

Jussi Ketonen

Pelikentän luominen Unreal Development Kit -editorilla

Opinnäytetyö

Kevät 2012

Tekniikan yksikkö

Tietojenkäsittelyn koulutusohjelma



SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Tietojenkäsittelyn koulutusohjelma

Suuntautumisvaihtoehto: Digitaalisen median suuntautumisvaihtoehto

Tekijä: Jussi Ketonen

Työn nimi: Pelikentän luominen Unreal Development Kit -editorilla

Ohjaaja: Erkki Koponen

Vuosi: 2012

Sivumäärä: 50

Liitteiden lukumäärä: 2

Tietokonepelejä on tehty jo useita kymmeniä vuosia. Ensimmäiset pelit olivat hyvin rajoittuneita ominaisuuksiltaan, koska tietokoneiden laskentateho oli hyvin pieni nykyisiin tietokoneisiin verrattuna. Tämä rajoitti myös pelien genrejä. Laskentatehon kasvaessa myös pelimoottorit ja kenttäeditorit yleistyivät.

Opinnäytetyön tarkoituksena oli luoda pelikenttä Unreal Development Kit -editorilla, sekä tutustua kenttäsuunnittelun perusteisiin.

Työssä luotiin pelikenttä sekä oma 3D-malli käytettäväksi kentässä. 3D-mallin luonti tapahtui Blender-ohjelmalla ja lopullinen kenttä koostettiin Unreal Development Kit-editorissa. Blenderissä luotua 3D-mallia käytettiin esimerkkinä itsetehdystä sisällöstä kentässä, ja loput kentän sisällöstä otettiin Unreal Development Kit -editorin mukana tulevista esimerkkimateriaaleista.

Työn tavoitteisiin kuului pääasiassa toimivan kentän luonti. Tarkoituksena oli saada myös moninpeli toimimaan, jotta kenttää olisi voitu pelata tarkoitukseensa luodulla tavalla. Moninpeliä ei saatu toimimaan, mutta itse kenttä toimii kuten oli tarkoituksena.

Lopputuloksena oli kohtuullinen. Kentän luonti avasi työn tekijän tietoisuutta peliteollisuuden monimuotoisuudesta käytännön näkökulmasta. Valmis kenttä pakattiin muille jaettavaan muotoon.

Avainsanat: tietokonepelit, unreal development kit, kenttäsuunnittelu, open source

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Business Information Technology

Specialisation: Digital media Production

Author/s: Jussi Ketonen

Title of thesis: Creating a game map with Unreal Development Kit -editor

Supervisor(s): Erkki Koponen

Year: 2012

Number of pages: 50

Number of appendices: 2

The goal of this thesis was to create a game map with Unreal Development Kit -editor, and to familiarize with the basics of game map design.

A custom 3D-model was created to be used in the map. The 3D-model was created in Blender and the final map was put together in Unreal Development Kit. The custom 3D-model was used as an example of self made content in the map, and the rest of the map's content were taken from the example content provided with the Unreal Development Kit -editor software.

The main goal was to create a working map. Another goal was also to include a multiplayer mode, so that the map could be played as it was intended. The multiplayer mode could not be made to work, but the map itself works as intended.

The final result was good, even though the multiplayer was not included. The creation process of the map broadened the creator's knowledge about game industry from the viewpoint of the practical creation process. The finished map was packed in a distributable format.

Keywords: computer games, unreal development kit, game map design, open source

SISÄLTÖ

Opinnäytetyön tiivistelmä	2
Thesis abstract	3
SISÄLTÖ	4
Kuvio- ja taulukkoluetelo	6
Käytetyt termit ja lyhenteet	7
1 JOHDANTO	9
1.1 Aihealueesta	9
1.2 Tavoitteet	9
1.3 Menetelmät	10
1.4 Raportin rakenne	11
2 MÄÄRITTELYT JA RAJAUKSET	12
2.1 Määrittelyjä.....	12
2.2 Rajauksia	14
3 VÄLINEET PELIKENTÄN LUONNISSA.....	15
3.1 Unreal Development Kit (UDK).....	15
3.2 3D-Grafiikka, Blender	16
3.3 Digitaalinen äänituotanto, Audacity	17
3.4 Kuvankäsittely –Gimp	18
3.5 Toteutustapoja	18
4 PELIKENTÄN SUUNNITTELU	20
4.1 Kentän esivisualisointi	20
4.2 Kentän yleishahmotelma	20
4.3 Pelikentän sisällön suunnittelu.....	22
4.4 Oman pelikentän esivisualisointi, yleishahmotelma ja sisältö	23
5 PELIKENTÄN TOTEUTUS	26
5.1 Oman sisällön luonti.....	26
5.2 Pelikentän koostaminen UDK-editorissa	31
5.2.1 UDK:n toimintatapa	32
5.2.2 Oman pelikentän koostaminen UDK:lla.....	36

5.2.3 Oman kentän toiminta	41
5.3 Pelikentän testaus.....	43
5.4 Valmis pelattavissa oleva pelikenttä	44
6 JOHTOPÄÄTÖKSET	47
LÄHTEET	49
LIITTEET	50

Kuvio- ja taulukkoluetelo

Kuvio 1. Kentän alustava suunnitelma	24
Kuvio 2. Jätänkynntilän boolean operaatio	26
Kuvio 3. Blender Exportointi	27
Kuvio 4. UDK 3D-mallin tuonti	28
Kuvio 5. UDK, 3D-malli tuotu	29
Kuvio 6. Tekstuurin liittäminen materiaaliin UDK-editorissa	30
Kuvio 7. Materiaalin liittäminen	31
Kuvio 8. UDK-perusnäkyvä	32
Kuvio 9. Ylin Tool Bar	33
Kuvio 10. Sivupalkki	34
Kuvio 11. Perusgeometria ulkopuolelta tarkasteltuna	37
Kuvio 12. Perusgeometria sisäpuolelta tarkasteltuna	37
Kuvio 13. Sisällön lisääminen kenttään	38
Kuvio 14. Valojen lisääminen kenttään	39
Kuvio 15. Äänien lisääminen kenttään	40
Kuvio 16. Koristeita	41
Kuvio 17. Kenttä ylhäältäpäin esitettynä	42
Kuvio 18. UDK-editorin Unreal Frontend	45
Kuvio 19. Pakattavan pelikentän nimi	46

Käytetyt termit ja lyhenteet

Audacity	Open source -ohjelmisto äänien muokkaamiseen.
Blender	Open source -ohjelmisto 3D-grafiikan luontiin.
Cinema 4D	3D-grafiikan luontiin tarkoitettu sovellus.
CryEngine	Massiivisiin pelikenttiin soveltuva pelimoottori.
Deathmatch	Tietyn tyylinen pelikenttä, jossa on tarkoituksena sotia kaikki kaikkia vastaan.
FPS	First Person Shooter (ensimmäisen persoonan ammunta) on peligenre, jossa pelinäkymää katsellaan pelaajan hahmon silmien takaa.
GIMP	GIMP on lyhenne GNU Image Manipulation Programista. Se on ilmainen avoimen lähdekoodin kuvankäsittelyohjelma.
Indie-peliteollisuus	Omakustanteinen peliteollisuus. Yleensä pieniä pelialan yrityksiä jotka eivät käytä suuria julkaisijoita peliensä julkaisuun.
Kismet	Noodipohjainen ohjelmointiympäristö Unreal Development Kit -editorissa.
Leveli	Pelin yksittäinen kenttä. Käytetään usein selkeästi eroteltavissa olevista pelikentistä.
Mappi	Yksi pelin monista kentistä. Voi tarkoittaa koko pelin yhtä kenttää, tai jos pelissa on vain yksi iso kenttä, tarkoittaa se sitä.

MMORPG	Massively Multiplayer Online Role Playing Game (massiiviroolimoninpeli) on peligenre, jossa tuhannet pelaajat pelaavat roolipeliä internetin välityksellä.
Noodi (Node)	Kismet-ohjelmointiympäristössä tietyn funktion omaava osa.
Pelikenttä	Pelimaailman yksi osa, voi olla myös yksi iso kokonaisuus.
Renderöijä	Pelimoottorin tai muun grafiikkaa käsittelevän ohjelman osa, joka piirtää kuvan näytölle.
RPG	Role Playing Game (roolipeli) on peligenre, jossa pelaaja omaksuu pelin tarinan hahmon roolin ja elää tarinaa tässä roolissa.
RTS	Real Time Strategy (reaaliaikainen strategia) on peligenre, jossa pelinäköymä kuvataan ylhäältä päin taktisesta näköymästä.
UDK	Unreal Development Kit. Pelien ja pelikenttien luomista varten luotu ohjelmisto. Käyttää pelimoottorina Unreal Engine 3:a.
Unreal Engine 3	Unreal Development Kit -editorin hyödyntämä pelimoottori.
UnrealScript	Unreal Development Kit -editorissa käytetty skriptikieli.
3D Studio Max	3D-grafiikan luontiin tarkoitettu sovellus.

(Webopedia. [viitattu 20.3.2012]; UDK Documentation. [Viitattu 20.3.2012]; Maxon. [Viitattu 20.3.2012]; Autodesk. [Viitattu 20.3.2012]; GIMP. [Viitattu 20.3.2012].)

1 JOHDANTO

1.1 Aihealueesta

Pelikenttien sekä muiden virtuaalisten ympäristöjen luonnin tärkeys on kasvanut tietotekniikan ja laskentatehon kehittyessä nopeaa vauhtia. Peliteollisuus on kasvava ala, ja tästä johtuen tarvitaan enemmän ammattimaisia sisällöntuottajia tuottamaan yhä laajempia pelejä.

Peliteollisuuden kasvaessa myös Indie-peliteollisuus, joka tarkoittaa omakustanteista peliteollisuutta, on tullut yhä suosittumaksi ajan myötä. Tämän on mahdollistanut uusien kehitystyökalujen syntyminen. Tässä työssä käytettävä Unreal Development Kit (UDK) on hyvä esimerkki tällaisista työkaluista. Epic Games -pelistudion kehittämä, ennen vain kalliin lisenssin maksaville saatavilla oleva pelimoottori, on jo jonkin aikaa ollut saatavilla myös ilmaiseksi. Tämä tarkoittaa sitä, että monien yksittäisten henkilöiden tai pienempien Indie-pelistudioiden ei tarvitse kirjoittaa omaa pelimoottoriaan alusta, vaan he voivat hyödyntää valmista moottoria halutessaan. Tämä säästää pelistudioilta huomattavasti aikaa ja rahaa.

Tämä työ on kokonaisuudessaan hyödyllinen tietopaketti niille, joita kyseinen ala kiinnostaa, joko ammattimielessä tai vain harrastepohjalta.

1.2 Tavoitteet

Tämän työn tavoitteena on perehtyä siihen, miten UDK-ohjelmistolla sekä muiden digitaalisen sisällöntuotannon ohjelmilla tuotetaan pelikenttään vaadittavaa sisältöä ja miten kaiken sisällön koostaminen tapahtuu Unreal Development Kit editorilla. Päättyöväline UDK-editorin lisäksi on 3D-grafiikan luontiin käytettävä open source -ohjelmisto Blender. Ominaisuuksiltaan ja käyttötarkoituksiltaan käydään läpi myös digitaalisen äänentuotannon open source -ohjelmisto Audacity sekä kuvankäsittelyohjelmisto GIMP (GNU Image Manipulation Program, myös open source). Lopullinen koostaminen tapahtuu Unreal Development Kitillä, joka

on ei-kaupallisessa käytössä oleva ilmainen pelimoottori. Työ keskittyy eniten siihen, mitä kentän tekemisessä tulee ottaa huomioon suunnitteluvaiheesta toteutuksen kautta lopullisen kentän testaukseen asti. Työssä ei käydä täydellisen tarkasti jokaista teknistä yksityiskohtaa läpi, vaan työssä luodaan yleinen ohjenuora ja kuvaus pelikentän suunnittelusta ja tekemisestä. Työn tarkoituksena on antaa asiasta kiinnostuneille hyvä yleiskatsaus siihen, mitä pelikentän luonti oikeasti on ja mitä tällaisessa projektissa tulee ottaa huomioon, sekä antaa hiukan ohjelmistokohtaisia ohjeita toteutuksen avuksi. Yleinen teoria, jota työssä kerrotaan ja kuvataan, on avuksi, jos joku haluaa käyttää eri ohjelmistoja oman kenttensä luomisessa.

1.3 Menetelmät

Työssä rakennetaan pelattava pelikenttä käyttäen open source -ohjelmistoja sekä Unreal Development Kit -editoria. Ensiksi kenttä suunnitellaan paperilla ja tietokonetta hyväksikäyttäen. Suunnitellessa otetaan huomioon kaikki mahdollinen sisältö, jota lopulliseen kenttään tulee. Suurin osa toteutustyöstä tapahtuu tietokoneella monien eri ohjelmistojen avulla. UDK:n toimintatapa kuvataan suhteellisen tarkasti työkalupalkkien tasolla sekä osin toiminnan tasolla, koska UDK:n toiminnan ymmärtäminen on edellytys pelikentän toteutuksen ymmärtämisessä.

Työn valmistuttua tarkastellaan miten työssä ollaan onnistuttu. Asiat, joita arvostellaan työn valmistuttua ovat: toimiiko pelikenttä tarkoitetulla tavalla? Lähteekö se käyntiin erilaisilla tietokoneilla? Onko pelikentän tunnelma sellainen kuin on ollut tarkoituksena? Toimivatko pelikentän sisäiset toiminnot kuten on tarkoituksena ollut?

1.4 Raportin rakenne

Toisessa luvussa käsitellään työn määrittelyä ja rajauksia. Tähän sisältyy määrittelyt käytetyistä ohjelmistoista ja käytetystä ”pelislangista”. Rajauksissa kerrotaan, miten työ on rajattu, mitä ohjelmistoja työn toteutukseen on käytetty ja mikä on lähestymistapa työhön.

Kolmannessa luvussa esitellään tarkemmin käytettävät ohjelmistot. Ohjelmistot käydään yksitellen läpi, esittäen esimerkkejä ohjelmistojen toiminnasta sekä erilaisista toteutustavoista.

Neljännessä luvussa suunnitellaan itse pelikenttä. Luvussa määritellään kentän yleishahmotelma, johon kuuluu genren valinta, pelaajamäärä, ulkonäkö, tunnelma sekä kentän tarina ja tempo. Luvussa suunnitellaan myös tarvittavan pelikentän sisältö, mitä tehdään itse ja mitä pelimoottorin valmiita sisältöjä käytetään.

Viides luku käsittelee toteutusvaihetta. Luvussa käydään läpi oman sisällön luonti, johon kuuluu 3D-grafiikka, äänet, itse kenttä ja valmiin kentän testaus, mahdollisten korjausten tekeminen sekä pelin saattaminen jaettavaan, muille pelattavaan muotoon.

Kuudes luku sisältää työn johtopäätökset. Luvussa pohditaan keskeisiä tuloksia ja arvioidaan niitä kriittisesti sekä objektiivisesti. Lisäksi pohditaan ongelmakohtia, missä onnistuttiin, missä epäonnistuttiin, kuinka jokin asia olisi voitu tehdä paremmin sekä kuinka välttää samat sudenkuopat tulevissa töissä.

Toteutettu pelikenttä on DVD-tallenteena liitteenä ja saatavissa SeAmk Tekniikan yksikössä.

2 MÄÄRITTELYT JA RAJAUKSET

Seuraavissa alaluvuissa keskitytään työn tarkempaan määrittelyyn sekä rajaukseen. Määrittelyissä keskitytään määrittelemään mitä pelit ovat, millaisia ovat eri genret, sekä mitä tarkoitetaan pelikentällä. Rajauksissa tarkastellaan, millaisia eri tapoja on luoda pelejä ja pelikenttiä, sekä rajataan työssä tehtävät asiat.

