarinen suunnittelu on useiden korkealaatuisten pelikentän osien tuottamista ja niiden järkevää uudelleenkäyttöä"*.

Modulaarisuus tulee sanasta moduuli. Moduuli tarkoittaa yksittäistä osaa, josta voidaan koota erilaisia kokonaisuuksia kuten taloja, laivoja tai vaikkapa pelikenttiä.

3.1 Modulaarisen grafiikan käyttö tämän päivän peleissä

Modulaarisen peligrafiikan käyttö erityisesti pelien kenttäsuunnittelussa on nykypäivänä erittäin suosittua. Modulaarista lähestymistapaa on viime vuosina käytetty useissa menestyneissä pelituotannoissa kuten Minecraft, Elder Scrolls 4: Skyrim, Mass Effect tai Halo sekä monissa muissa. Seuraavassa kuvassa pyrin havainnollistamaan punaisella värillä modulaarisia osia joista esimerkkikuvan pelikenttä on koottu (Kuva 8).



Kuva 8. Modularity of Halo 4, Microsoft 343 Industries Studio (2013)

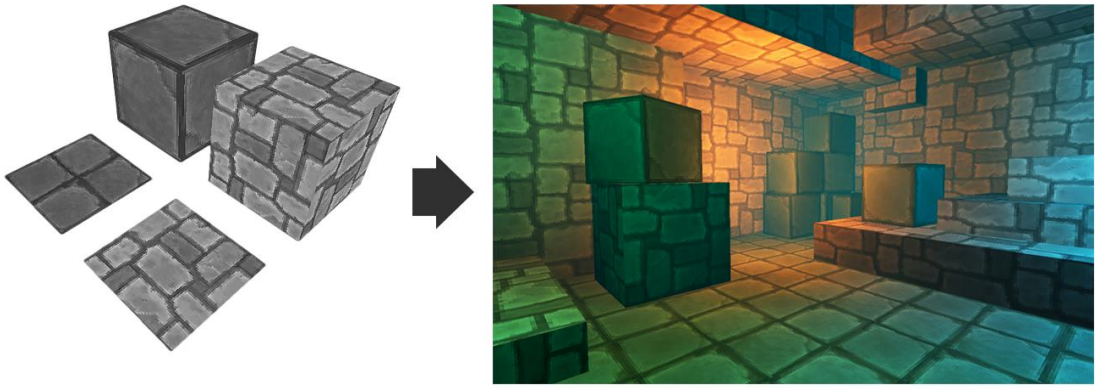
On monia syitä miksi pelituotannoissa voidaan käyttää modulaarisia objekteja. Modulaarisuus mahdollistaa erimerkiksi suurten pelikenttien tuottamisen pienemmällä vaivalla. Tuotantoaikojen lyheneminen sekä pelin suorituskyvyn paraneminen ovat etenkin 3D-peleissä modulaarisuuden merkittävimpiä etuja. Oikein käytettynä modulaarinen kenttäsuunnittelu voi mahdollistaa monimutkaisen ja laajan pelimaailman tuottamisen merkittävästi pienemmillä resursseilla. Käyn modulaarisuuden hyötyjä tarkemmin läpi työn seuraavassa osassa.

Kuva 9. Deus Ex: Human Revolution. Eidos Montréal. Wiktor Öhman.

Yllä olevassa kuvassa (Kuva 9) voidaan huomata modulaaristen osien käyttöä sekä toistuvuuden rikkomista yksittäisillä elementeillä. Modulaarinen lähestymistapa kenttäsuunnitteluun tuo mukanaan myös haasteita, sillä samojen modulaaristen objektien liiallinen käyttö vaikuttaa usein pelikokemukseen negatiivisesti. Jos sama objekti toistuu pelikentässä liian usein tai sitä käytetään väärin voi pelaaja tiedostaa sen, jolloin illuusio monimutkaisesta ja mielenkiintoisesta pelimaailmasta voi rikkoutua. Tämä haaste voidaan ratkaista esimerkiksi rikkomalla toistuvia elementtejä ja kuvioita mielenkiintoisemmilla dekoratiivisilla objekteilla tai muuttamalla yksittäisten objektien visuaalisia ominaisuuksia kuten väritystä, muotoa tai kokoa.

3.2 Modulaarisuuden hyödyt pelituotannossa

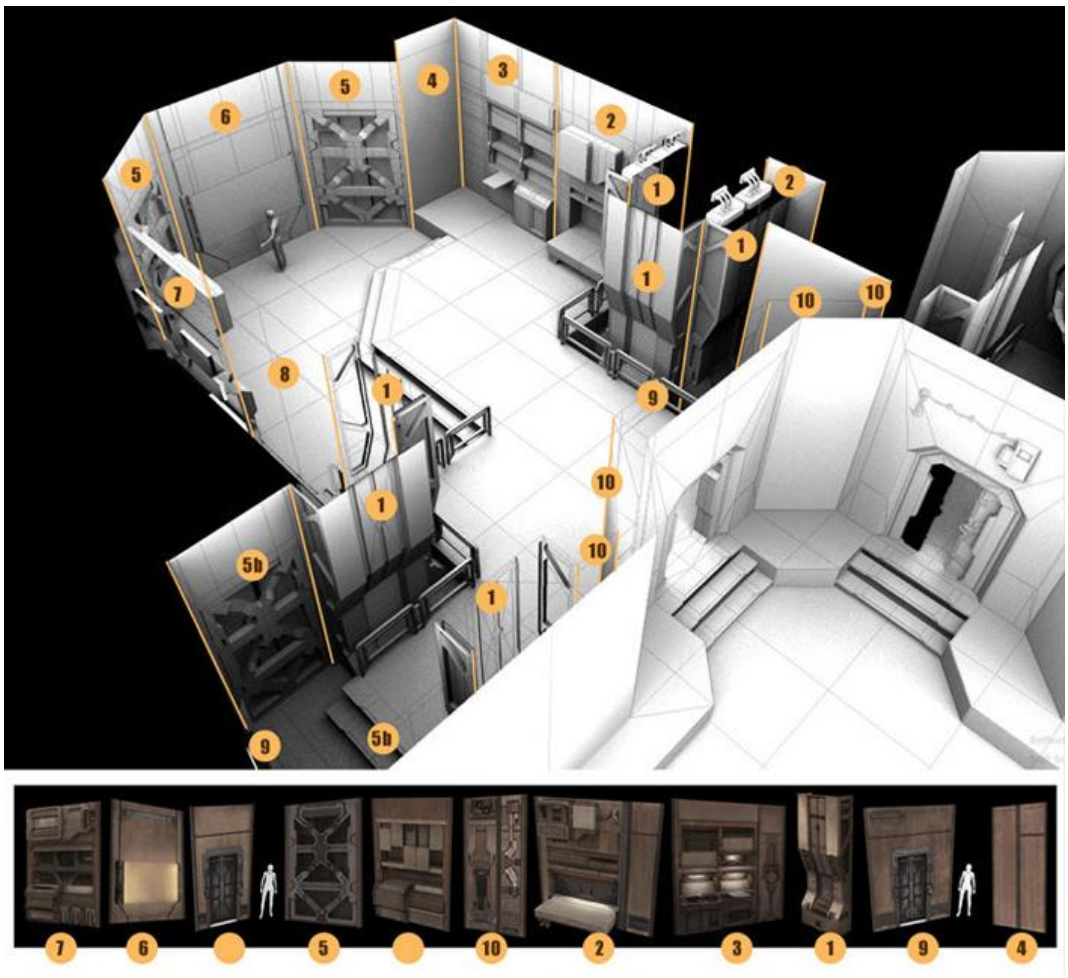
Modulaarisen peligrafiikan merkittävin ominaisuus on sen uudelleenkäytettävyys. Tästä syystä jopa yhdellä ainoalla modulaarisella objektilla voidaan toteuttaa kokonaisia pelikenttiä. Modulaaristen objektien käyttö mahdollistaa nopeamman kehitysprosessin pelimaailmojen luomiseen kuin useampien uniikkien objektien käyttö.



Kuva 10. Kahden eri modulaarisen objektin käyttö Unity 3D:ssa

Vaikka modulaarisuutta ei vietäisi niin pitkälle kuin edellä mainitussa esimerkissä, on se useimmissa tapauksissa merkittävästi nopeampi lähestymistapa pelikenttien toteuttamiseen.