2.1 Määrittelyjä

Peli voi tarkoittaa hyvin monenlaista erilaista asiaa. Yksinkertaisimmillaan peliksi voidaan kutsua lasten leikkiä, esimerkiksi ruutuhyppelyä. Ruutuhyppelyssä on tietyt säännöt, joita noudatetaan sekä lisäksi säännöt joiden mukaan suoritusta arvioidaan. Kaikille peleille on yhteistä säännöt. Seuraava monimutkaisempi pelimuoto on Pen&Paper eli kynä ja paperi -pelit. Näissä peleissä on yleensä monimutkaisemmat säännöt, jonka vuoksi mukaan on otettu peliä helpottamaan kynä ja paperi, joiden avulla pelin kulkua voidaan tallentaa ja helpottaa mahdollisia laskutoimituksia joita peli vaatii. Usein kynä ja paperi -peleissä on myös pelilauta mukana. Esimerkkipelinä voidaan pitää vaikkapa suosittua Monopoly-peliä.

Monopoly-lauta sisältää jokaiselle pelaajalle omat pelimerkinsä, sekä yhteisiä pelimerkkejä kuten talot ja hotellit. Pelilauta on Monopolyn ”pelikenttä”, joka toimii visuaalisena ”alustana” pelin kululle. Monopolyyn ja moniin lautapeleihin kuuluu myös nopat, jotka ovat pelin eteenpäinviennissä tärkeitä. Nopat sattumanvaraistavat peliä, mikä tekee pelistä arvaamattomamman sekä mielenkiintoisemman.

Seuraava taso pelaamiselle on erilaiset elektroniset pelit, kuten tietokoneet sekä pelikonsolit. Elektronisessa pelaamisessa pelilaitte pitää huolen pelin sääntöjen noudattamisesta sekä laskee laskutoimitukset joita pelin pelaaminen vaatii. Elektroniset pelit ovatkin kehittyneet vanhoista kynä ja paperi -peleistä. Nykyaikaiset pelit eivät juurikaan vaadi pelaajalta mitään muuta kuin esimerkiksi pelihahmon ohjaamista.

Pelejä on paljon ja niitä tehdään koko ajan lisää. Tämän vuoksi eri pelejä jaotellaan selkeyden vuoksi niiden sisällön pohjalta eri peli-genreihin. Peligenrejä ovat mm. FPS-pelit (first person shooter) eli ammutapelit, kuten Battlefield-pelisarjan pelit, RTS-pelit (real time strategy) eli strategiapelit. Tästä esimerkkinä sähköiseksi urheilulajiksi mainittu StarCraft 2 ja sen edeltäjät, RPG-pelit (role playing game) eli roolipelit, joissa pelaaja omaksuu jonkin pelimaailman hahmon roolin ja etenee tarinan mukaan.

Kaikki peligenret voivat myös saada lisäksi erilaisen jaottelun sen pelimuodon ja genren mukaan, esimerkiksi scifiroolipeli Star Wars The Old Republic on MMORPG (massively multiplayer online roleplaying game) tarkoittaen sitä, että peliä pelaa samalla palvelimella yhtäaikaan satoja ihmisiä. Tässä tapauksessa scifiroolipeli on pelin genre ja MMORPG on pelityyppi. Pelejä jaotellaan myös genren omaisesti suoraan sen sisällön mukaan, esimerkiksi autopelit, lentokonepelit, urheilupelit sekä lastenpelit. Simulaatiopelit ovat oma genrensä, joissa sisältö voi vaihdella laidasta laitaan, mutta yhteistä kaikille simulaatiopeleille on se, että niissä pyritään jäljittelemään erilaisia oikean maailman tilanteita mahdollisimman realistisesti (Stahl 2005.)

Pelikenttä käsitteenä tarkoittaa yleensä yksittäistä kenttää pelissä. Tämä käsite on ajanmyötä muuttunut uusien pelimoottoreiden ja tietokoneen sekä konsolien laskentatehon myötä. Kymmenen vuotta sitten kun, tietokoneet ja konsolit olivat tehottomampia, jouduttiin iso pelimaailma jakamaan useaan pieneen osaan, pelikenttiin. Pelikenttä sisälsi pienen osan koko pelin maailmasta, sillä vanhat pelimoottorit ja tietokoneet ja konsolit eivät olisi pystyneet ajamaan reaaliajassa yhtä yhtenäistä kenttää kerrallaan.

Nykyään on toisin, vaikkakin samoista syistä pelimaailmoja jaetaan silti pieneen osiin. Esimerkiksi CryEngine-pelimoottori lataa kerralla koko pelimaailman, mutta lataa kaiken pienimmän sisällön sen mukaan missä pelaaja liikkuu. Tämä tarkoittaa sitä, että pelaaja näkee ympärillään metsän, puita, ruohikkoa sekä muuta vastaavaa yksistyiskohtaista, mutta jos pelaaja näkee kaukana horisontissa olevan vuoren, ei pelaaja näe sen puita, koska pelaaja on sen verran kaukana, ettei suorituskykyistä pelimoottori anna piirtää kyseisiä puita. Kun pelaaja siirtyy

maailmassa tarpeeksi lähelle kyseistä vuotta, ilmestyvät myös puut ja muut yksityiskohdat pelaajan nähtäväksi.

Tässä työssä käytettävä Unreal Development Kit ajaa pelejä Unreal Engine 3:ssa. Tätä moottoria kehitetään jatkuvasti, ja nykyisellään on jo mahdollista tehdä isoja pelikenttiä, eikä pelikentän tekijällä enää ole pakottavaa tarvetta jakaa koko maailmaa pienenpiin palasiin. Nykyisellään kuitenkin toimitaan usein niin, että kentät jaetaan osiin, koska se edelleen on parempi suorituskyvyn kannalta.

Pelikenttä, *mappi* ja *leveli* ovat osaltaan synonyymejä keskenään. Yleensä levelistä puhuttaessa tarkoitetaan pelikenttien yhteydessä jotain tiettyä kenttää monien joukosta. Esimerkiksi, jos koko peli sisältää 10 pelikenttää, voidaan viitata esimerkiksi leveliin 5, joka tarkoittaa pelikenttien järjestyksessä viidettä yksittäistä kenttää. Mappi (eng. map) voi tarkoittaa yhtä kenttää useammasta. Jos on yksi ainut iso pelikenttä, jossa pelataan, voi se tarkoittaa myös mappia. Pelikenttä ja mappi ovat synonyymejä toisilleen.

2.2 Rajauksia

Tässä työssä luodaan valmiiseen pelimoottoriin vain yksi pelikenttä, jossa pyritään hyödyntämään pelimoottorin valmista moninpeliominaisuutta, sekä muita valmiita osia, kuten aseet ja valmiita esineitä moottorin kirjastosta. Myös esimerkin omaisesti luodaan kenttään omaa 3D-grafiikkaa. Lähesymistapa on harrastepohjainen työ, jossa kuitenkin on aikataulu sekä deadline, jolloin työn on oltava valmis. Tämä rajoittaa pelikentän laajuutta ja laatua, sillä pelikentän tekoa voisi jatkaa loputtomiin tekemällä sitä paremmaksi ja paremmaksi, mutta aikarajoitteen takia kenttä tulee saada ajoissa vähintään tyydyttävään, pelattavaan sekä toimivaan muotoon.

Oman sisällön luonti pelikenttään esitellään yhdellä esimerkillä 3D-mallin tuonnista editoriin. Vaikka kenttään olisi mahdollista luoda kaikki grafiikka ja muu sisältö itse, jätetään se kuitenkin yhteen esimerkkiin aikarajoitusten takia. Unreal Development Kit sisältää paljon valmista 3D-grafiikkaa sekä ääntä, joiden avulla kentästä saadaan hyvin uskottava, vaikka kaikkea ei ole itse luotu.

3 VÄLINEET PELIKENTÄN LUONNISSA

Tämän työn työvälineinä käytetään open source 3D -grafiikan luontiin tarkoitettua Blender-ohjelmistoa, sekä ei-kaupallisessa käytössä ilmaista Unreal Development Kit -editoria, joka sisältää Unreal Engine 3 pelimoottorin ja täydet työkalut kyseisen pelimoottorin muokkaamiseen. Tärkeimpänä syynä open source -ohjelmistojen käyttöön on niiden ilmaisuus. Kaikki mitä työssä tehdään ja luodaan, voidaan tehdä täysin ilmaiseksi, kun käytössä on soveltuva PC. Toinen tärkeää syy käyttää open source -ohjelmistoja on niiden muokattavuus, sekä tästä muokattavuudesta johtuva liitännäisten (pluginien) määrä, josta on hyötyä erityisesti importoinnissa sekä exportoinnissa. Seuraavissa alaluvuissa kerrotaan myös open source -kuvankäsittelyohjelmasta GIMP, sekä myös open -äänieditointiohjelmistosta Audacity. Nämä ovat hyviä työkaluja sisällön tuottamiseen, jos UDK-editorin esimerkkisisällöt eivät riitä omiin tarkoituksiin.

3.1 Unreal Development Kit (UDK)

Unreal Development Kit on pelitalo Epic Gamesin kehittämä editori kehittämälleen Unreal Engine 3:lle. UDK:n renderöijä on monisäikeinen ja tukee 64-bittisiä käyttöjärjestelmiä. UDK sisältää työkalut luupohjaisten animaatioiden ajoon, sisältää oman UnrealScript-skriptikielen, oikeaa maailmaa jäljittelevät fysiikanmallinnukset, realistiset oikeaa maailmaa mukailevat valaistustyökalut, elokuvaohjaajatason animaatioiden ohjaustyökalut, maastonluontityökalut, sisäänrakennetut verkko-ominaisuudet, 3D-äänien tuen, edistyneet partikkelieffektityökalut, keinoälytyökalut, reaaliaikaisten shadereiden tuen, tuen hajautettuun laskentaan useiden verkossa olevien koneiden välillä, työkalut tuhoutuvien ympäristöjen luontiin, Bink Video -koodekin, Speedtree-ohjelmiston kasvien ja puiden luontiin sekä FaceFX-ohjelmiston kasvoanimaatioiden luontiin (Features. 2010.)

Unreal Development Kit on aloittelijaystävällinen. Osatakseen tehdä pelikentän, ei tarvitse tietää mitään ohjelmoinnista. UDK muistuttaa käyttöliittymältään hyvin

paljon 3D-grafiikan luontiin tarkoitettuja ohjelmistoja, kuten esimerkiksi Blenderiä, 3D Studio Maxia sekä Cinema4D:tä. Tämän vuoksi käyttäjälle on hyötyä aiemmasta kokemuksesta vastaavien 3D-grafiikkaohjelmistojen käytöstä.

Ilman aiempaakin kokemusta UDK:ta on nopea ja helppo oppia käyttämään. UDK sisältää aloittelijoille ja ammattilaisille Kismet-työkalun, joka on visuaalinen noodi-pohjainen ohjelmointiympäristö, jota osaa pienen harjoittelun jälkeen käyttää myös ilman minkäänlaista ohjelmointitaitoa. UDK:n materiaalikirjasto sisältää paljon valmista materiaalia käytettäväksi kentässä, kuten esimerkiksi 3D-objekteja, tekstuureita, materiaaleja, ääniä, fysiikkamääritelmiä eri materiaaleille, animoituja objekteja kuten puita ja ruohoa ja jopa valmiita pelihahmoja. Tämän vuoksi pelikentän luojan ei tarvitse välttämättä luoda mitään omaa sisältöä itse luodakseen näyttävän pelikentän.

UDK antaa myös mahdollisuudet muokata valmista sisältöä monella tapaa eri näköiseksi. Tämä antaa enemmän mahdollisuuksia suoraan kentän luontiin ilman, että tarvitsee ensiksi opetella muita ohjelmistoja, kuten 3D-grafiikkaohjelmistoja, äänentuotanto-ohjelmistoja, kuvankäsittelyohjelmistoja tai ohjelmointikieliä. (Features. 2010.)

3.2 3D-Grafiikka, Blender

Blender on avoimen lähdekoodin 3D-grafiikan ja animaatioiden luontiin käytettävä ohjelmisto. Blenderin käyttöliittymä on täysin kustomoitavissa omien tarpeiden ja mieltymysten mukaan, mikä lisää joustavuutta sekä mukavuutta työskentelyyn. Oman käyttöliittymän voi myös tallentaa ja se toimii samoin kaikilla alustoilla (PC, Linux, Mac OS X, FreeBSD). Blenderin mallinnustyökalut tukevat erityyppisiä 3D-objekteja, kuten polygoni meshejä, NURBS-pintoja, bezier- ja B-spline-käyriä, metaballseja sekä vektorifontteja (TrueType, PostScript, OpenType).

Blender sisältää täydet moniresoluutio-veistotyökalut, 2D-bittikarttatuen ja 3D-proseduraalisten siveltimien tuet (Paint, Smooth, Pinch, Inflate, Grab) sekä symmetriatuen. Blenderissä on myös täysi tuki luupohjaisten animaatioiden

luontiin, GI-renderöintituki, muokattavien shadereiden tuki, UV-karttojen luontien tuki, työkalut partikkeliefektien luontiin, kompositiointityökalut, sisäänrakennettu Bullet-fysiikkamoottori sekä sisäänrakennettu pelimoottori. Blender tukee myös monia tiedostomuotoja, mikä helpottaa erityisesti ohjelmien välistä importointia sekä exportointia. Blender on kehittynyt vuosien varrella hyvin paljon, ja sen käyttöliittymä on rakennettu täysin uudelleen, mikä on eduksi uusille käyttäjille ja muiden 3D-grafiikkaohjelmistojen käyttäjille, sillä käyttöliittymä on muutettu hyvin samankaltaiseksi kuin 3D Studio Maxissa ja muissa 3D-grafiikkaohjelmistoissa. Blender on hyvin yhteensopiva Unreal Development Kit -editorin kanssa. Ainoa ongelma on heikosti yhteensopivat animaatioiden exportoinnit UDK-ympäristöön. (Features & Gallery. 2012.)

3.3 Digitaalinen äänituotanto, Audacity

Audacity on vapaan lähdekoodin sovellus äänien editointiin ja äänitykseen Windowsille, Mac OS X:lle, Linuxille sekä muille käyttöjärjestelmille. Audacityä voidaan käyttää liveäänien nauhoitukseen, analogisten äänilähteiden digisoimiseen, digitaalisten äänien muokkaamiseen (leikkaa, kopioi, liitä, mikkaa, äänitteen korkeus ja nopeus) sekä moneen muuhun. Äänitys voidaan tehdä käytännössä mistä tahansa lähteestä, importointi ja exportointi tukee monia formaatteja, kuten WAV, AIFF, AU, OGG Vorbis, MPEG (MP2, MP3), MP3 sekä raw (headerless). Audacity sisältää paljon sisäänrakennettuja efektejä, joilla saadaan helposti muokattua mielenkiintoisia äänitehosteita käytännössä mistä äänistä tahansa. Äänenlaatu tukee 16-, 24-, sekä 32-bittisiä näytteitä, jopa 96 kilohertsin taajuudella. Eri näytteenottotaajuudet miksatessa eivät haittaa, sillä Audacity tekee tarvittavat muunnokset automaattisesti reaaliajassa. Audacity sisältää myös spektrogrammin, jonka avulla käyttäjä voi visualisoida ja analysoida äänitteidensä taajuuksia. (Audacity ominaisuudet. 2012.)