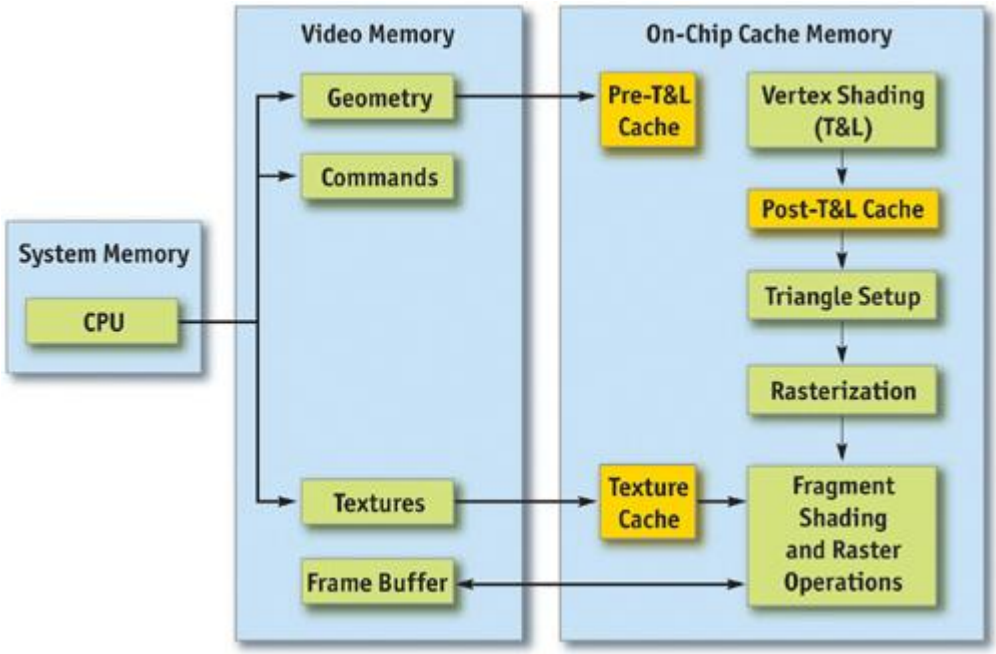
Epic Gamesin modulaarista kenttäsuunnittelua koskevassa artikkelissa (Epic Games 2011. 13.) mainitaan kuitenkin, että joissakin tapauksissa on nopeampaa tehdä pelikenttä uniikkeja objekteja käyttäen modulaaristen sijaan. Koska modulaaristen objektien suunnittelu vie enemmän aikaa, pienten pelikenttien, kuten yksittäisen huoneen toteuttamiseen se ei välttämättä ole optimaalista.



Kuva 11. Huoneen jakaminen modulaarisiin osiin. Eve Online. CCP games (2011)

Modulaaristen objektien käyttö voi myös nopeuttaa pelin suorituskykyä, sillä saman objektin kopioiden (instanssi) piirto on merkittävästi tehokkaampaa kuin usean uniikin objektin. Kun objekti ladataan kerran grafiikkaprosessorin muistiin, voidaan muistissa oleva data piirtää ruudulle saman piirtokäskyn aikana useita kertoja.

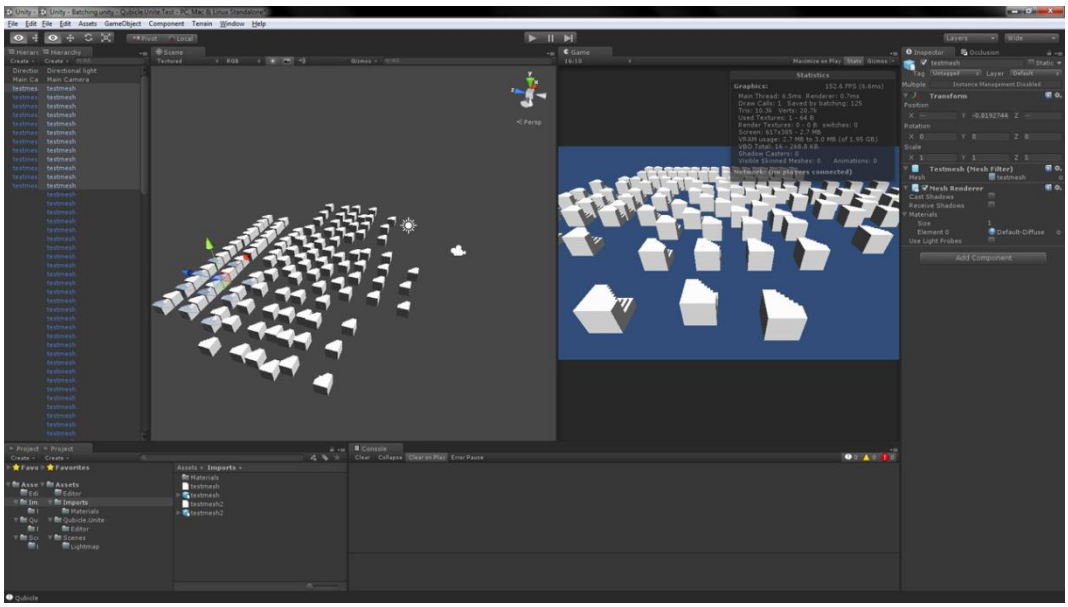
3D-mallien piirto alkaa järjestelmän muistista, josta data siirtyy prosessorin kautta videomuistiin. Videomuistissa data jakautuu kahdeksi yhtäaikaisesti liikkuvaksi osaksi; geometria sekä tekstuuridata. Tekstuuridata ja geometria toimivat eri instansseina ja siirtyvät videomuistista grafiikkaprosessorin välimuistiin piirrettäväksi yksittäisen piirtokäskyn aikana. Tapa jolla GPU piirtää objektin on kuitenkin pelimoottorin vastuulla.



Kuvio 3. GPU rendering pipeline. NVIDIA 2007.

Unity 3D –pelimoottorin data siirretään ensin keskusmuistista videomuistiin jossa data jakautuu kahteen prosessiin; geometria- sekä tekstuuridata (Kuvio 3). Komennot (Commands) ovat pelimoottorilta lähetettyä dataa joka ohjaa GPU:n operaatioita grafiikan piirrossa. Monimutkaiset shaderit vaativat kuitenkin useita piirtokäskyjä ja lähes poikkeuksetta rikkovat objektien sarjoitusprosessin (batching), joka mahdollistaa usean instanssin samanaikaisen piirron samalla piirtokäskyllä. Esimerkki monimutkaisemmasta shaderista on heijastavat pinnat, joiden prosessointi vaatii useita piirtokertoja.

Tästä syystä mobiilipeleissä käytetään useimmiten erittäin yksinkertaisia shadereita ja pyritään toteuttamaan tarvittavat erikoisefektit käyttäen muita tekniikoita ja kiertotapoja. Alla olevassa esimerkkikuviossa piirretään ruudulle 125 peliobjektia yhden piirtokäskyn aikana (Kuva 12).



Kuva 12. Unity 3D game object batching. Tobias Pott.

Tekstuurien piirtoa voidaan nopeuttaa vähentämällä piirtokäskyjä atlasointitekniikalla, jossa useita yksittäisiä tekstuureja yhdistetään yhdeksi bitmapiksi. Tekstuuriatlaksesta voidaan UV-koordinaattien avulla määrittää eri objekteille kuuluvat osat. Atlasointi vähentää piirtokäskyjä useista kerroista jopa vain yhteen ja sen käyttö on lähes poikkeuksetta kannattavaa erityisesti modulaaristen objektien suunnittelussa. Alla olevassa kuvassa voidaan nähdä atlasoitu tekstuuri (Kuva 13).

Kuva 13. Tekstuuriatlas. 3DMotive 2012.

3.3 Modulaarisuuden haasteet

Suurin haaste modulaaristen osien käytössä on se, että pelaaja saattaa huomata objektien uudelleenkäytön. Kyseinen tilanne on erittäin epätoivottava, koska se antaa vaikutelman huolimattomasta työstä sekä vaikuttaa myös pelattavuuteen liittyviin seikkoihin. Esimerkiksi modulaaristen osien liiallinen toistuminen saattaa aiheuttaa tilanteita, joissa pelaaja ei tunnista pelikentän alueita ja eksyy, koska pelialueet näyttävät liian samanlaisilta.