3.4 Kuvankäsittely –Gimp

Gimp on avoimen lähdekoodin kuvankäsittelyohjelma käyttöjärjestelmille Windows, Linux, Mac OS X, OpenSolaris sekä FreeBSD. Gimp on ominaisuuksiltaan hyvin samankaltainen kuin sen kaupallinen vastine Adobe Photoshop, joitain ominaisuuksia lukuunottamatta. Gimpistä on olemassa erilaisia versioita, jotka lähinnä vaikuttavat käyttöliittymän ulkoasuun, mutta jotkin versiot sisältävät erilaisen käyttöliittymän lisäksi myös uusia ominaisuuksia. Tästä on esimerkkinä Gimp Painter, joka sisältää oikeita siveltimiä mukailevat työkalut, kuten esimerkiksi öljyvärit, jotka käyttäytyvät kuin oikeat värit niillä maalatessa. Eri värit sekoittuvat realistisesti keskenään virtuaalisella kankaalla. Kaikki käyttöliittymät ovat muokattavissa mieleisekseen, mikä lisää joustavuutta ja mielekkyyttä työskentelyssä. (GIMP Feature overview. 2011.)

3.5 Toteutustapoja

Edellä esitellyt välineet mahdollistavat monia erilaisia toteutustapoja digitaalisen sisällön luontiin. Jokainen sovellus yksinään riittää pelikentän sisällön luontiin, eli käytännössä UDK sisältää paljon valmista materiaalia, jota voidaan halutessa muokata editorin sisällä niin paljon, että materiaali vaikuttaa täysin uudelta. Blender yksinään sisältää välineet uuden 3D-grafiikan luontiin, ja sillä onnistuu myös 2D-grafiikan luonti ja muokkaus, jota tarvitaan tekstuureiden luontiin kyseisille 3D-malleille.

Audacity on tässä työssä esitellyistä välineistä yksipuolisin, sillä se on yksinomaan äänituotantoon keskittynyt sovellus, joka ei sisällä mitään muita ominaisuuksia joita muut ohjelmat sisältävät.

Gimp sisältää monipuoliset työkalut 2D-grafiikan luontiin ja muokkaamiseen, joten sillä voitaisiin tarvittaessa korvata Blenderin puutteelliset 2D-grafiikkatyökalut. Tämän työn toteutus voitaisiin siis tehdä pelkällä UDK:lla: pelkästään muokkaamalla valmista editorin sisäistä materiaalia omaksi, ja käyttämällä sitä kentässä. Tämän työn toteutustapana on kuitenkin käyttää vain Blenderiä UDK-editorin lisäksi, sillä UDK -editori sisältää paljon valmista ja erittäin laadukasta

materiaalia, että tähän työhön ei tarvitse tehdä muuta materiaalia kuin suunnitelmaan kirjoitettu soihtu/jätkänkynttilä.

4 PELIKENTÄN SUUNNITTELU

Pelikentän suunnittelu on yksi tärkeimpiä osia kenttien luomisessa. Suunnitelmassa luodaan pohja sille, millainen kentän tulisi olla. Seuraavissa alaluvuissa käydään läpi suunnittelun eri vaiheita.

4.1 Kentän esivisualisointi

Ennen kuin kenttää lähdetään toteuttamaan, on hyvä kirjoittaa suunnitelma kentästä, mitä noudatetaan toteutusvaiheessa. Tätä vaihetta kutsutaan joskus nimellä esivisualisointi (previsualization). Luotiin kenttää sitten taistelu-, ampumatai roolipeliin, on aina hyvä olla luotuna selkeä esivisualisointi. Se antaa kentän tekijälle selkeän visuaalisen ohjenuoran, kuinka kentän tulisi rakentua ja miltä sen tulisi näyttää. Esivisualisointia voidaan käyttää ehdottomana ohjeena kentän teossa, mutta siihen voidaan myös tehdä muutoksia ja sitä voidaan jalostaa eteenpäin samalla kun kenttä on toteutusvaiheessa. Useille pelitaloille mahdollisuus tehdä tehokasta esivisualisointia on ylellisyyttä. Esivisualisointiin käytetty aika säästää aikaa vastaavasti toteutuksen puolelta. Kesken toteutusprosessin käytetty aika, joka menee esimerkiksi visuaalisen ilmeen uudelleenmietintään, saattaa pysäyttää toteutuksen. Tällaiset ongelmat voidaan monesti ehkäistä jo esivisualisointivaiheessa, säästäen aikaa ja muita resursseja. (Meigs 2003, 3-4.)

4.2 Kentän yleishahmotelma

Pelikenttien arkkitehtuurin suunnittelussa voi käyttää mielikuvitustaan ja ottaa ”taiteilijan vapauksia”. Tämä tarkoittaa, että pelikenttien suunnitteluun on voitu ottaa inspiraatiota oikean maailman kohteista ja paikoista ja tehdä niistä peliin sopivia. Tämä taas tarkoittaa sitä, että kenttäsuunnittelijan täytyy tasapainoilla graafisen realistisuuden ja suorituskyvyn välillä. Liian realistinen ja muutoin mahtava graafinen kokemus pelaajalle saattaa koitua kohtaloksi pelin suorituskyvyn kannalta. Etenkin PC-alustalle kenttiä luodessa tulee ottaa

huomioon, että monien käyttäjien PC:t eivät ole yhtä tehokkaita. Tästä syystä tulee rakentaa kentät niin, että ne toimivat heikommallakin koneella. Jokaisessa kentässä jota tehdään, on omat suorituskykyhaasteet selvitettävänä. Moninpelikentissä, joita tämän opinnäytetyön kenttä sekä useiden nykyisten pelien kentät tyyliltään edustavat, on otettava huomioon joitain yksinkertaisia seikkoja. Esimerkiksi useiden pelaajien yhtäaikainen navigointi kentässä tulee ottaa tarkkaan harkintaan kentän pohjapiirrustusta suunniteltaessa. (Meigs 2003, 5-6.)

Yleisesti ottaen pullonkaulat ja umpikujat eivät toimi kovin hyvin moninpeliympäristöissä. Kyseistä sääntöä kuitenkin pidetään pehmeänä sääntönä (soft rule), koska sen suhteen on olemassa poikkeuksia. Voidaan esimerkiksi sanoa, että johonkin tiettyyn kenttään pullonkaula voisi sopia oikein hyvin pelilliseksi kiinnekohdaksi. Kova sääntö (hard rule) sen sijaan on sellainen sääntö josta ei ole poikkeuksia. ”Vältä turhauttamasta pelaajaa!” on hyvä nyrkkisääntö suunnittelijalle. Testauksesta tuleva palaute on tärkein yksittäinen mittari siitä, onko kentän suunnittelussa onnistuttu luomaan hyvä ympäristö pelaamiselle. (Meigs 2003, 6-7.)

Esimerkkinä hyvästä kenttäsuunnittelusta voidaan pitää kentän osana olevaa huonetta johon on vain yksi sisäänkäynti. Huoneessa on jotain pelaajalle tärkeää poimittavaa, esimerkiksi aseita ja ammuksia. Kun pelaajat oppivat tämän umpikujan, josta saa tärkeitä resursseja, muut pelaajat alkavat kytäämään pelaajia tämän alueen läheisyydessä, sillä tämä on helppo ottaa päiviltä tällaisessa umpikuja-tilanteessa. Tällaiset tilanteet turhauttavat helposti pelaajia, joten kenttää suunnitellessa tällainen huone muuttuu paremmaksi pelillisesti, kun huoneeseen lisätään useampi sisään/uloskäynti. Useampi sisään/uloskäynti tekee tilanteesta arvaamattomamman, joten pelaajat eivät voi täysin varmasti ennakoida toisen pelaajan liikkeitä kyseisessä pelitilanteessa. Nyrkkisääntönä voidaan pitää, että kenttää suunnitellessa kenttäsuunnittelijan tulisi suunnitella peliä ja pelattavuutta tukeva kenttä. Jokaisen kentän muodon ja ominaisuuden tulisi tukea pelimuotoa itseään ja sillä tulisi olla jokin funktio pelin kannalta. Tämä pätee kaikissa peligenreissä, ei pelkästään moninpeliammunnoissa. Kenttäsuunnittelijan pitää suunnitellessaan olla tietoinen siitä mikä on kohdepelin toiminnan tarkoitus (heartbeat). Esimerkkinä toimii vaikkapa nettiammuntapelit, joissa tarkoituksena

on tappaa muut ja pysyä itse mahdollisimman pitkään hengissä. Tällaiseen peliin kenttää suunnitellessa on aina pidettävä tämä pelin "heartbeat" mielessä, sillä muuten kentän suunnitelmaan voi tulla tarpeettomia asioita mukaan tai sitten puhtaita virheitä jotka rikkovat pelin immersiiivisyyden ja viihteellisyyden. (Meigs 2003, 7-9.)

Suunnitteluvaiheessa tulee ottaa myös huomioon loogisuus, sekä symmetria, joka korostuu erityisesti joukkuepohjaisissa peleissä. Loogisuus on tärkeää pelin uskottavuuden ja pelattavuuden kannalta. Jos esimerkiksi tullaan ulos keskiaikaisesta tyrmästä futuristiseen kaupunkimaisemaan, ei tällöin voida puhua loogisesta kentästä. Toki tähänkin on poikkeuksensa, jos esimerkiksi kyseinen peli käsittelisi aikamatkustuksen saloja, olisi tämä looginen siirtymä pelikentässä. Symmetrisyys on tärkeää myös muunlaisissa peleissä, sillä jos symmetrisyys puuttuu täysin, pelaaja voi tuntea olevansa hukassa. Joukkuepohjaisissa peleissä symmetrisyyden puuttuminen voi rikkoa pelin monella tavalla. Esimerkiksi joukkuepohjaisen lipunkaappauspelin (capture the flag) tulee olla symmetrinen, sillä epäsymmetrisyys saattaa sekoittaa pelaajan suuntavaiston ja antaa toiselle joukkueelle etulyöntiaseman kyseisessä kentässä. Epäsymmetrisyys voi myös tehdä fyysistä haittaa joukkueille, sillä toisella joukkueella voi olla lyhyempi matka juosta tiettyyn paikkaan suhteessa toiseen joukkueeseen. (Meigs 2003, 11.)

4.3 Pelikentän sisällön suunnittelu

Kentän suunnittelu tapahtuu pääosin paperille tehdyillä konseptisuunnitelmilla ja tekstillisillä kuvauksilla siitä, mitä kenttään tulee vähintään luoda. Konseptisuunnitelma on ruutupaperille piirretty ilmakuva kentästä, jossa hahmotellaan kentän perusmuoto ja elementtien sijoitus. Ruutupaperi on hyvä apuväline, sillä UDK:ssa perusmittana on gridien yksiköt. Käytännössä ruutupaperille piirrettäessä voidaan sanoa, että yksi ruutu paperilla on yksi gridiruutu UDK:ssa.

Kenttään luotavan sisällön tulee olla modulaarista, eli samoja elementtejä on hyvä pystyä käyttämään useissa eri tilanteissa ja paikoissa, joko sellaisenaan tai hiukan muokattuna. Tämä pätee 3D-malleihin, materiaaleihin sekä ääniin. 3D-mallien

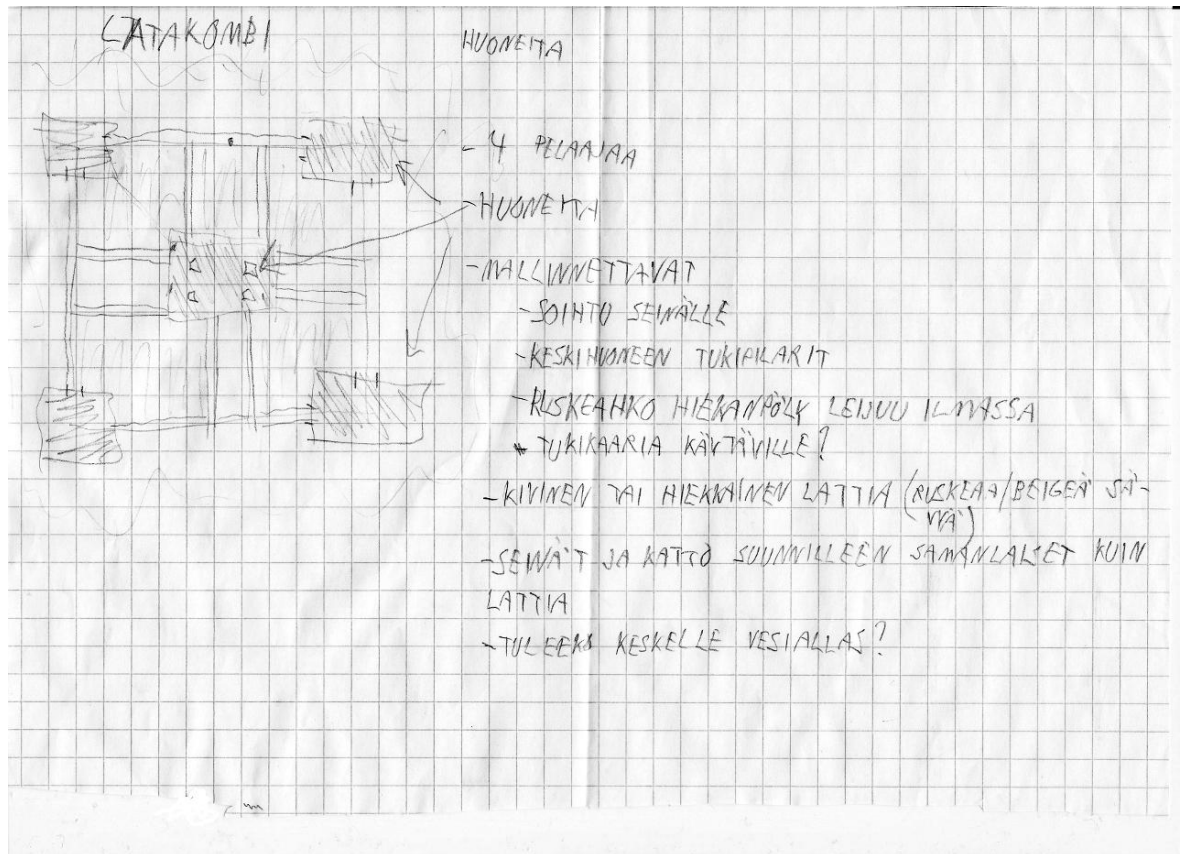
tulee olla mahdollisimman yksinkertaisia topologiaaltaan, sillä se parantaa kentän suorituskykyä, erityisesti jos kentässä erilaisia objekteja on paljon.

4.4 Oman pelikentän esivisualisointi, yleishahmotelma ja sisältö

Tämän työn kenttä tulee olemaan sisätilakenttä, eräänlainen katakombi. Kenttä on hiljainen ja äänillä on tapana kaikua kivisissä tunneleissa. Tarvittavia itsetehtäviä malleja on soihtu/jätkänkynttilä, jota monistetaan katakombin käytäville valaisemaan ympäristöä.

Tämän työn kenttä on deathmatch-pelikenttä, jossa tarkoituksena on tappaa muita pelaajia mahdollisimman tehokkaasti. Kentän tulee siis olla yksinkertainen ja helposti hahmotettava kokonaisuus. Pelaajamäärältään kenttä tulee soveltumaan parhaiten neljän pelaajan pelattavaksi. Kentän tulee soveltua nopeaan pelitempoon, sekä omata samantyyllisen nopeutta korostavan tunnelman. Suuri osa tunnelmasta tulee pelikentän koosta, joka oikein mitoitettuna on juuri sopivan pieni, että pelaajat törmäävät toisiinsa hyvin nopeaan tahtiin, eikä odottelua juuri ole.

Tarinaa kentässä ei juurikaan ole, sillä kenttä tulee olemaan oma yksittäinen kenttensä ilman mitään sidoksia muihin kenttiin. On toki hyödyllistä ja suotavaa keksiä kentälle taustatarina. Sitä on helpompi työstää, jos kenttä on osa isompaa pelikenttäkokonaisuutta.