Tästä syystä pelikenttää kootessa artistin täytyy kiinnittää erityistä huomiota eri pelialueiden erottuvuuteen ja modulaaristen osien toiston rikkomiseen eri tavoin. Tapoja, joilla pelikentän eri alueita voi erotella on useita. Esimerkiksi pelikentän valaistuksen värejä sekä voimakkuutta voidaan muuttaa eri osissa. Modulaarisia osia voi käyttää eri järjestyksessä ja osien päälle voidaan asetella alpha-kanavaa käyttäviä ‘sprite’ -objekteja kuten vaikkapa graffiteja tai likaa. Modulaarisia osia suunnitellessa on myös hyvä ottaa huomioon objektien mahdollinen variointi pelkkää UV-karttaa tai materiaalia muuttamalla.

On valitettavan totta, että modulaarinen kenttäsuunnittelu luo sille ominaisia rajoitteita ja saattaa helposti johtaa objektien liialliseen toistuvuuteen. Sen edut ovat kuitenkin monessa tilanteessa niin merkittäviä, että modulaarisuuden käyttö on kiistämättä yksi tehokkaimmista peligrafiikan tuotantotavoista. Hyvin suunniteltujen modulaaristen objektien käyttö mahdollistaa laajankin pelimaailman täyttämisen korkealaatuisella 3D-grafiikalla suhteellisen pienellä tiimillä.

4 TYÖKALUT

Työssäni käytän useita pelinkehityksessä yleisesti käytössä olevia ohjelmistoja. Pyrin lyhyesti esittelemään ohjelmien keskeisimmät ominaisuudet sekä tarjoamaan myös vaihtoehtoisia ohjelmistoja, jotka sopivat esim. pienemmän budjetin tuotantoihin kuten indie-tiimeille tai opiskelijoille. Lähes kaikki tekniikat joita esittelen ovat toteutettavissa millä tahansa mainituista ohjelmistoista.

4.1 Mallintaminen

3D-mallien ja UV-karttojen toteuttamiseen käytän Autodesk 3D Studio Max sekä Blender -mallinnusohjelmia. Kaupallisesta 3D Studio Maxista poiketen Blender on ilmainen ja kehitetty GNU GPL -lisenssin alaisena.

3D Studio Max on vakiinnuttanut asemansa yhtenä alan käytetyimmistä mallinnusohjelmista kahden vuosikymmenen aikana ja sisältää erittäin kattavat työkalut kaikkiin tämän päivän 3D-tuotantojen tarpeisiin. Omien kokemuksieni mukaan Maxin erityiset vahvuudet ovat korkeatasoiset mallinnus- sekä renderöintityökalut. Ohjelman haittapuolina voidaan mainita suhteessa esim. Blenderiin merkittävästi pidemmät käynnistysajat sekä ohjelman vakaus.

Blenderin ominaisuuksista työssäni käytetään erityisesti UV-työkaluja. Blender on tänä päivänä yhä useamman peligraafikon valinta, sillä se tarjoaa lähes kaikki kaupallisten mallinnusohjelmien ominaisuudet ja ohjelmaa päivitetään huomattavasti useammin. Haittapuolina voidaan mainita dokumentaation puute sekä käyttöliittymä, joka poikkeaa monella tavalla yleisistä standardeista ja aiheuttaa monella aloittelevalla käyttäjällä ylimääräistä päänvaivaa.

4.2 2D-grafiikka

Tekstuurien ja konseptikuvien toteuttamiseen käytän Adobe Photoshop -tuoteperheen CS5 -versiota. Photoshoppia on pidetty alan standardina ja yleisimmin käytettynä

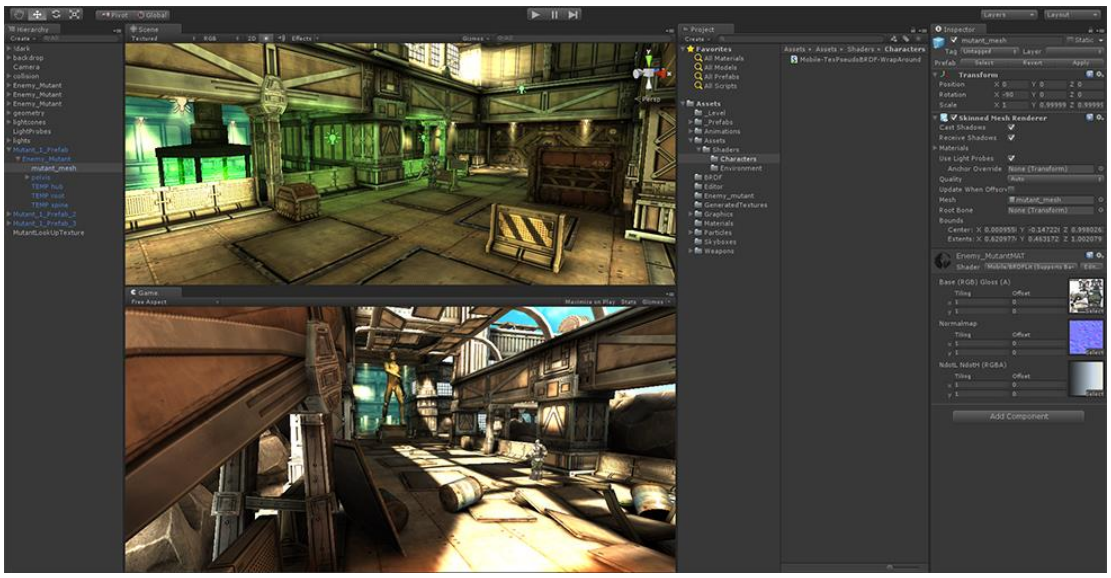
kuvankäsittelyohjelmana jonka monipuoliset työkalut mahdollistavat korkealaatuisten tekstuureiden tuottamisen.

Muita varteenotettavia kuvankäsittelyohjelmia peligraafikoille ovat Corel Painter sekä ilmainen Gimp. Mac OS X -käyttäjille on saatavilla myös Pixelmator jonka lisenssi on merkittävästi halvempi kuin Adoben tai Corelin tuotteet. Pixelmatorin voi ostaa ja ladata App Storesta.

4.3 Pelimoottori

Projektin pelimoottorina toimii Unity 3D. Unity on erittäin suosittu ja edullinen pelimoottori, jolla voidaan kääntää valmis projekti usealle eri alustalle. Tällä hetkellä Unity 3D 4.3 tukee seuraavia alustoja: PC, Mac, Linux, Xbox 360, PlayStation 3, Wii U, iOS, Android, Blackberry ja Windows Phone. Nimestä huolimatta Unity 3D:lla voi myös vaivatta tuottaa 2D-pelejä kuten Rovion Unityllä kehittämä (Artikkeli. Will Freeman. 14.) Bad Piggies. Unityn vahvuudet ovat helppokäyttöinen käyttöliittymä sekä yksinkertaiset mutta erittäin muokattavissa olevat ominaisuudet. Unityyn on saatavilla myös paljon yhteisön kehittämiä liitännäisiä joita ohjelman Asset Store -moduulin kautta voi ostaa ja ladata.

Unityn ominaisuuksien ansiosta pelimoottorin avulla voi kehittää pelien prototyyppejä erittäin nopeasti. Monet pelinkehityksessä käytetyt työkalut kuten PhysX-fysiikkamoottori sekä Mecanim-animaatiojärjestelmä ovat valmiiksi integroitu Unityyn, jonka takia niitä ei tarvitse alusta asti ohjelmoida itse. Unityyn on integroitu myös muita työkaluja kuten mm. Box2D -fysiikkamoottori 2D-peleille sekä Beast lightmapper valokarttojen luomiseen.



Kuva 7. Unity 3D:n editor-näkymä

Unityn käyttöliittymä (Kuva 7) on suhteellisen helppokäyttöinen ja selkeä. Pelikuvan voi nähdä Game -viewport alueella samalla kun objekteja muokkaa reaaliajassa toisessa näkymässä. Objekteja voi siirtää suoraan pelikentälle yksinkertaisesti vetämällä ne projektihierarkiasta Scene -näkymään.

Unity tukee lähes kaikkia yleisimpiä tiedostomuotoja joita peligrafiikan tuottamisessa käytetään. Pelissä käytettävien graafisten materiaalien tuonti Unityyn on tehty erittäin vaivattomaksi, sillä tuettuja tuontitapoja on mm. tiedostojen siirtäminen suoraan drag-and-drop -mekaniikalla hakemistoista tai työpöydältä Unityn editorissa sijaitsevaan projektihierarkiaan. Unityssä tiedostojen tuontiasetukset ovat tehty helposti muokattavaksi projektin tarpeiden mukaan.