Kuvio 1. Kentän alustava suunnitelma

Kuvio 1 sisältää ruutupaperille piirretyn karkean suunnitelman pelikentästä. Vasemmalla on pohjapiirustus kentän rakenteesta, ja oikealla on ranskalaisin viivoin lueteltuna kentän tärkeimmät asiat, jotka tulivan suunnitteluvaiheessa mieleen. Kentän tärkeimmiksi ominaisuuksiksi suunniteltiin:

- sisältää huoneita
- 4 pelaajaa
- mallinnettavat:
 - soihtu seinälle
 - keskihuoneen tukipilarit
 - ruskeahko hiekanpöly leijuu ilmassa
 - tukikaaria käytäville

- kivinen tai hiekkainen lattia (ruskeaa/beigeä sävyä)
- seinät ja katto suunnilleen samanlaiset kuin lattia
- Tuleeko keskelle vesiallas?

Nämä ominaisuudet tulivat suunnitteluvaiheessa spontaanisti ideoiden. Hyvä mielikuvitus on hyödyksi, kun suunnitellaan kenttää, sillä lopullinen kentän olemus täytyy olla jollain tasolla valmiiksi mietitty.

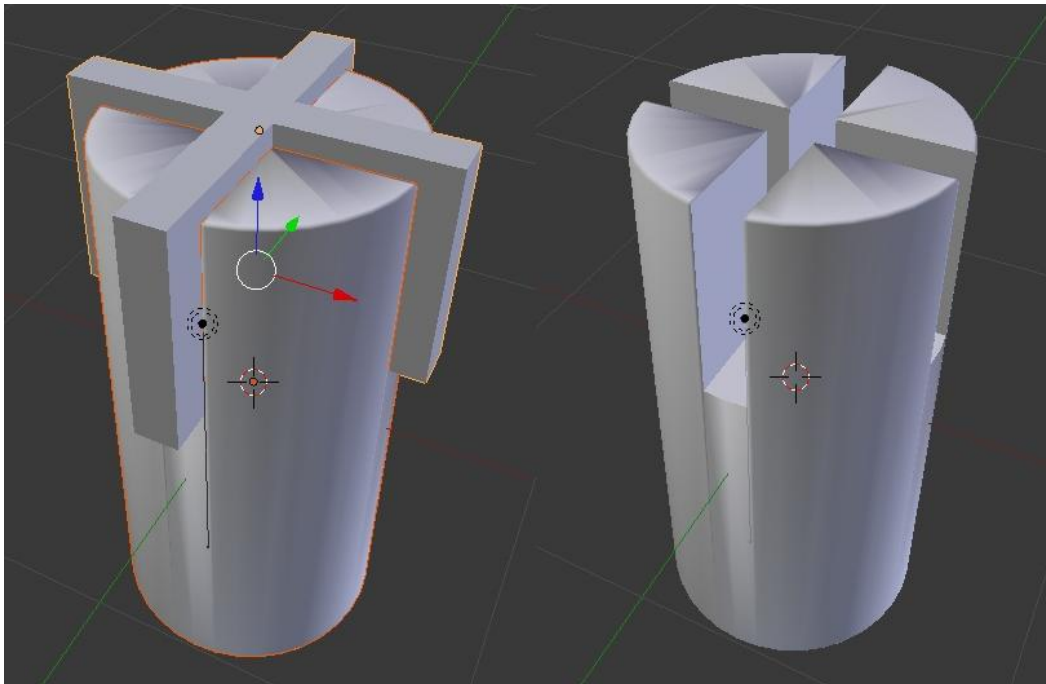
Tämän kaltainen karkea suunnitelma on tämän työn kenttää varten tarpeeksi hyvä, koska ilman aiempaa kenttäsuunnittelukokemusta on hyvin vaikeaa tehdä tarkempia suunnitelmia kentästä. Tämän suunnitelman kriittisen arvioinnin jälkeen päätettiin ryhtyä tekemään itse kenttää.

5 PELIKENTÄN TOTEUTUS

Seuraavissa alaluvuissa käydään pelikentän toteutusta läpi eri ohjelmien näkökulmasta. On tärkeää käydä läpi, kuinka kenttä tehdään ja kuinka siihen luodaan omaa sisältöä ja kuinka se tuodaan lopuksi kenttään. Luvussa 5.1 selitetään yksinkertaisesti 3D-mallin luonti Blenderissä, ja kuinka se viedään Unreal Development Kittiin käytettäväksi. Tarkemmin UDK:n käyttöä ja käyttöliittymää käsitellään luvussa 5.2.1.

5.1 Oman sisällön luonti

Sisällön luonti aloitetaan, kun kentän suunnitelma on saatu tehtyä tarpeeksi hyvin, että sitä voidaan käyttää ohjeena sisällön luontiin. Tämän työn suunnitelmasta kävi ilmi, että ainut itse tehtävä 3D-malli on jätkänkynttilä.

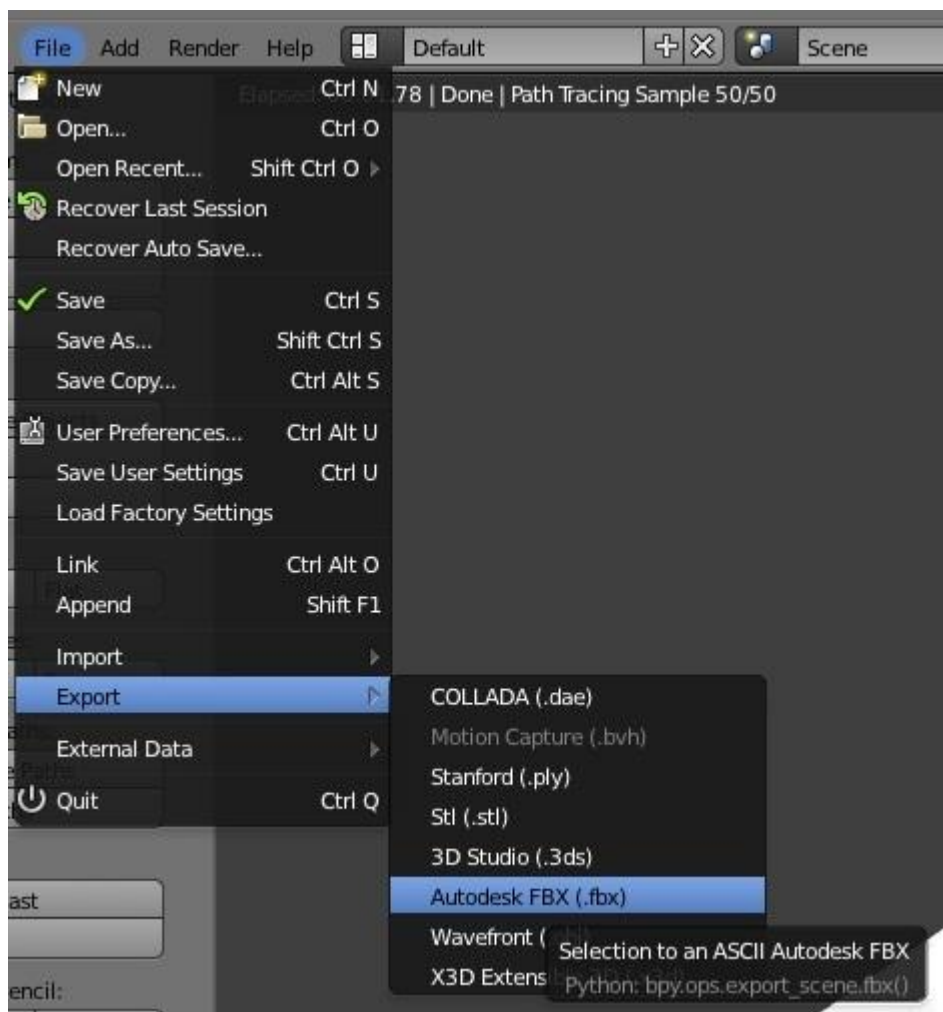


Kuvio 2. Jätkänkynttilän boolean operaatio

Jätkänkynttilä mallinnettiin ja teksturoitiin Blenderissä. Malli tehtiin käyttämällä skaalattua sylinteri-primitiiviä ja tekemällä jätkänkynttilässä oleva ”sahaus” boolean-operaatiota hyväksikäyttäen (kuvio 2). Vasemmalla kuviossa 2 on sylinteri

ennen boolean operaatiota, ja oikealla boolean operaation jälkeen. Boolean operaatio tarkoittaa tässä tapauksessa sitä, että on olemassa kaksi 3D-mallia, ja niiden jakamasta tilasta toinen kappale vähentää oman tilansa toisen mallin tilasta. Eli kuviossa 2 plus-merkin muotoinen malli vähentää itsensä sylinterin muotoisesta kappaleesta.

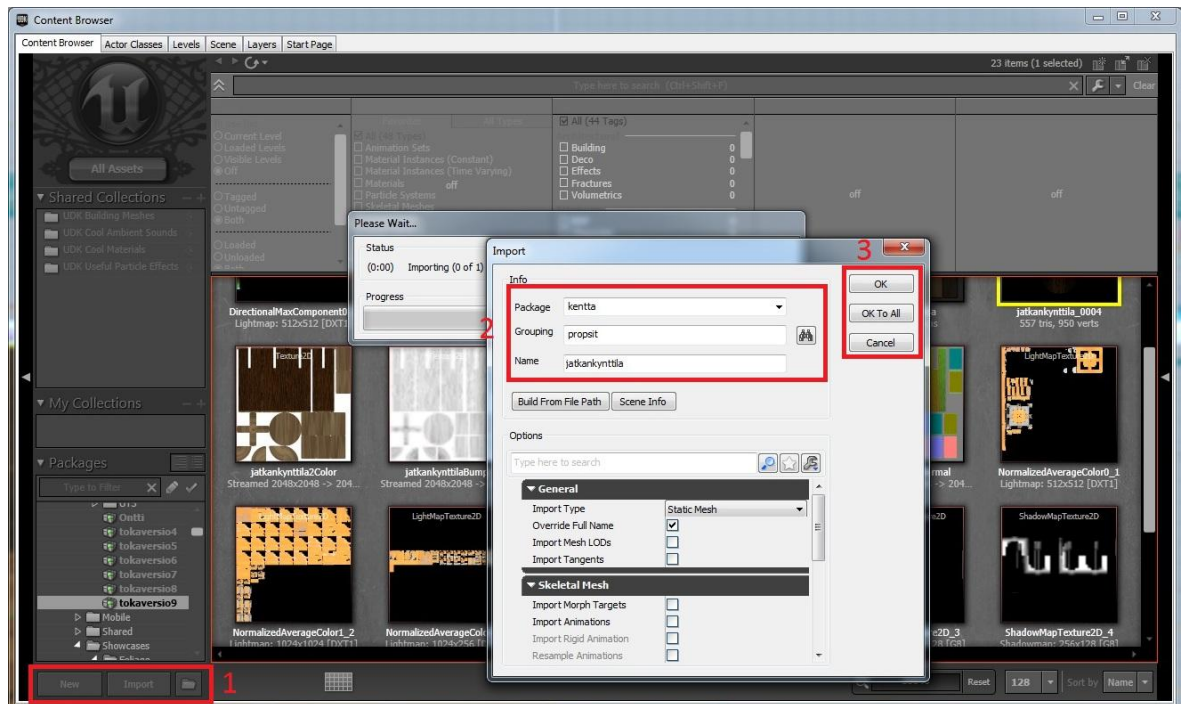
Tekstuureiksi hankittiin internetistä valmiita ilmaisia tekstuureita, joihin kuuluivat puun kaarna, sekä puun syitä kuvaava tekstuuri. Jätäkänkynttilän valmistuttua tulee 3D-malli viedä UDK-editoriin. Kuviossa 3 valitaan Blenderin File-valikosta Export > Autodesk FBX (.fbx), joka on UDK:n ymmärtämä tiedostomuoto 3D-malleille.



Kuvio 3. Blender Exportointi (Features & Gallery. 2012)

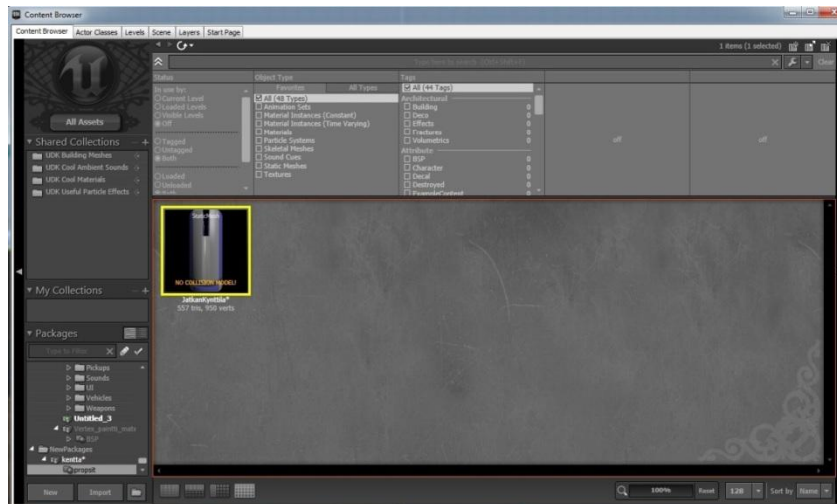
3D-malli tuodaan UDK -editoriin kuvion 4 esittämällä tavalla. UDK -editorissa avataan Content Browser. Content Browserin avauduttua ikkunan vasemmasta

alukulmasta löytyy painike Import (kuvio 4, ruutu 1). Import-painike aukaisee tiedostonselain-ikkunan, jonka avulla etsitään Blenderistä Exportoitu 3D-malli. Tämän jälkeen aukeaa Import-ikkuna, jossa tuotavalle 3D-mallille määritetään, mihin pakettiin, ryhmään ja millä nimellä malli halutaan tuoda (kuvio 4, ruutu 2). OK-painikkeella hyväksytään tämä vaihe ja UDK-editori tuo 3D-mallin määriteltyyn paikkaan (kuvio 4, ruutu 3).