5 PROJEKTI: TROUSERHEART (iOS / Android)

Trouserheartin kehitys aloitettiin täydellä teholla vuoden 2013 Tammikuussa Dicework Oy:n toimesta. Oma vastuualueeni projektissa on 3D-grafiikan sekä animaatioiden tuottaminen.

Trouserheartin värikkäällä ja käsinmaalatulla 3D-tyylillä pyritään erottumaan geneerisistä peligenren tuotteista ja luomaan vahva ja persoonallinen brändi. Helposti lähestyttävä ja positiivinen visuaalinen tyyli tukee pelin kevyttä toimintaa ja seikkailua.

Kuva 14. Trouserheart. Dicework Games (2013).

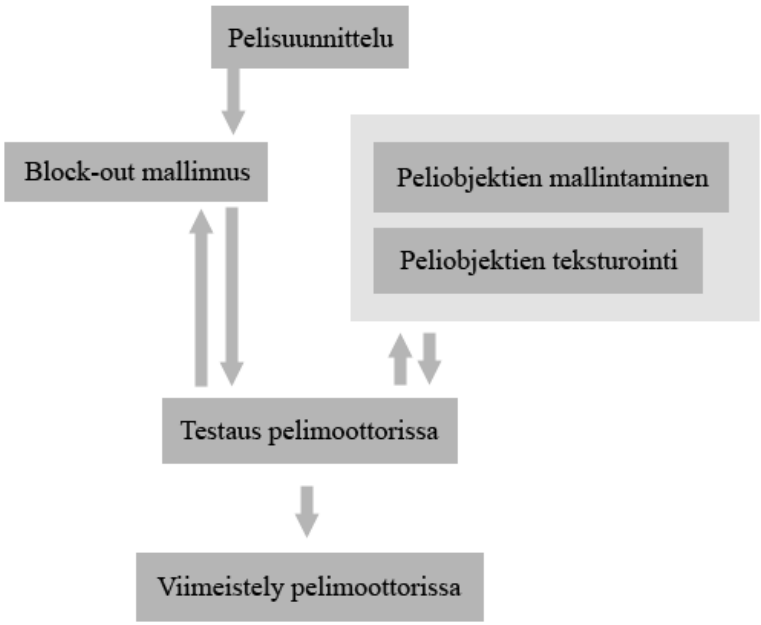
Pelissä kentät ovat lähes jokaisella pelikerralla erilaisia. Jokaisen latauksen aikana kenttä kootaan ohjelmallisesti ennalta määrättyjen sääntöjen mukaan arvotuista modulaarisista osista. Trouserheartin kentät koostuvat siis staattisesta kehikosta sekä kahdestatoista arvotusta kenttäpalasta, joita kutsutaan yhdessä tilesetiksi. Yllä olevassa kuvassa (Kuva 14) voidaan nähdä viimeistellyn kentän tileset. Pelin lopullisessa versiossa on 10 tilesettiä, joista jokaisessa on yksi staattinen ja 6 täysin erilaista dynaamista osaa. Kokonaisuudessaan pelin kentät koostuvat yhteensä 70:stä objektista, joista valtaosa on muuttuvia.

Kuvio 4. Modulaaristen objektien käyttö suhteessa uniikkeihin. Trouserheart.

Yllä olevassa kuviossa (Kuvio 4) havainnollistan koko pelin toteuttamiseen käytettyjen uniikkien sekä modulaaristen peliobjektien määrä suhteessa toisiinsa. Kuten kuviosta voidaan nähdä, uniikkien peliobjektien määrä on merkittävästi pienempi suhteessa modulaarisiin. Staattisten ja uniikkien kehikoiden toteutus vaatii suhteessa moninkertaisen ajan verrattuna modulaarisiin osiin. Yhden tilesetin tuottaminen vaatii yhdeltä graafikolta 10-15 työpäivää, joista suurin osa kuluu uniikin grafiikan, kuten pelialuetta ympäröivän kehikon toteuttamiseen.

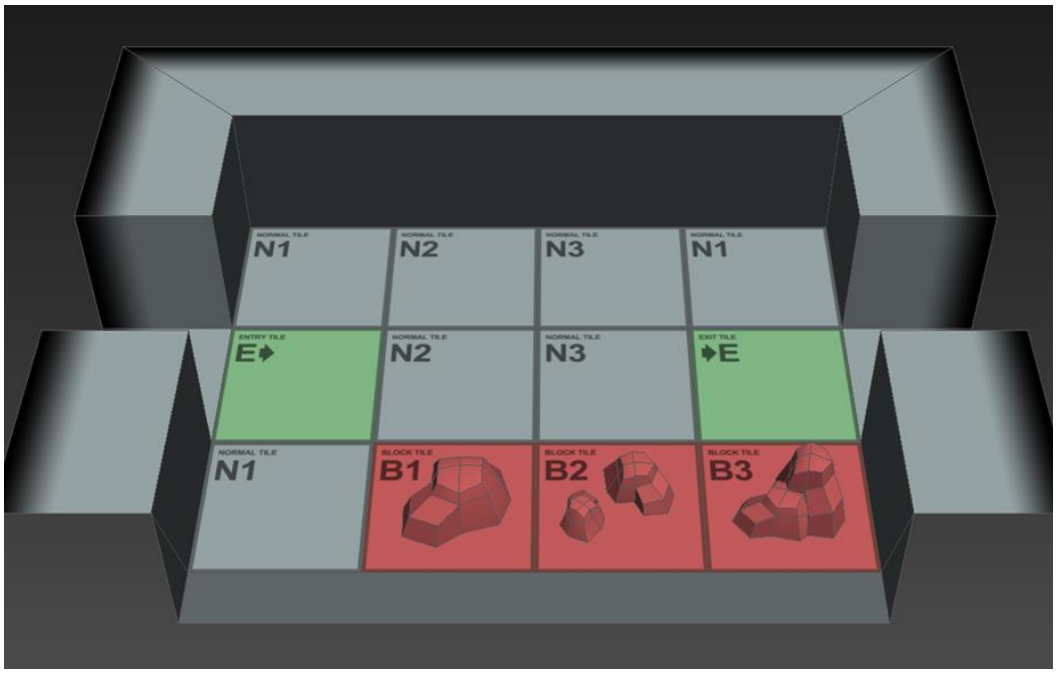
5.1 Suunnittelu

Aloitin modulaaristen objektien suunnittelun luomalla ensin grid-pohjaan yksinkertaisen kehikon, jonka pohjalta koko kentän mittasuhteet saadaan mahdollisimman aikaisessa vaiheessa selvitettyä. Tätä vaihetta kutsutaan usein block-outiksi. Rakennettaessa 3D-pelikenttää kokonaisuuden näkeminen auttaa artistia huomaamaan ongelmat mittasuhteissa ja modulaaristen osien yhteensopivuudessa ennen kuin varsinaisia peliobjekteja on lähdetty tekemään. Alla olevassa kuviossa (Kuvio 5) esittelen projektin aikana muodostuneen kenttäsuunnitteluprosessin vaiheet.



Kuvio 5. Modulaarinen kenttäsuunnitteluprosessi Trouserheartissa.

Projektin metsä-tyyliseen pelikenttään suunniteltiin 3 erilaista esteenä toimivaa palaa joiden läpi pelaaja ei voi kulkea, sekä 3 uniikkia kenttäpalaa, joilla pelikenttään saadaan visuaalista variaatiota. Kenttiin kuuluu aina myös sisään- ja uloskäyntien eteen sijoitetut palat, joilla ei saa olla pelimekaniikkaan liittyviä ominaisuuksia, kuten esimerkiksi vihollisten luomiseen vaikuttavia trigger- collidereita, sillä pelaajaa ei saa estää sisääntulon tai kentästä poistumisen aikana.