Kuvio 4. UDK 3D-mallin tuonti (UDK Documentation. 2010)

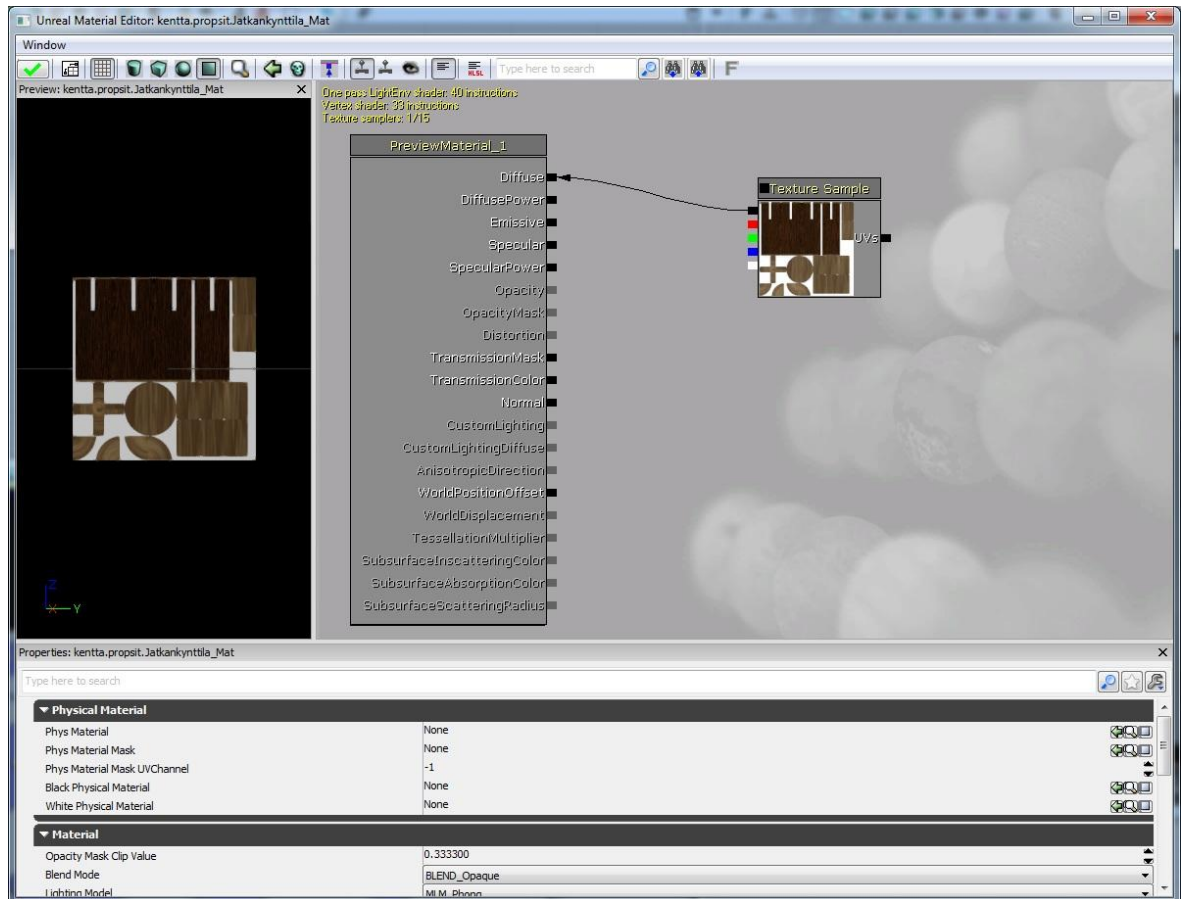
Kun UDK on importoinut 3D-mallin, Content Browser -ikkuna näyttää tuodun mallin ikkunassa kuvion numero 5 mukaisesti.



Kuvio 5. UDK, 3D-malli tuotu

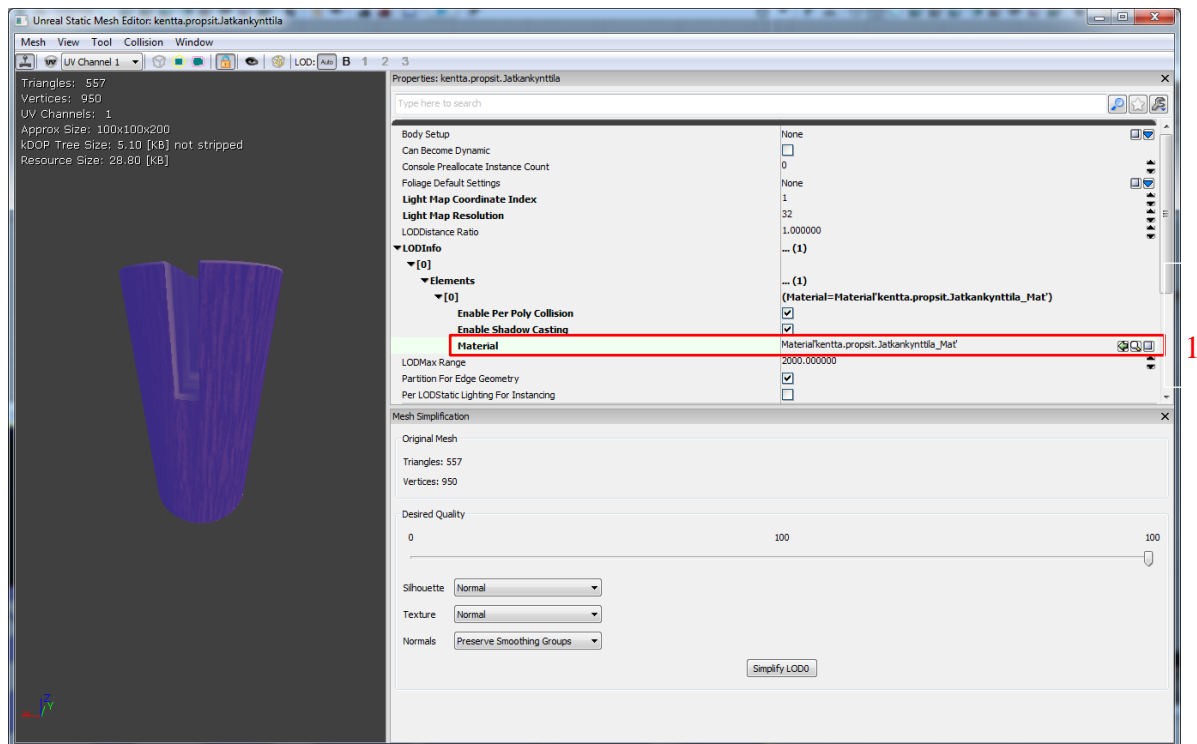
Kun 3D-malli on tuotu, siihen täytyy liittää luodut tekstuurit. Tekstuurit tuodaan samalla tavoin kuin itse malli, import-painiketta käyttämällä. Kun tekstuurit on tuotu UDK-editoriin, näkyvät ne samalla tavoin kuin 3D-malli Content Browserissa. Esikatselukuva näyttää miltä tekstuuri näyttää, mikä helpottaa työskentelyä, jos yhteen malliin kuuluu useampi tekstuuri.

Ennenkuin 3D-malliin voidaan liittää tekstuurit, tulee tekstuurit liittää materiaaliin. Vasta itse materiaali liitetään 3D-malliin. Uusi materiaali luodaan painamalla hiiren oikeaa näppäintä Content Browserissa ja aukeavasta valikosta valitaan New Material. Material editor aukeaa automaattisesti. Content Browserista valitaan aktiiviseksi Jätkäkynttilän tekstuuri, ja seuraavaksi palataan Material Editor -ikkunaan, jossa painetaan tyhjälle alueelle kirjain T+hiiren oikea näppäintä, mikä lisää tekstuuri-noodin sekä liittää kyseiseen noodiin Jätkäkynttilän tekstuuriin, joka oli valittuna aktiiviseksi Content Browserissa. Tämän jälkeen hiiren oikealla näppäimellä yhdistetään tekstuuri-noodi materiaalin Diffuse-kanavaan. Esikatselukuva vasemmalla näyttää tuloksen (kuvio 6). Tämän ikkunan suljettaessa tulee kysymysikkuna, jossa kysytään tallennetaanko muutokset. Tässä vaiheessa halutaan muutosten tallentuvan, eli valitaan vaihtoehto YES.



Kuvio 6. Tekstuurin liittäminen materiaaliin UDK-editorissa (UDK Documentation. 2010)

Seuraavaksi avataan 3D-malli kaksoisnapauttamalla sitä Content Browserissa. Tämä aukaisee Static Mesh -editorin, josta malliin liitetään juuri luotu materiaali. Content browserista valitaan juuri luotu materiaali, jonka jälkeen palataan Static Mesh editor -ikkunaan. Seuraavaksi painetaan vihreää nuolta kohdassa 1 (kuvio 7), joka liittää materiaalin 3D-malliin.



Kuvio 7. Materiaalin liittäminen (UDK Documentation. 2010)

Kyseiselle objektille luodaan myös törmäysgeometria. Tämä tapahtuu helpoiten valitsemalla Static Mesh -editorin ikkunan ylälaudasta Collision>Auto Convex Collision, sekä avautuvasta ikkunasta Apply. Nyt 3D-malli on valmis käytettäväksi kentässä.

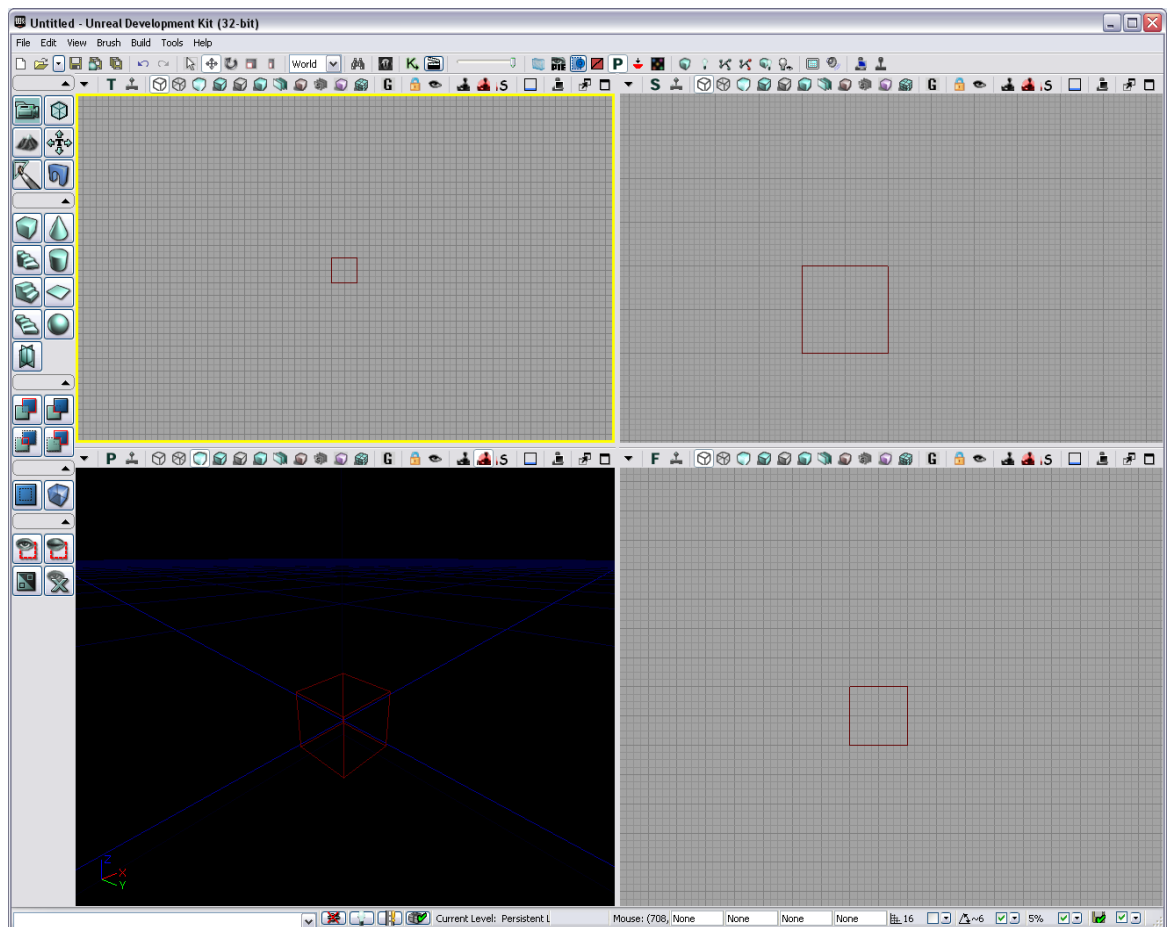
5.2 Pelikentän koostaminen UDK-editorissa

Seuraavissa luvuissa käydään läpi kentän koostaminen vaiheittain. Luvussa 5.2.1 tutustutaan UDK:n toimintatapaan, ja käydään läpi UDK:n peruskäyttöliittymä, sekä näytetään kuinka kenttään lisätään sisältöä. Tämä on tärkeää käydä läpi, jotta lukija pysyy mukana seuraavissa luvuissa. Luvussa 5.2.2 käydään läpi oman kentän koostaminen vaiheittain. Kentän koostamisen tärkeimmistä vaiheista esitellään kuvakaappauksia selventämään työn kulkua. Luvussa 5.2.3 käsitellään kentän toimintaa kuvakaappauksien avulla.

5.2.1 UDK:n toimintatapa

Unreal Development Kit on toimintatavaltaan WYSIWYG (What You See Is What You Get –editori), eli se mitä näet editorin ikkunoissa, on se mitä tulet näkemään kenttää pelattaessa. UDK:n käyttöliittymä voi aluksi vaikuttaa vaikealta, joten selvitetään ensiksi käyttöliittymää.

Kun UDK avataan ja mitään asetuksia ei ole muutettu, niin perusnäkökuva aukeaa kuvan 8 mukaiseksi.



Kuvio 8. UDK-perusnäkökuva. (UDK Documentation. 2010)

Ylimpänä perusnäkökuva (kuvio 8) on ohjelman perusvalikot: File, Edit, View, Brush, Build, Tools ja Help. Näiden alta löytyy paljon toimintoja, joista osa on samoja kuin perusnäkökuvaan napeista tapahtuvat toiminnot. Perusnäkökuva vasen ylempi ruutu on oletuksena näkökuva kentästä suoraan ylhäältä päin kuvattuna, oikean ylempi ruutu on sivunäkökuva kentästä, vasen alempi ruutu on

perspektiivinäkymä, jossa voidaan liikuttaa näkymää vapaasti, ja oikea alempi ruutu on näkymä kentästä suoraan edestä päin kuvattuna.

Heti näiden valikoiden alla sijaitsee ylin Tool Bar (Kuvio 9):



Kuvio 9. Ylin Tool Bar. (UDK Documentation. 2010)

Kuvion 9 palkki sisältää paljon hyödyllisiä toimintoja. Toiminnot vasemmalta oikealle ovat: New, Open, Nuoli alas (antaa valita viimeksi auki olleista kentistä haluamansa, sama kuin File-valikossa Recent), Save current level (tallenna nykyinen kenttä), Save all levels (tallenna kaikki kentät), Save all writable packages (tallentaa kaikki tallennettavissa olevat paketit), tutut kumo- ja tee uudelleen -nuolet. Näiden jälkeen on Select-työkalu, jolla yksinkertaisesti vain valitaan kentän objekteja, Move (esimerkkikuvassa aktiivisena) toimii valinta- sekä objektien liikutustyökaluna, seuraavana Rotate-työkalu objektien pyörittämistä varten. Seuraavana on skaalaustyökalu (uniform): tämä skaalaa valitun objektin kokoa kaikkien akselien suhteen. Seuraava on skaalaustyökalu (non-uniform), muuten sama kuin edellinen, mutta skaalaa vain sillä akselilla jolla skaalataan. Seuraavana on pudotusvalikko, josta valitaan tiettyjen toimintojen ”referenssi-akseli”. Esimerkiksi jos skaalaataan jotain objektia Local-valinta valittuna x-akselilla isommaksi, niin skaalaustyökalu käyttää referenssinä objektin omaa akselistoa, eli sitä samaa, joka oli käytössä objektia mallinnettaessa. Jos tehdään sama World-valinta valittuna, niin skaalaus tapahtuu kentän yleisen akseliston mukaan. Eli kentän akselistolla skaalaaminen tuottaa eri tuloksen kuin objektin omalla akselistolla skaalaaminen.

Seuraavana on kiikarinkuvan omaava painike, Find Actors. Kun kenttää luodaan ja se saa yhä enemmän sisältöä itseensä, käy tämä painike käteväksi, jos ja kun täytyy löytää jokin tietty aktori (actor). Actor on mikä tahansa kenttään lisätty objekti tai toiminto. Tämä siis on actorien etsimiseen tarkoitettu työkalu.

Seuraavana on Content Browser (sisältöselain). Tämä on hyödyllinen painike, sillä kenttää luodessa tarvitsee usein mennä Content Browseriin.

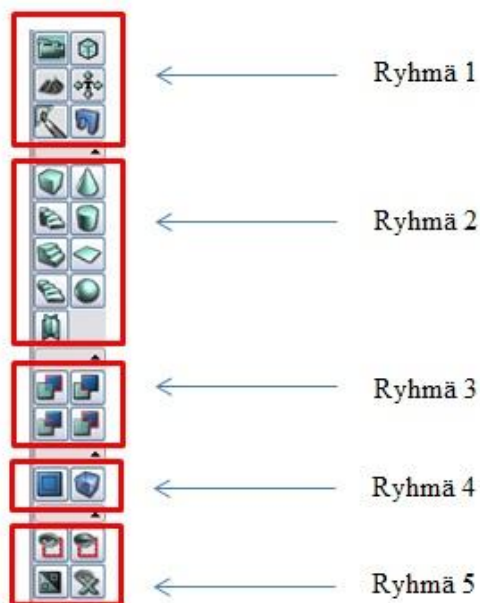
Seuraavana on Open UnrealKismet, joka avaa UnrealKismetin, jolla voidaan noodipohjaisesti ohjelmoida tapahtumia ja toimintoja kenttään, esimerkkinä vaikkapa hissi, avautuva ja sulkeutuva ovi tai vaikka vilkkuva valo.

Seuraava nappi on Open UnrealMatinee. Matineeella hallitaan kaikkea animaatioista ja partikkeliefekteistä kamera-ajoihin. Käytännössä se on käyräpohjainen ohjelma erilaisiin tilanteisiin.

Ainoa liikusäädin tässä palkissa on Distance to far clipping plane (etäisyys etäiseen ”katkaisutasoon”). Tämä säätää miten kaukaiset kohteet piirtyvät viewporttiin. Tätä voi kokeilla säätämällä liikusäädintä ja katsomalla mitä viewportissa (vasen ruutu alhaalla) tapahtuu. Tämä on kätevä ominaisuus, jos kenttä käy liikaa tietokoneen tehojen päälle ja on tarvetta saada turhia yksityiskohtia pois viewportista.

Seuraavana on Allow Translucent Selection, jolla annetaan valintojen mennä objektien ”läpi”, eli voidaan valita yksi tai monta objektia toistensa lävitse, jos tarve näin vaatii.

Sivupalkki (kuvio 10) on tärkeä osa kentän luomista. Tästä palkista löytyy tärkeimmät painikkeet kentän perusgeometrian luomiseen.



Kuvio 10. Sivupalkki. (UDK Documentation. 2010)

Ensimmäinen painike ylimmän kuuden ryhmästä (Ryhmä 1), kameran kuvan omaava, on nimeltään Camera mode (kamera tila). Tämä asettaa käyttäjän Camera-modeen. Tämä tila tulee tutuksi kaikille käyttäjille, sillä tämä on oletustila. Tässä tilassa ei voi muokata rakennussiveltimen (builder brush) geometriaa muuten kuin skaalaamalla sen asetuksia numeerisesti muuttaen. Kameran oikealla puolella on Geometry mode (geometria tila) -painike. Tämä antaa työkalut muokata Builder Brushia ja sen geometriaa hyvin samaan tapaan kuin 3D-mallinnusohjelmistoissa on tapana muokata 3D-mallien geometriaa.

Seuraavana on vuorenkukkuloilta näyttävä painike Terrain Editing Mode (maaston editointitila). Tässä tilassa voi luoda ja muokata kenttään maastoalueita korkeuskarttoineen, ruohoineen ja metsineen.

Painikkeen Texture Alignment Mode (tekstuurien suuntaus/linjaus) -tilassa käyttäjä voi liikutella, pyöritellä ja asetella tekstuureita kentässä.

Mesh Paint Mode (maalaustyökalu) -painike asettaa käyttäjän tilaan, jossa voidaan maalata valitsemiaan kentän osia.

Viimeinen painike ensimmäisten kuuden ryhmässä on Static Mesh Mode (staattisen mallin tila). Tämä tila avaa ikkunan, jossa voidaan asettaa raja-arvoja, joiden avulla voidaan lisätä staattisia objekteja kenttään asetettujen arvojen mukaan satunnaistettuina. Tämä tapahtuu valitsemalla haluttu malli Content Browserista, ja sitten lisäämällä halutussa näkymässä malli pitämällä pohjassa ALT+S ja klikkaamalla hiiren vasenta painiketta. Näppäinyhdistelmä ALT+S toimii myös normaalissa Camera Modessa, mutta siihen ei vaikuteta niillä arvoilla, jotka on määritelty Static Mesh Modessa.

Seuraava yhdeksän painikkeen ryhmä (Ryhmä 2) koostuu erilaisista geometrian luomistyökaluista. Ensimmäisenä on Cube (kuutio), peruskuutio. Tätä ja jokaista muuta tämän ryhmän painikkeista voi käyttää hiiren oikealla painikkella, joka avaa ikkunan, johon voi asettaa käsin haluamansa arvot. Kuution tapauksessa voi esimerkiksi päättää miten iso kuutio on missäkin suunnassa, onko se ontto, ja jos se on ontto, niin miten paksu seinämä siinä on.

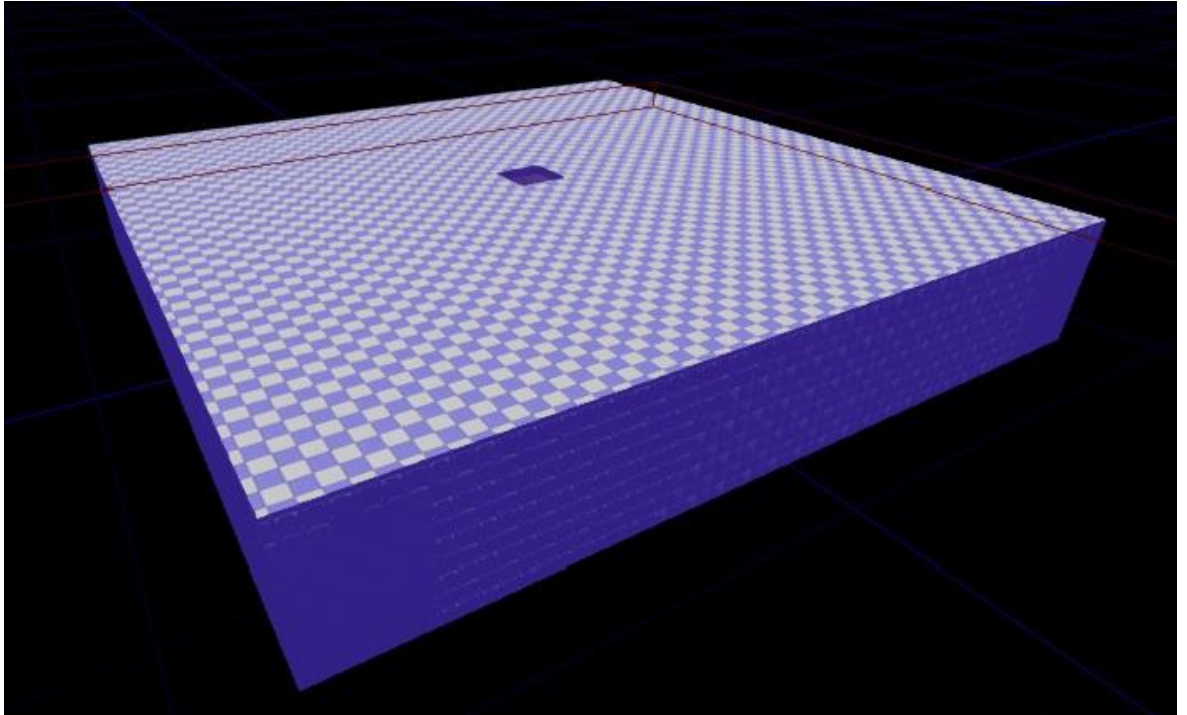
Seuraava neljän painikkeen ryhmä (Ryhmä 3) sisältää painikkeet boolean-tyyppisiin toimintoihin, jotka toimivat edellisten painikkeiden ”toteuttajana”. Ensimmäinen painike on Add, joka lisää Ryhmän 2 painikkeilla määrätyn geometrian kenttään. Seuraavana on Subtract, joka taas poistaa geometrian halutulta alueelta. Seuraavana on Intersect, jota painamalla geometriasivellin muotoutuu alkuperäisen siveltimen ja kentässäolevan geometrian leikkauskohtaan. Seuraava näppäin Deintersect on edellisen vastakohta, sillä tämä valitsee sen alueen, jossa sivellin ei leikkaa alkuperäistä geometriaa.

Kahden painikkeen ryhmä (Ryhmä 4) sisältää näppäimen erityissiveltimien luontiin (Add Special Brush), sekä painikkeen volyymien luontiin (Add Volume).

Viimeinen ryhmä (Ryhmä 5) sisältää näppäimet valintojen näkyvyyksille. Show Selected Only -painike piilottaa näkyvistä kaiken paitsi sillä hetkellä valittuna olevan kohteen, Hide Selected piilottaa vain valitun kohteen, Invert Selection kääntää valinnan jo valituista kohteista valitsemattomiin ja viimeinen painike Show All palauttaa kaiken takaisin näkyville.

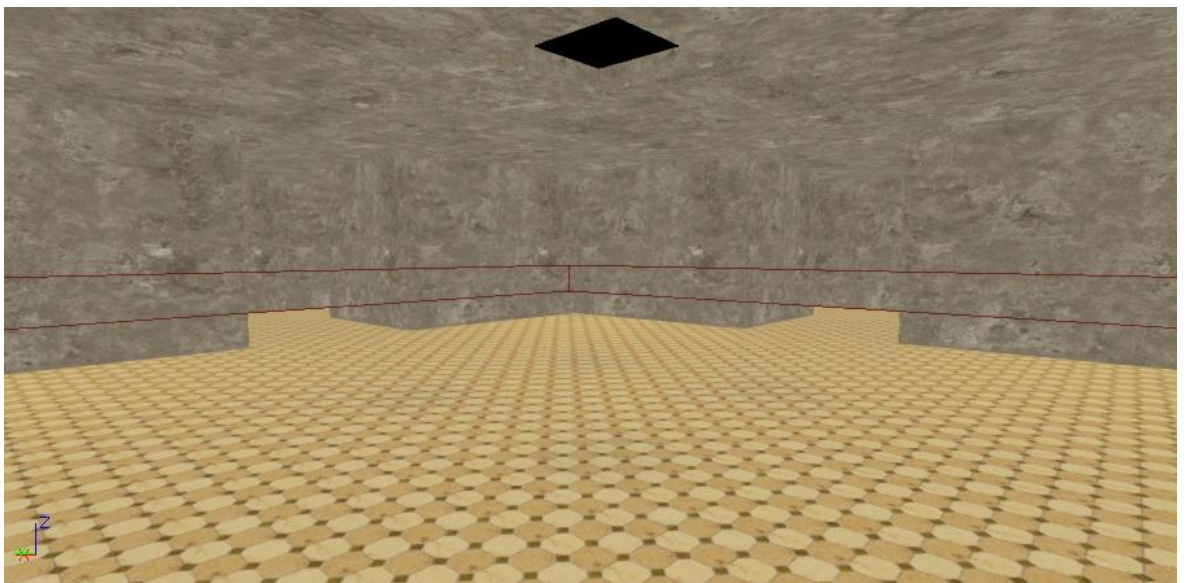
5.2.2 Oman pelikentän koostaminen UDK:lla

Kentän koostaminen aloitetaan perusgeometrian luonnilla. Tämä tapahtuu kokonaisuudessaan Kuvion 10 työkaluilla. Kentän perusgeometrian luonnin jälkeen kenttään kannattaa laittaa alustava valonlähde. Ensimmäisen kentän testauksen tulisi tapahtua tässä vaiheessa, jotta nähdään toimiiko perusgeometria halutulla tavalla.



Kuvio 11. Perusgeometria ulkopuolelta tarkasteltuna

Kuvio 11 esittää kentälle luotua perusgeometriaa ulkoa päin tarkasteltuna. Tämä näyttää hyvin yksinkertaiselta, sillä kenttä on sisätilakenttä, ja kentän ulkopuolella ei siksi ole mainittavia erikoisuuksia.

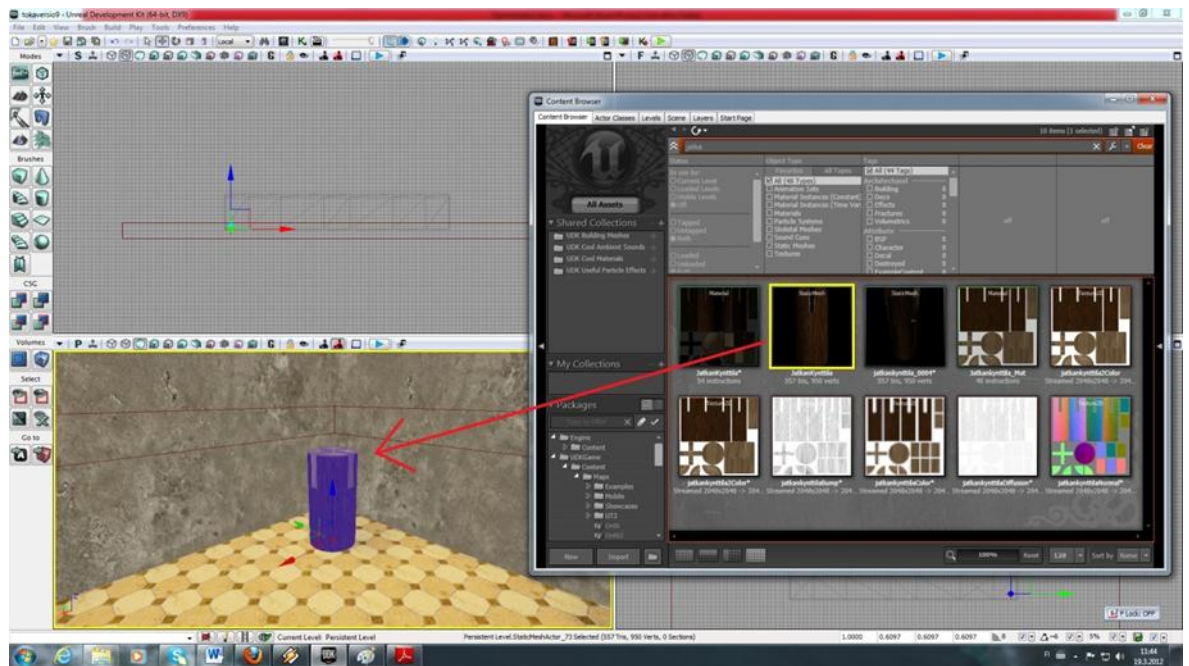


Kuvio 12. Perusgeometria sisäpuolelta tarkasteltuna

Kuviossa 12 nähdään kentän perusgeometriaa sisältäpäin tarkasteltuna. Näkymässä on myös lattian, seinien ja katon tekstuurit aseteltuina, jotta

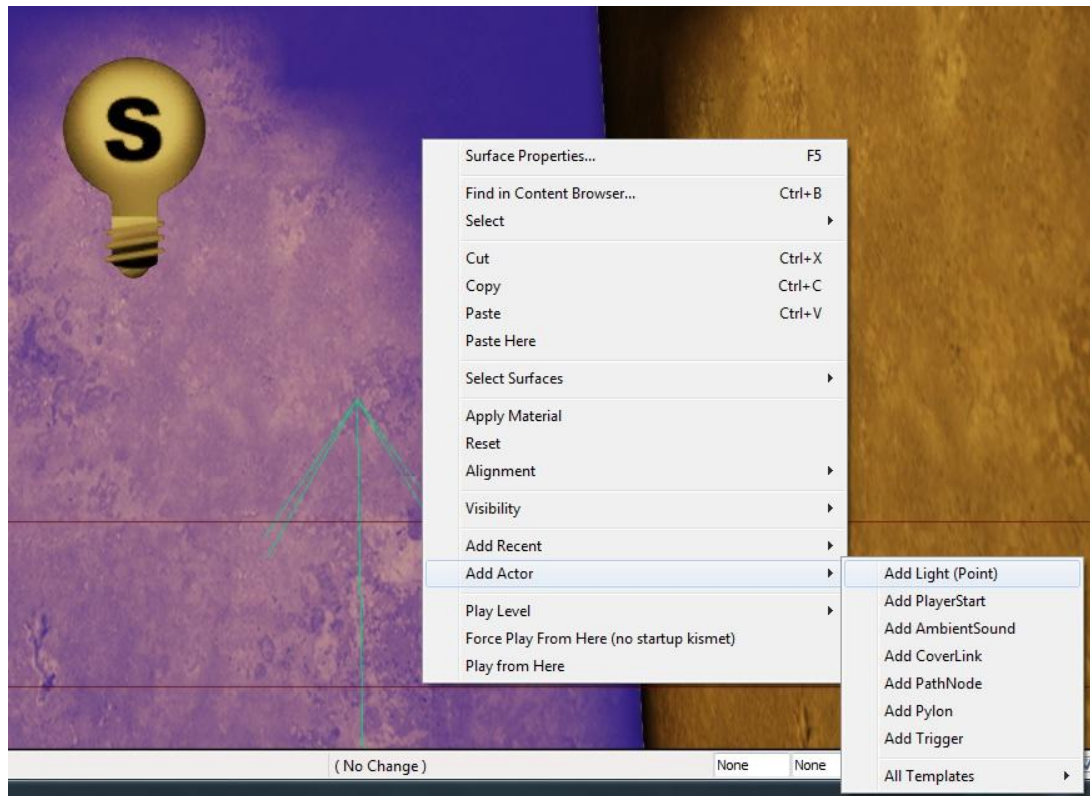
ympäristöstä saa paremmin selvää. Valaistusta ei ole tässä vaiheessa vielä rakennettuna kenttään, joten katto ja seinät näyttäivät yhdeltä ”massalta”.