Kuva 15. Pelikentän suunnittelumalli (block-out)

Yllä olevassa kuvassa (Kuva 15) voidaan nähdä 3D-suunnittelumalli jonka avulla pelikentän pelattavuus todettiin toimivaksi. Seuraavaksi otin pelimoottorissa suunnittelumallista kuvakaappauksen, jonka päälle maalasin konseptin metsäkentästä (Kuva 16). Konseptin avulla voidaan helpommin siirtyä tuotantovaiheeseen, sillä konseptointi selkeyttää visuaalista kuvaa ja sen avulla pyritään hakemaan mielenkiintoinen visuaalinen kokonaisuus pelikentästä. Block-outin käyttäminen pohjana tuo myös konseptiin tekniset puitteet ja rajoitteet, jotka auttavat artistia näkemään paremmin pelikentän osat, joiden pelimekaanisia ominaisuuksia ei saa muuttaa, peittää tai sekoittaa visuaalisilla osa-alueilla.

Kun konsepti sekä suunnittelumalli on valmis ja testattu pelimoottorissa, voidaan siirtyä varsinaisten peliobjektien mallintamiseen block-out -elementtien pohjalta.

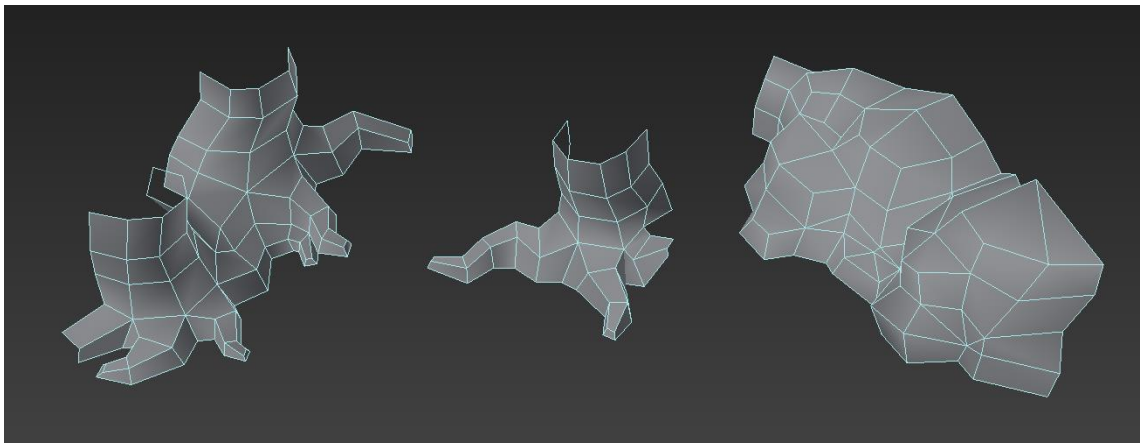
Kuva 16. Pelikentän konseptimaalaus

5.2 Osien mallintaminen

Kuten edellisessä luvussa mainitsin, mallintaminen suunnittelumallien pohjalta on huomattavasti helpompaa kuin tyhjältä pöydältä aloittaminen. Konseptikuvan avulla

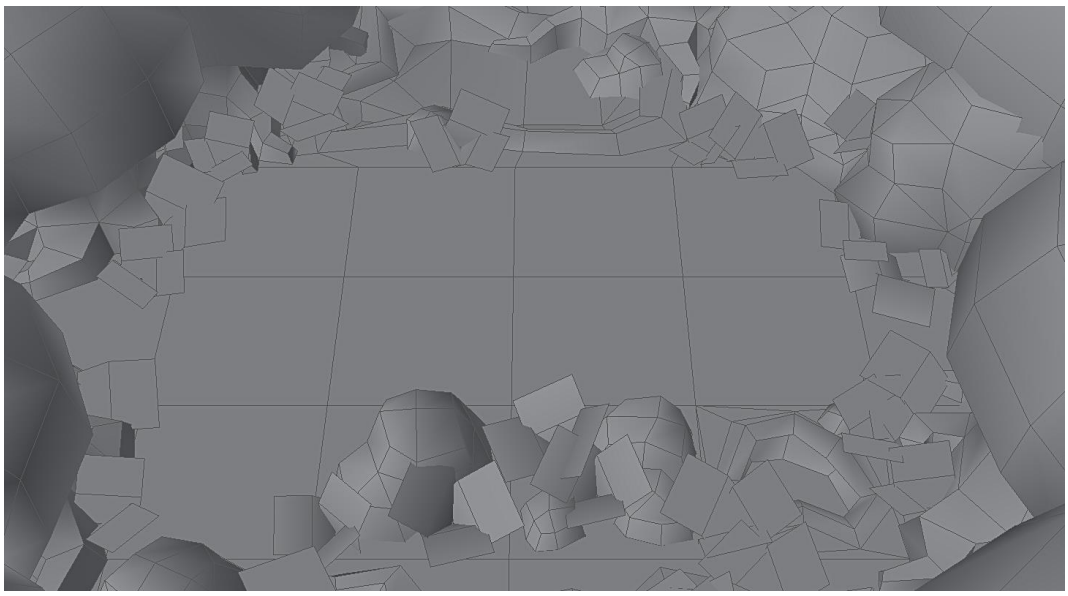
mallinsin kaksi erilaista puuta, kolme aluskasvillisuutena toimivaa mallia sekä muokkasin jonkin verran suunnittelumallin este-osien geometriaa. Muutin yhden kolmesta esteenä toimivista paloista konseptin mukaiseksi kuopaksi, jonka pohjalla on vettä.

Mallintamisessa on tärkeää pitää geometria mahdollisimman optimoituna. 3D-mallien optimoinnissa voidaan käyttää useita tekniikoita, joilla pyritään vähentämään verteksien sekä polygonien määrää. Projektiin asetettiin 6000 polygonin raja. Tämä tarkoittaa sitä, ettei ruudulla saa missään vaiheessa olla yli 12,000 trianglea, kun otetaan huomioon myös ruudulla liikkuvat viholliset, partikkelit sekä pelihahmo. Edellä mainittu luku asetettiin yrityksen edellisen projektin, Rimelands: Hammer of Thorin kehityksen aikana saatujen kokemusten ja käsitysten pohjalta.



Kuva 17. Taustakehikkoon kuuluvia uniikkeja objekteja

Kun pelikentän osat olivat valmiit, siirryin collidereiden mallintamiseen. Unity 3D -pelimoottorissa on yksinkertaiset colliderit sisäänrakennettuna, mutta katoin sopivammaksi tehdä projektiin mukautetut collider-mallit käsin. Collider -meshit toimivat pelissä kahdella tavalla; trigger ja normaali. Trigger -colliderin läpi voi kävellä ja se on täysin huomaamaton pelaajalle. Triggerit havaitsevat kuitenkin kosketuksen muihin collision -objekteihin ja niiden avulla voi koodillisesti kerätä törmäys ja kosketusdataa, jonka avulla voidaan luoda erilaisia pelimekaanisia toimintoja.



Kuva 18. Valmiiksi mallinnetun pelikentän geometriaa.