Tämän jälkeen kenttään aletaan laittamaan haluttuja materiaaleja, koristeita ja staattisia esineitä, jotka luovat kenttään tunnelmaa sekä halutun tasoista realismia. Kaikki sisältö, jota kenttään halutaan laittaa, tapahtuu vetämällä se Content Browserista kenttään (Kuvio 13).



Kuvio 13. Sisällön lisääminen kenttään

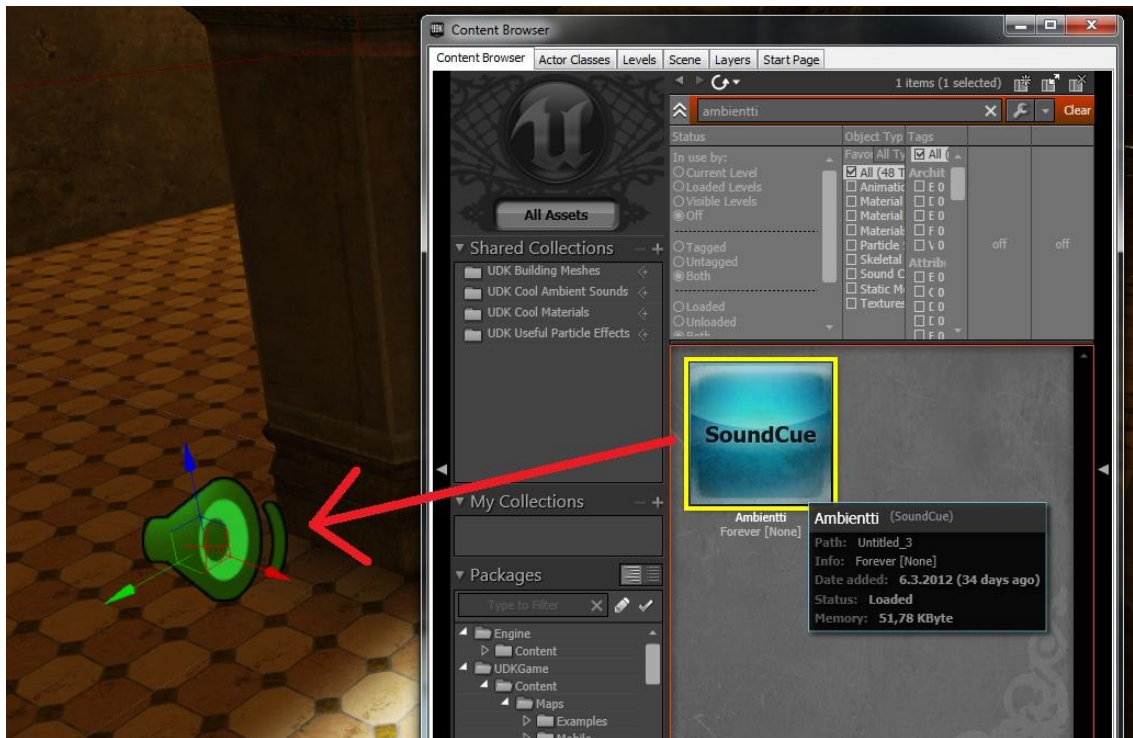
Kentän koostaminen on yksinkertaista. Tässä vaiheessa kannattaa pitää mielessään alkuperäinen suunnitelma (kuvio 1), jonka pohjalta annetaan luovalle prosessille valta. Kentän luova koostaminen tarkoittaa tässä tapauksessa sitä, että kentän tekijä lisää kenttään koristeita, muokkaa valaistusta sekä äänimaailmaa. Kenttää työstetään niin kauan kuin mahdollista tai kun lopputulos miellyttää kentän luoja.



Kuvio 14. Valojen lisääminen kenttään

Kuviossa 14 lisätään valo. Valon lisääminen tapahtuu klikkaamalla johonkin kohtaan kentässä hiiren oikealla painikkeella ja valitsemalla aukeavasta valikosta Add Actor>Add Light (Point). Tästä valikosta voidaan lisätä myös haluttaessa esimerkiksi pelaajan aloituspiste, ääniä tai tapahtumia aloittavia ”triggereitä”.

Äänien lisäys onnistuu myös Content Browserin kautta. Haluttu ääni etsitään Content Browserista, ja sitten se vedetään kenttään (kuvio 15).



Kuvio 15. Äänien lisääminen kenttään

Kaikki sisältö lisätään samalla periaatteella, eli etsitään Content Browserista haluttua sisältöä, ja vedetään se kenttään haluttuun sijaintiin. Kuvio 16 esittää kenttään lisättyjä koristeita ja valmista valaistusta. Kenttään on lisätty alkuperäisen suunnitelman mukaan jätkänkynttilöitä, sekä kuvion 16 oikeassa laidassa olevat keskimmäisen huoneen tukipilarit. Myös kattoon on aseteltu kulkemaan putkia, ja seinien kulmiin on laitettu betonisia pilareita koristeiksi. Kentän alkuperäinen suunnitelma ei sisältänyt kattoon aseteltuja putkia eikä kulmiin tulleita betonipilareita. Tämä on yksi esimerkki siitä, kuinka alkuperäistä suunnitelmaa ei aina voi noudattaa täysin kirjaimellisesti. Kenttää tehdessä huomattiin, että terävät kulmat nurkissa näyttävät vähän oudoilta, joten siksi kulmiin lisättiin betonipilarit. Samoin kävi katon putkien kanssa, koska katto näytti liian paljaalta.

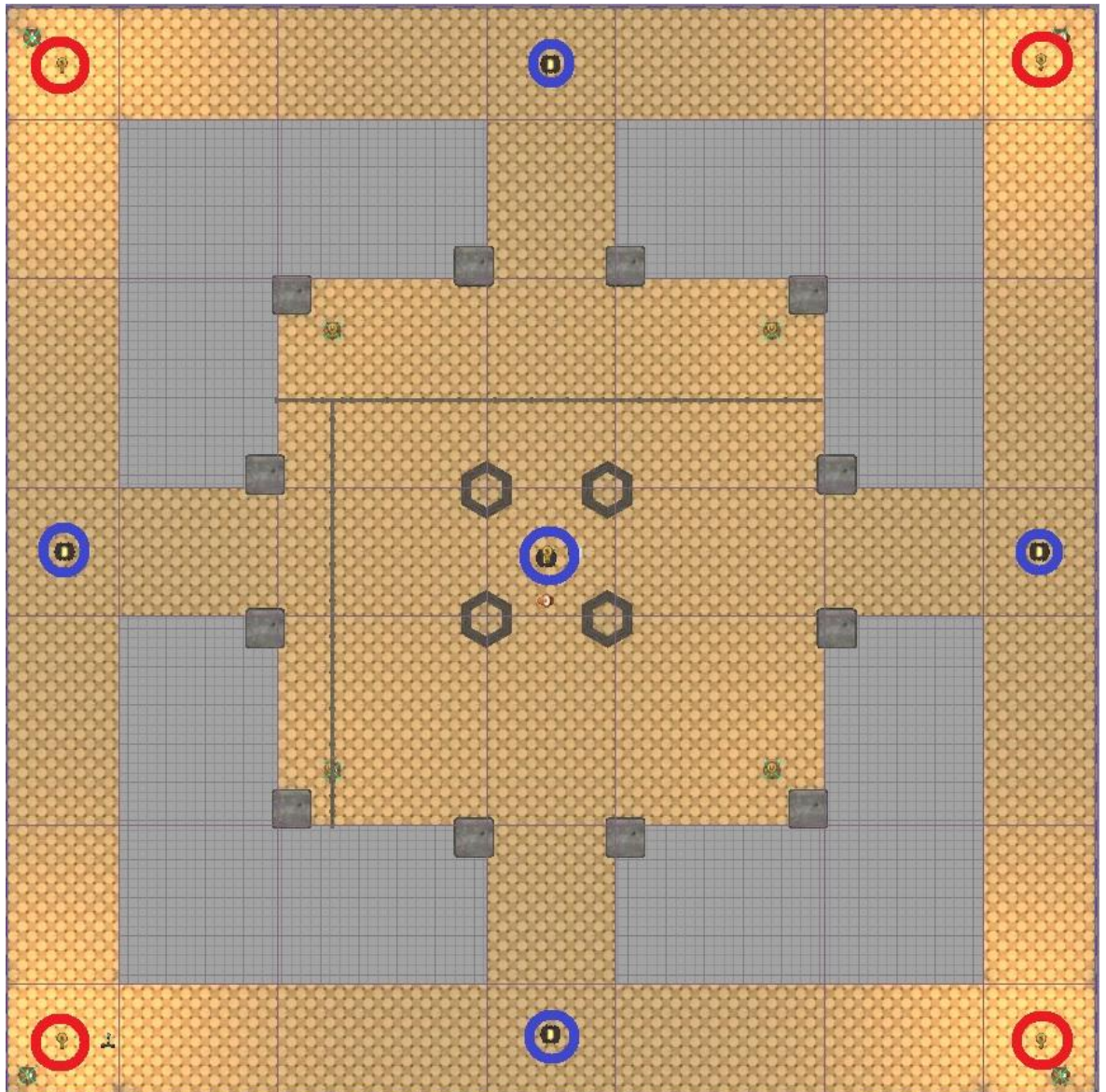


Kuvio 16. Koristeita

Tässä vaiheessa ajateltiin kentän olevan visuaalisesti tyydyttävä, joten moninpeliä yritettiin saada toimimaan, joka ei sitten kuitenkaan onnistunut. Tämän vuoksi pelin asetuksia ei käydy läpi tässä työssä, koska niitä säätämällä ei saatu mitään lisäarvoa kenttään. Moninpeli on kyllä mahdollista saada toimimaan, koska UDK-editorissa on valmiiksi sisäänrakennettu moninpelituki. Tähän työhön ei kuitenkaan moninpeliä mukaan saatu, koska sen toimivaksi saattamiseen olisi kulunut liian paljon aikaa, sillä UDK-editorin mahdollisten asetusten määrä on valtava. Internetistä löytyy paljon ohjeita moninpelitilan toteuttamiseen, joihin perehtymällä moninpelitila varmasti saataisiin toimimaan ajan kanssa.

5.2.3 Oman kentän toiminta

Kentän toiminta on hyvä havainnollistaa, jotta työn lukija saa kuvan siitä, miten kentän ollaan ajateltu toimivan. Usein voi käydä niin, että kentän toiminta paljastuu kokonaisuudessaan kentän toteuttajallekin vasta jälkeenkäpäin.



Kuvio 17. Kenttä ylhäältäpäin esitettynä

Kuviossa 17 esitetään kenttä ylhäältäpäin katsoen. Punaiset ympyrät merkitsevät pelaajien aloituspaikkoja ja siniset merkitsevät mistä pelaajat poimivat aseensa. Kuviossa 17 oleva harmaa alue tarkoittaa seiniä, ja oranssi kuvioitu alue on kentässä kuljettavaa aluetta. Kentän toimivuus perustuu pelaajien aloituspaikkojen ja poimittavien aseiden sijaintiin, sekä käytävien asetteluun. Kun pelaajat aloittavat kentän, he näkevät saman tien oikealle tai vasemmalle katsoessaan vastustajan, olettaen että pelissä on mukana 4 pelaajaa, sekä poimittavan aseeseen. Heti alussa pelaajan tulee siis tehdä päätös, lähteäkö oikealle vai vasemmalle keräämään asetta itselleen. Pelaajaan vaikuttaa myös psykologisesti epävarmuus siitä, mitä

toiset pelaajat aikovat tehdä. Pelaajat eivät voi varmuudella tietää lähtevätkö toiset pelaajat oikealle vai vasemmalle? Keskellä oleva ase on raketinheitin, joka on erittäin vahva ja haluttu ase, joten pelaajat joutuvat kisaamaan ja juoksemaan kilpaa kentän keskelle jos haluavat saada yliotteen muihin pelaajiin. Käytävillä olevat aseet ovat tavallisia plasmakivääreitä, jotka ovat tehokkaita, mutta paljon tehottomampia kuin kentän keskellä oleva raketinheitin. Tällainen alkuasetelma on hyvin tiivistunnelmainen ja edellyttää pelaajalta nopeaa toimintaa ja päätöksen tekoa.

5.3 Pelikentän testaus

Kentän testaus toteutettiin toimivuuden eli virheettömyyden osalta, ja sisällön vaikutus pelattavuuteen testattiin myös.

Kentän testaaminen tapahtuu kentän toteuttamisen alkuvaiheista loppuun saakka. Tämä on tärkeää siksi, että mahdolliset pelin rikkovat virheet huomataan mahdollisimman hyvissä ajoin.

Kentän pelitestaus alkaa heti kun kenttään saadaan perusgeometria rakennettua. Tämän työn kentässä havaittiin juuri tässä vaiheessa paha ongelma perusgeometriassa. Pelitestissä juostiin ympäri kentän käytäviä seinä "halailleen", ja tällä tavoin löydettiin sattumalta virhe geometriassa, jonka takia pelaaja pääsi liikkumaan seinän sisällä. Tällaista ei saisi tapahtua, joten virhe korjattiin pian sen löytämisen jälkeen, sekä kun oltiin selvitetty mistä kyseinen virhe johtuu.

Perusgeometrian virheen löytymisen jälkeen virheen syytä ruvettiin tutkimaan heti. Tällaiset virheet on hyvä havaita ajoissa, sillä jos tällaisia virheitä ei korjata heti, ei kentän työstämistä voida jatkaa.