5.3 Teksturointi

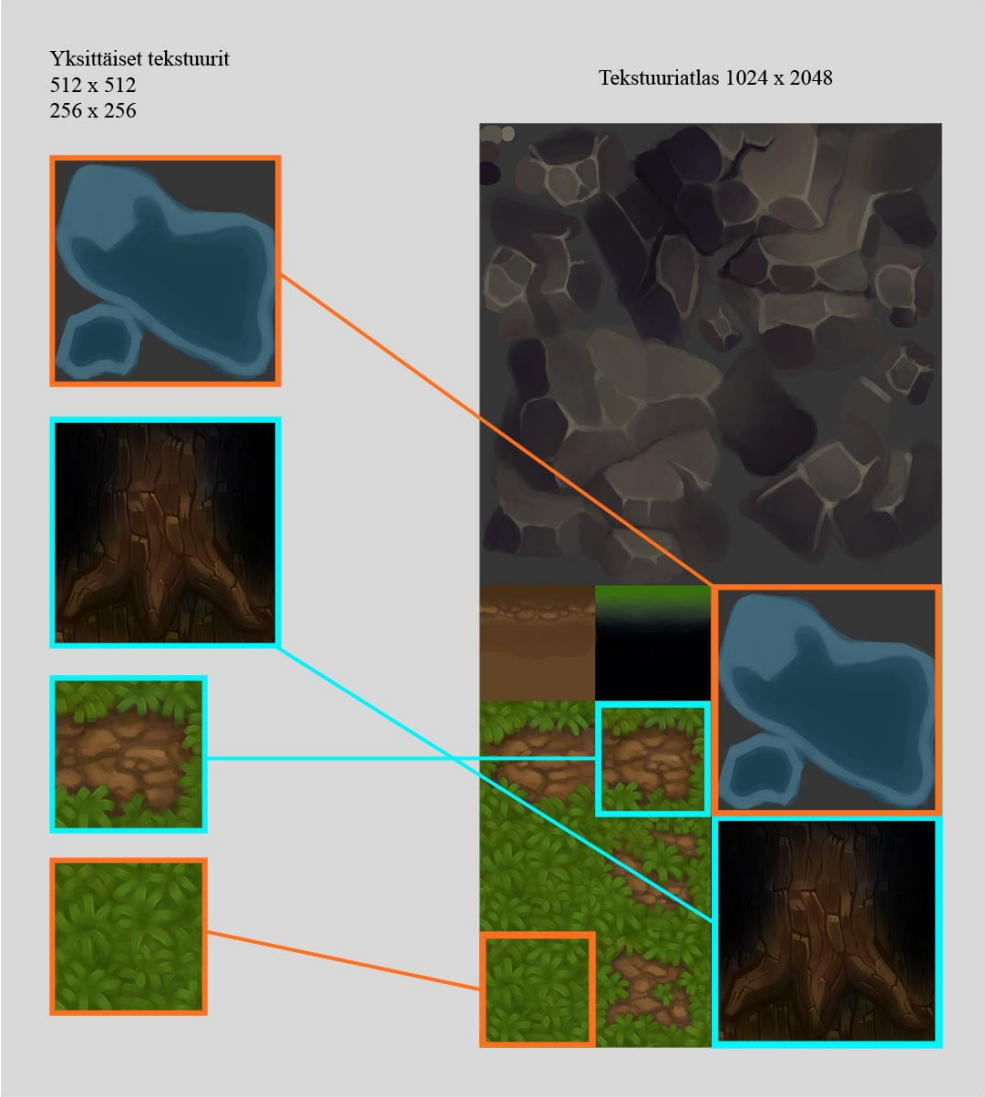
Yksi tärkeimmistä visuaalisista osa-alueista Trouserheartissa on teksturointityyli. Päätimme jo projektin alussa käyttää yksinkertaistettua maalauksellista tyyliä pelin visuaaliikassa. Tämä tarkoitti myös sitä, että tekstuureihin ei käytetty valokuvamateriaalia, joka on muuten erittäin yleistä 3D-pelien tekstuureissa.

Valokuvaa käyttävä tekstuuri

Tyylitelty käsin maalattu tekstuuri

Kuva 19. Esimerkki Trouserheart-pelin tekstuurien visuaalisesta tyylistä oikealla.

Käytin projektin toteuttamisessa kahta erilaista prosessia. Perinteinen teksturointiprosessi toteutetaan mallien sekä UV-karttojen valmistumisen jälkeen, käyttäen UV-karttoja apuna maalauksessa. Toinen tapa toteuttaa teksturointi on käänteinen suhteessa edellä mainittuun tekniikkaan. Käänteinen prosessi aloitetaan luomalla ensin tekstuuriatlas (Kuva 20). Teksturoinnin jälkeen mallinnetaan geometria käyttäen apuna itse atlasta, esimerkiksi mallintamalla geometriaa suoraan tekstuurin päälle, ja korjaamalla mahdolliset virheet UV-kartassa vasta sen jälkeen.



Kuva 20. Pelikentän tekstuuriatlas

Molemmissa tekniikoissa on omat hyötynsä sekä haasteensa. Valitsin teksturointiprosessin sen mukaan, kuinka selkeä kuva minulla oli kyseisestä objektista. Mikäli tiesin jo minkälaisen tekstuurin objekti vaatii ja kuinka geometria pääpiirteittäin tullaan toteuttamaan, käytin käänteistä prosessia objektin luomisessa aloittaen

tekstuurista ja mallintamalla objektin suoraan tekstuurin päälle. Tällä tavoin UV-kartat ovat automaattisesti hyvin lähellä lopullista muotoaan.

Kuva 21. Valmiiksi teksturoitu pelikenttä

5.4 Pelikentän kokoaminen pelimoottorissa

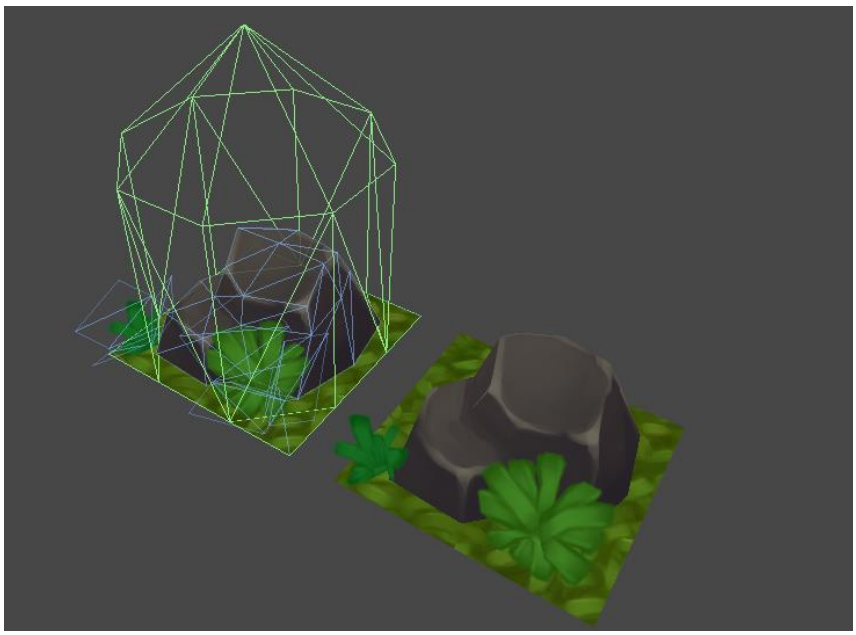
Lopulliset kenttäobjektit siirretään Unity 3D -pelimoottoriin ohjelman tuontiominaisuutta käyttäen. Unityn tuontiasetukset ovat erittäin monipuoliset sekä helposti muokattavissa, ja niiden avulla pystytään myös vaikuttamaan optimointiin, jota käyn läpi seuraavassa työn seuraavassa osassa.

Trouserheartin projektitiedostot on jaoteltu pelimoottoriin selkeiksi kokonaisuuksiksi. Projektitiedostojen sijoittaminen selkeään hakemistorakenteeseen helpottaa tiedostojen myöhempää muokkausta sekä mahdollistaa objektien luomisen pelin ollessa käynnissä. Unity luo jokaiselle tuodulle objektille meta-tiedoston, johon on määritetty objektin aktiiviset asetukset sekä tuontiasetukset.

Kuva 22. Esimerkki käytännöllisestä hakemistorakenteesta Unityssä.

Yllä olevassa kuvassa on esimerkki käytännöllisestä ja selkeästä projektirakenteesta Unityn Editor -näkymässä. Resources -hakemiston alta pystytään helposti tuomaan mm. peliobjekteja kentälle pelin aikana ohjelmoitavien scriptien avulla.

Kun kenttäobjektit on tuotu pelimoottoriin, ne siirretään Scene -näkymään raahaamalla ja erotellaan omiksi Prefab -objekteiksi. Prefab on Unityssä käytetty objekti, jonka aliobjekteiksi voi lisätä mitä tahansa pelimoottorin tukemia komponentteja, kuten scriptejä, ääniä ja materiaaleja. Unityn prefabien alla voi olla lähes loputon määrä muita prefabeja ja jokaisella prefabilla on ominaisuuksina rotaatio, koko sekä koordinaatit, joilla määritetään objektin positio kolmiulotteisessa avaruudessa.



Kuva 23. Mesh collider. Trouserheart.

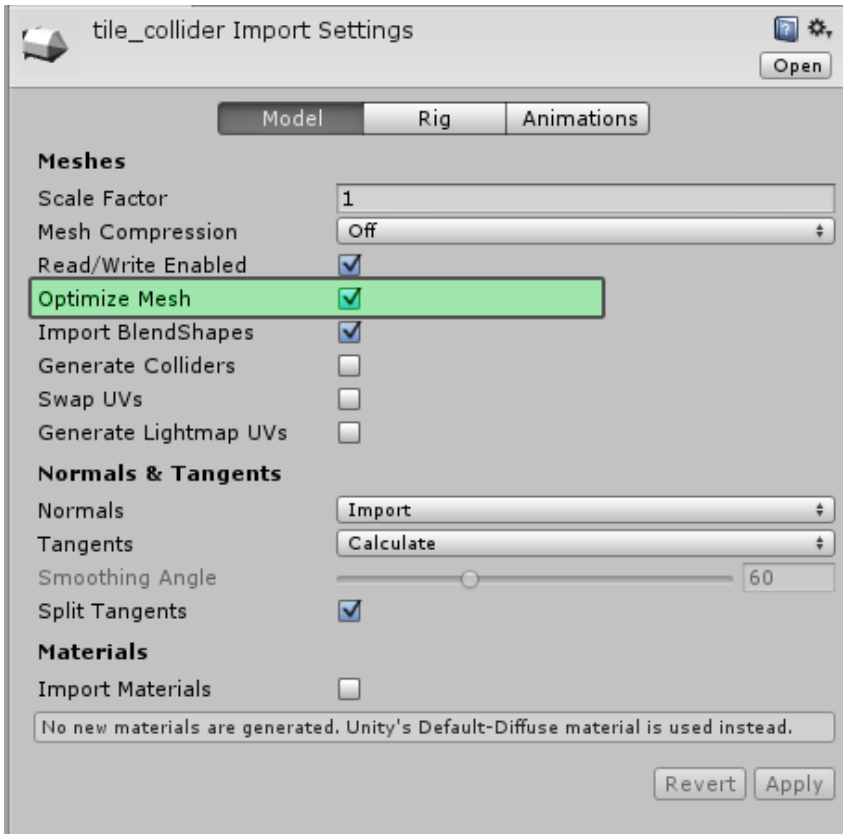
Objekteihin, kuten kentän esteisiin, kiinnitetään lopuksi tarvittavat colliderit. Yllä olevassa kuvassa (Kuva 23) voidaan nähdä käsin mallinnettu mesh collider (vaalea kehikko). Mallinsin myös kentän kehikolle oman colliderin, joka estää pelaajaa liikkumasta ulos pelialueelta.

Tekstuurit tuodaan pelimoottoriin Textures -hakemistoon ja molemmille kentän tekstuuriatlakselle luodaan materiaali, joka pitää sisällään tiedot käytetystä shaderista sekä mahdolliset shaderin asetukset, jotka vaikuttavat tapaan jolla objekti piirretään pelikentälle.

6 OPTIMOINTI PELIMOOTTORISSA

6.1 Geometria

Projektin mallien optimointi pelimoottorissa on suhteellisen rajattua, sillä pelimoottoriin tuotuja 3D-malleja ei voi tässä vaiheessa muuttaa. Import -vaiheessa on kuitenkin mahdollista tehdä joitakin muutoksia, jotka saattavat joissakin tapauksissa parantaa pelin suorituskykyä.



Kuva 24. Unity 3D:n tuontiasetukset.

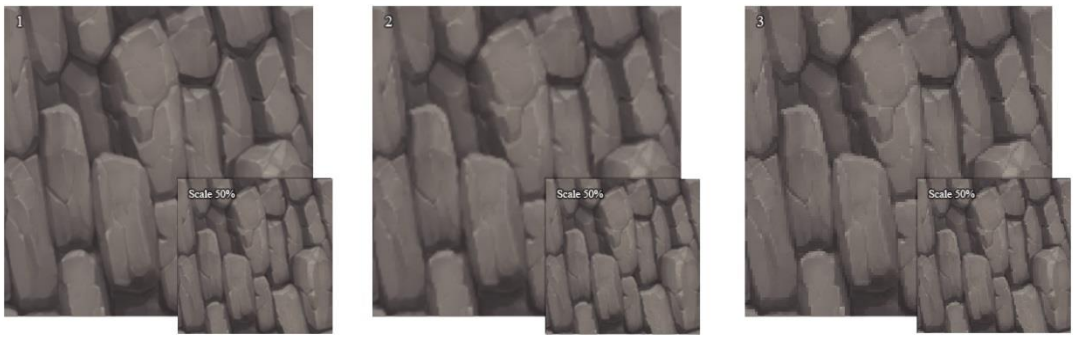
Yllä olevassa kuvassa vihreällä värillä huomioitu asetus järjestää tuontivaiheessa 3D-objektin verteksien tunnistenumeroiden järjestyksen optimaalisemmaksi grafiikkaprosessorille. Mikäli objektilla ei ole animaatioita, on kannattavaa poistaa Rig -valikosta Unityn oletusasetukset. Täten myös animaatioihin liittyvät asetukset poistuvat.

Modulaariset osat täytyy myös muuttaa staattisiksi Unity Editorin Inspector –näkymästä, jotta objektien sarjoitus toimisi. Sarjoitus eli batching laskee geometrian piirtämiseen

käytettävien piirtokäskyjen määrää merkittävästi parantaen pelin suorituskykyä. Unityn Pro -versiossa on myös mahdollista käyttää dynaamisten objektien sarjoitusta.

6.2 Tekstuurit

Tekstuurien optimointi Unityn työkaluilla on erittäin tehokas tapa parantaa suorituskykyä huomattavasti etenkin mobiililaitteille. Tekstuurin tuontiasetuksia muuttamalla voidaan vaikuttaa erityisesti tekstuurien pakkaukseen sekä piirtoon. Tekstuurien pakkaus Unityllä saattaa kuitenkin joissakin tapauksissa vaikuttaa negatiivisesti visuaaliseen laatuun, ja sen käyttö ei aina ole pelin visuaalisen laadun takia kannattavaa.



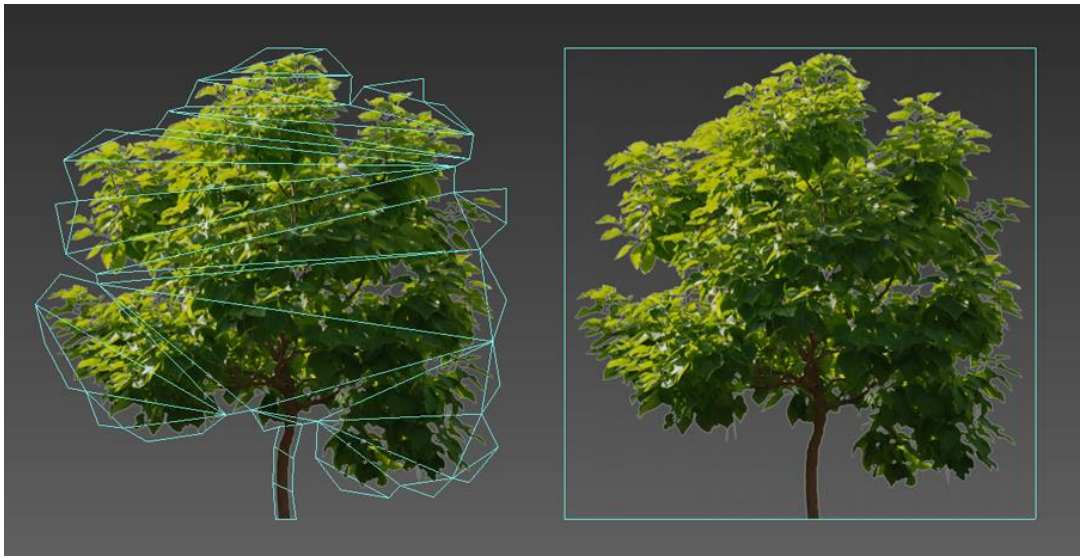
Kuva 25. Eri tuontiasetusten vaikutuksia esimerkkitekstuuriin.

Yllä olevassa kuvassa voidaan nähdä miten eri tuontiasetukset vaikuttavat tekstuurin piirtämiseen pelimoottorissa. Vasemmalla (#1) tekstuurin koko on 256 x 256 pikseliä, tekstuurimuoto RBG 24bit, box mipmap sekä trilinear filtering. Keskellä (#2) tekstuurikoko 128 x 128 pikseliä, muoto RBG24bit, box mipmap sekä trilinear filtering. Oikealla (#3) tekstuurikoko on 128 x 128 RBG24bit, Automatic Compression, no mipmapping, point filtering. Vaikka tekstuurin piirrossa on eroa, tekstuuri #3 on huomattavasti optimoidumpi ja sen käyttö esimerkiksi taustalla olevissa objekteissa on suositeltavaa. Jokaisen tekstuurin ja objektin kohdalla täytyy kuitenkin harkita optimoinnin kannattavuutta. Tärkeimmät objektit, kuten pelihahmo kannattaa useimmiten piirtää mahdollisimman korkealaatuisena.

6.2.1 Alpha

Yksi suurimmista suorituskykyyn liittyvistä haasteista mobiililaitteilla on alpha-kanavan, eli läpinäkyvyyden piirtäminen. Alphan ongelmallisuus on erityisesti suuren resoluution omaavilla laitteilla huomattava, sillä pikseleitä joita GPU:n täytyy prosessoida on paljon, ja alphan läpi piirrettävien pikseleiden data joudutaan laskemaan useaan kertaan jokaisella piirtokäskyllä.

Unityn mobiilioptimointiin liittyvässä artikkelissa kehoitetaan välttämään myös alpha testing -tekniikkaa ja käyttämään sen sijaan alpha blendingiä. Tämä voi tulla joillekin yllätyksenä, sillä pehmennetyllä alpha blendingillä aikaan saatu läpinäkyvyys on yleisen käsitteen mukaan ollut raskaampi tapa piirtää alphaa kuin kovareunainen alpha testing. Alpha testingin hitaus johtuu PowerVR GPU:n rajoitteista kuten HSR:n (Hidden Surface Removal) puutteesta. Useimmat mobiililaitteet, kuten iPhone, iPad sekä Samsung Galaxy -mallistot käyttävät kyseistä GPU:ta.



Kuva 26. Esimerkki alpha-kanavaa käyttävän 3D-objektin optimoinnista.

On tapoja, joilla läpinäkyvyyden piirtoa voi vähentää, kuten leikkaamalla 3D-mallin reunat lähemmäksi alphan reunaa (Kuva 26) ja siten välttämällä alpha-pintojen asettamista etualalle lähemmäksi kameraa. Tätä tekniikkaa olen demonstroinut yllä olevassa kuvassa. Vasemmalla olevan mallin geometria on leikattu seuraten tekstuurin alphaa. Vaikka geometria vasemmanpuoleisessa mallissa on paljon raskaampi, sen

tuomat hyödyt suorituskyvyssä ovat paljon suuremmat kuin käytettäessä oikeanpuolimmaista 2 trianglen meshiä, erityisesti mobiililaitteilla.

Toinen alphan tuoma haaste Unity 3D -pelimoottorissa on alphan reunoihin kohdistuvat visuaaliset artifaktit, kuten kameran taustavärillä piirretyt pikselit, jotka näkyvät objektin siluetin reunoilla. Kyseiset artifaktit johtuvat tavasta, jolla lähes kokonaan läpinäkyvien pikseleiden arvot lasketaan.

Kuva 27. Levitetty bitmap-data oikealla.

Artifaktit saadaan eliminoitua muun muassa tekniikalla, jolla levitetään mallin reunoilla olevaa bitmap -dataa ulommaksi kuten demonstroin yllä. Oikeanpuolimmaisen puun reunavärit on levitetty Photoshopissa käyttäen filter –tekniikoita, kuten radial ja gaussian blur usealla layerilla, samalla pitäen alkuperäisen koskemattoman bitmapin päälimmäisenä. Tekniikkaa käyttäessä täytyy muistaa, että alpha-kanavan dataa ei muuteta.

6.3 Colliderit

Mesh colliderit vaativat prosessorilta ylimääräisiä törmäyksiin liittyviä laskutoimituksia, mutta tehonkäyttö ei kuitenkaan esimerkin mukaisia yksinkertaisia malleja käyttämällä ole merkittävä, ellei mesh collidereita ole montaa pelikentällä.

Kaikista tehokkain ratkaisu on sphere collider, koska sitä käyttämällä kaikki törmäyksiin liittyvät laskennat voidaan silloin laskea vain yhden pisteen ympärille. Trouserheartiin tehtiin kuitenkin käsin useita mesh collidereita, koska ne vaikuttivat pelattavuuteen huomattavasti. Projektissa käytettiin sphere -tyyppisiä collidereita aina kun se oli mahdollista.

7 YHTEENVETO

Olen käynyt työssäni tutkinut mobiilialustojen sekä modulaarisen kenttäsuunnittelun keskeisimpiä ominaisuuksia sekä haasteita, sekä soveltanut tutkimusosassa saatuja tietoja Trouserheart –pelin kehityksessä. Vaikka kyseessä on mobiilipeli, käyttämäni tekniikat eivät rajoitu ainoastaan mobiilipelien kehitykseen. Modulaarisuuden käyttö on erittäin tehokas tapa pelikenttien toteuttamiseen. Samankaltaisia tekniikoita voidaan siis hyödyntää pelien kehittämisessä myös uuden sukupolven PC- tai konsolialustoille.

Kävin käytännön osassa läpi keskeisimmät tekniikat joita käyttämällä kehitin modulaarisuutta hyödyntävän pelikentän sekä esittelin käytännöllisiä optimointitekniikoita. Optimointiin liittyvien käytäntöjen ja tekniikoiden avulla parannetaan pelikokemusta sekä grafiikan laatua huomioiden erityisesti mobiilialustojen suorituskykyyn liittyviä haasteita.

Trouserheart on ollut erittäin opettavainen projekti ja koen kehittyneeni erityisesti 3D-peligrafiikan suunnittelussa. Kokonaisen peliprojektin grafiikan tuotantoprosessit ovat helpompia hahmottaa ja koen että projektin aikana saatu kokemus etenkin modulaarisen grafiikan tuotannosta on vahvistanut osaamistani peligraafikkona. Tulen varmasti käyttämään työn aikana saatuja tietoja seuraavissa projekteissani.

Lopuksi haluan esittää vielä muutaman kuvakaappauksen pelistä. Pyrin esittelemään kuvia, joissa näkyy opinnäytetyössäni kehittämiä modulaarisia objekteja, sekä myös muita samalla prosessilla ja tekniikoilla peliin luotuja kenttiä. Projektin esimerkkinä käytetty forest-kenttä toimi erinomaisena pohjana pelin lähes kymmeneen kenttään, jotka toteutin projektin aikana. Kuvat liitteenä.

LÄHTEET

1. Skooldays.com. Artikkeli: Little Professor calcularor. http://www.skooldays.com/categories/toys/ty1283.htm

2. Giant Bomb. CBS Interactive Inc. Game Boy. http://www.giantbomb.com/game-boy/3045-3/

3. McGraw-Hill. BusinessWeek. 2008. A Brief History of Game Console Warfare: Game Boy. http://images.businessweek.com/ss/06/10/game_consoles/source/7.htm

4. McGraw-Hill. BusinessWeek. 2008. A Brief History of Game Console Warfare: Sega Game Gear . http://images.businessweek.com/ss/06/10/game_consoles/source/8.htm

5. GSMHistory.com. Hagenuk MT-2000 – The world's first mobile providing a game to play. http://www.gsmhistory.com/vintage-mobiles/#hagenuk_mt2000_1994

6. Joel Willans. Artikkeli: Snake Charmed! Nokia Blog. http://conversations.nokia.com/2012/02/02/snake-charmed-10-fascinating-facts-about-the-worlds-most-popular-game/

7. Snow, Blake. GamePro. Artikkeli: The 10 Worst-Selling Handhelds of All Time http://web.archive.org/web/20071012194600/http://gamepro.com/gamepro/domestic/games/features/125748.shtml

8. O'Brien, Terrence. Engadget. Artikkeli. http://www.engadget.com/2013/01/23/apple-announces-q1-2013-earnings/

9. Rivington, James. Artikkeli.TechRadar. http://www.techradar.com/news/gaming/handhelds/nintendo-ds-shoots-past-50-million-mark-161515

10. International Data Corporation, Mobile market analysis, 2013. http://www.idc.com/getdoc.jsp?containerId=prUS23946013#.UTCOPjd4DlY

11. Gartner Inc. Smartphone market analysis 2013. https://www.gartner.com/newsroom/id/2623415

12. Artikkeli. Apple. Q3 2013 conference call. http://appleinsider.com/articles/13/07/23/notes-of-interest-from-apples-q3-2013-conference-call

13. Artikkeli. Epic Games. 2011. https://udn.epicgames.com/Three/rsrc/Three/ModularLevelDesign/ModularLevelDesign.pdf

14. Artikkeli. Will Freeman. Develop. http://www.develop-online.net/tools-and-tech/unity-focus-bad-piggies/0117527

8 LIITTEET