Perusgeometrian virheen lopullista syytä ei koskaan löydetty, mutta tilanne saatiin korjattua rakentamalla seinien kohdalle Blocking Volumet, joka on näkymätön fyysinen este pelaajille. Tällä tavoin pelaajat saatiin pysymään halutulla alueella, eikä seinien sisälle enää päästy juoksemaan. Blocking volumet lisättiin kuvion 10 ryhmän 4 painikkeella Add Volume hiiren oikealla painikkeella.

Samalla varmistettiin Builder Brush -työkalulla, että Blocking Volume tulee oikeaan paikkaan.

Kenttää myöhemmin koristeltaessa huomattiin kentän käytävillä olevien holvikaarien olevan väärin aseteltuja, sekä huomattiin virhe lattian materiaalissa, joka teki siitä osittain läpinäkyvän. Nämä ongelmat oli helppo korjata poistamalla läpinäkyvyys kyseisestä materiaalista Material editorin kautta. Nämä eivät estäneet kentän toimivuutta merkittävästi.

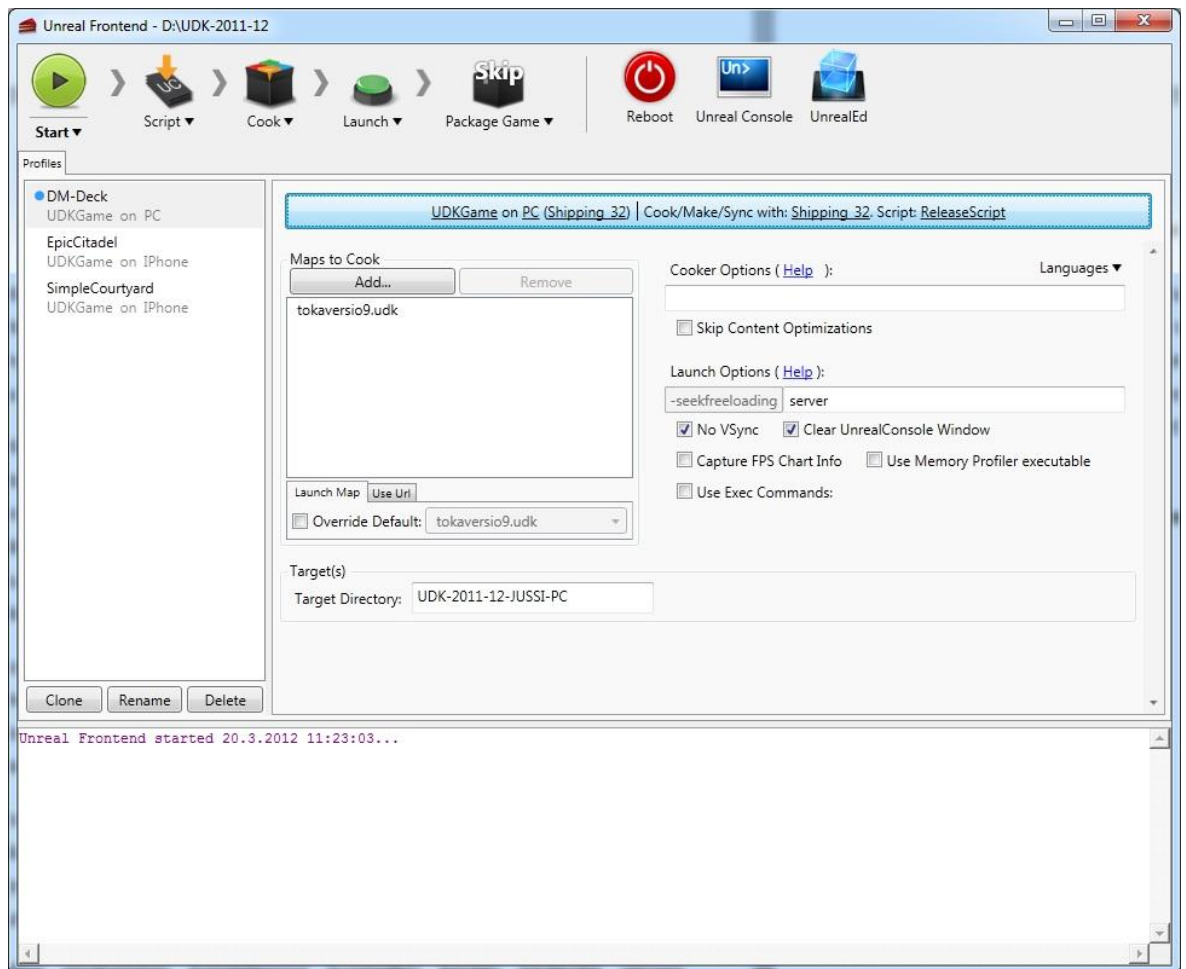
Kentän sisältö sopi alkuperäiseen suunnitelmaan ja korosti ankeaa katakombimaista tunnelmaa. Kattoon lisätyt putket eivät voineet vaikuttaa pelattavuuteen, koska pelaajat eivät pääse kattoon käsiksi. Kulmiin lisätyt betonipilarit lisäsivät tunnelmaa ja uskottavuutta odotetusti, eivätkä haitanneet kentän pelattavuutta.

Pelin kulun testaaminen toi esiin piirteitä, joita ei alkuperäisessä suunnitelmassa osattu ajatella. Esimerkiksi kentän psykologista puolta ei osattu suunnitteluvaiheessa ottaa huomioon. Kentässä pelaaminen on arvaamatonta, etenkin kun kuvitellaan neljä pelaaja pelaamaan yhtäaikaaisesti. Kentän yksinkertainen ja symmetrinen muoto asettaa kaikki pelaajat identtiseen lähtökohtaan, ja ainoana muuttujana on pelaajien päätös siitä, mihin liikkua alkuasetelmista. Tämä luo psykologisen paineen pelaajalle, jollaista ei suunnitellessa osattu ennakoida. Pelin kulku osoittautui testatessa siis paremmaksi kuin oltiin oletettu.

Kokonaisuutena testaus onnistui, ja vakavat virheet saatiin poistettua jo alkuvaiheessa.

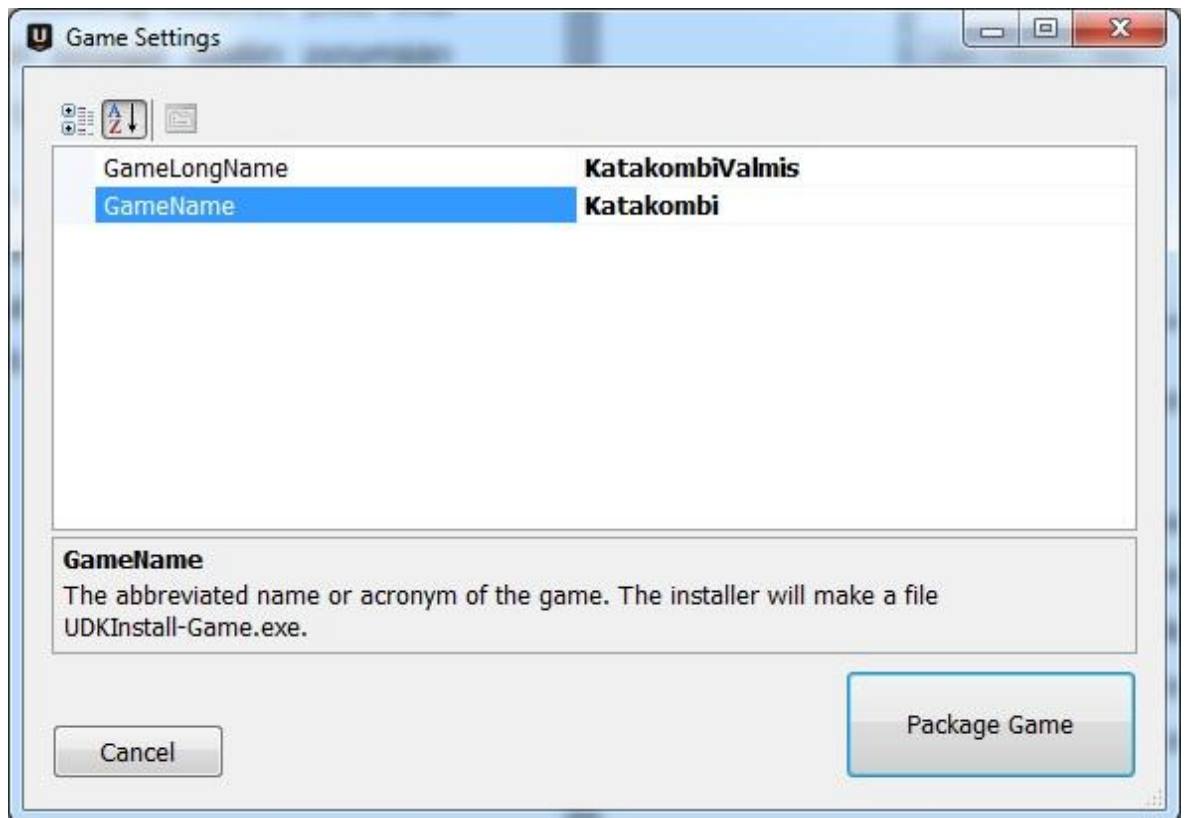
5.4 Valmis pelattavissa oleva pelikenttä

Kun pelikenttä on saatu halutun näköiseksi, ja se on läpikotaisin testattu virheiden ja sisällön pelattavuuden osalta, on se pakattava sellaiseen muotoon, että muut halukkaat voivat asentaa kentän ja pelata sitä. Tämä tarkoittaa myös sitä, että muut pelaajat eivät tarvitse UDK-editoria pelatakseen kenttää. Pakkaaminen tapahtuu UDK-editorin mukana tulevalla Unreal Frontend -ohjelmalla (kuvio 18).



Kuvio 18. UDK-editorin Unreal Frontend

Kentän pakkaaminen Unreal Frontendissä tapahtuu painamalla painiketta Maps to Cook, jonka jälkeen näyttöön aukeaa lista josta valitaan kenttä, joka halutaan pakata. Tämän jälkeen painetaan vasemman yläkulman painiketta Start. Tämä painike käskee Unreal Frontendin tekemän ”asennuksen” kyseisestä kentästä. Seuraava vaihe on painaa Package Game -painiketta, joka pakkaa kentän itsenäiseksi asennustiedostoksi. Painiketta painettua aukeaa ikkuna, jossa määritellään kentän asennuspaketin nimi (kuvio 19).



Kuvio 19. Pakattavan pelikentän nimi

Package Game -painike aloittaa kentän pakkaamisen. Kun pakkaaminen on valmis, näkyy pakatun kentän asennustiedoston polku Unreal Frontendin konsolissa viimeisimpänä. Esimerkkinä tämän työn kentän polku on "D:\UDK-2011-12\UDKInstall-Katakombi.exe". Kenttä on nyt valmiina jaettavaksi muille. Liitteessä 1 on kuvakaappauksia valmiista kentästä, pelattavissa oleva kenttä on liitteessä 2 (DVD-levy).

6 JOHTOPÄÄTÖKSET

Tämän työn keskeisin tulos on valmis pelattavissa oleva kenttä, sekä kenttäsuunnittelun oppimisprosessi ja itse kentän toteutus, sekä 3D-mallin luonti ja käyttö UDK-editorissa. Työ onnistui, vaikka tietyt tavoitteet eivät tulleet täytettyä. Kenttään ei saatu moninpeliominaisuutta toimimaan lukuisista yrityksistä huolimatta, mutta tämä ei ollut päätavoite, vaan päätavoitteena oli yleisesti toimiva kenttä. Oppimisprosessi oli haastava, sillä ennestään kentän suunnittelusta ei ollut mitään kokemusta, ja kaikki tieto tuli omaksua uutena samalla kun työtä tehtiin. Kentän luominen onnistui odotuksien mukaan. Unreal Development Kit on helppo oppia ja omaksua.

Suurin onglema työssä oli kentän perusgeometrian outo ja viallinen käyttäytyminen, jossa pelaajat pääsivät liikkumaan seinän sisässä. Tällaista onglemaa ole ennen ohjelmaa opeteltaessa tullut vastaan. Moninpeliominaisuuden toimimattomuus oli toinen ongelma, mutta tämä oli odotettavissa, sillä moninpeliominaisuudet vaativat suurempaa tietämystä UDK – editorin asetuksista, mitä työn tekemiseen käytettävissä olevaan aikaan ei ollut mahdollista opetella. Moninpeliominaisuuden puuttuminen ei kuitenkaan haitannut työn päätavoitteen loppuunsaattamista, koska kenttä on toimiva ja se oli päätavoitteena.

Kenttää voisi myös jatkokehittää. Jatkokehityksenä olisi tämän työn osalta puuttumaan jäänyt moninpelitilan toimivaksi saattaminen. Tähän löytyy runsaasti ohjeita Internetistä, sekä runsaamman opettelulla ja suuremmalla ajan investoinnilla moninpeli varmasti saataisiin toimimaan.

Tämän työn pohjalta toimenpide-ehdotelmana kenttäsuunnittelusta kiinnostuneille suositellaan UDK -editoria ja siihen internetistä löytyviä tutoriaaleja ja artikkeleita. UDK –editori on suhteellisen aloittelijaystävällinen, koska se ei vaadi juuri mitään erityisiä taitoja ennestään päästäkseen alkuun kenttien luonnissa. Yleisenä ohjenuorana pidetään hyvää kentän suunnittelua ja esivisualisointia, sillä se säästää aikaa ja vaivaa itse kentän toteutusvaiheessa.

UDK:lla ja muita tässä työssä esiteltyjä ohjelmia hyväksikäyttäen voidaan rakentaa kokonaisia pelejä käyttäen Unreal Engine 3 -pelimoottoria. Voidaan sanoa, että vain mielikuvitus on rajana, kun vain löytyy kärsivällisyyttä ja halua oppia uusia asioita.

LÄHTEET

Audacity ominaisuudet. 2012. [WWW-dokumentti]. [Viitattu 20.3.2012].
Saatavissa: <http://audacity.sourceforge.net/about/features>

Autodesk. 2012. [WWW-dokumentti]. [Viitattu 20.3.2012]. Saatavissa:
<http://www.autodesk.fi/>

Features. 2010. [WWW-dokumentti]. [Viitattu 20.3.2012]. Saatavissa:
<http://udk.com/features>

Features & Gallery. 2012. [WWW-dokumentti]. [Viitattu 20.3.2012]. Saatavissa:
<http://www.blender.org/features-gallery/>

GIMP. 4.5.2012. [WWW-dokumentti]. [Viitattu 20.3.2012]. Saatavissa:
<http://www.gimp.org/>

GIMP Feature overview. 2011. [WWW-dokumentti]. [Viitattu 20.3.2012].
Saatavissa: <http://www.gimp.org/features/>

Maxon. 2012. [WWW-dokumentti]. [Viitattu 20.3.2012]. Saatavissa:
<http://www.maxon.net/>

Meigs, T. 2003. Ultimate game design: Building game worlds. U.S.A.: McGraw-Hill/Osborne.

Stahl, T. 2005. Video Game Genres. [WWW-dokumentti]. [Viitattu 20.3.2012].
<http://www.thocp.net/software/games/reference/genres.htm>

UDK Documentation. 2010 [WWW-dokumentti]. [Viitattu 20.3.2012]. Saatavissa:
<http://www.udk.com/documentation/>

Webopedia. 2012 [WWW-dokumentti]. [Viitattu 20.3.2012]. Saatavissa:
<http://www.webopedia.com>

LIITTEET

LIITE 1. Kuvia valmiista kentästä

Kuva 1. Kentän keskialue

Kuva 2. Käytävä ulos keskialueelta

Kuva 3. Näkymä käytävälle

Kuva 4. Pelaaja ja raketinheitin

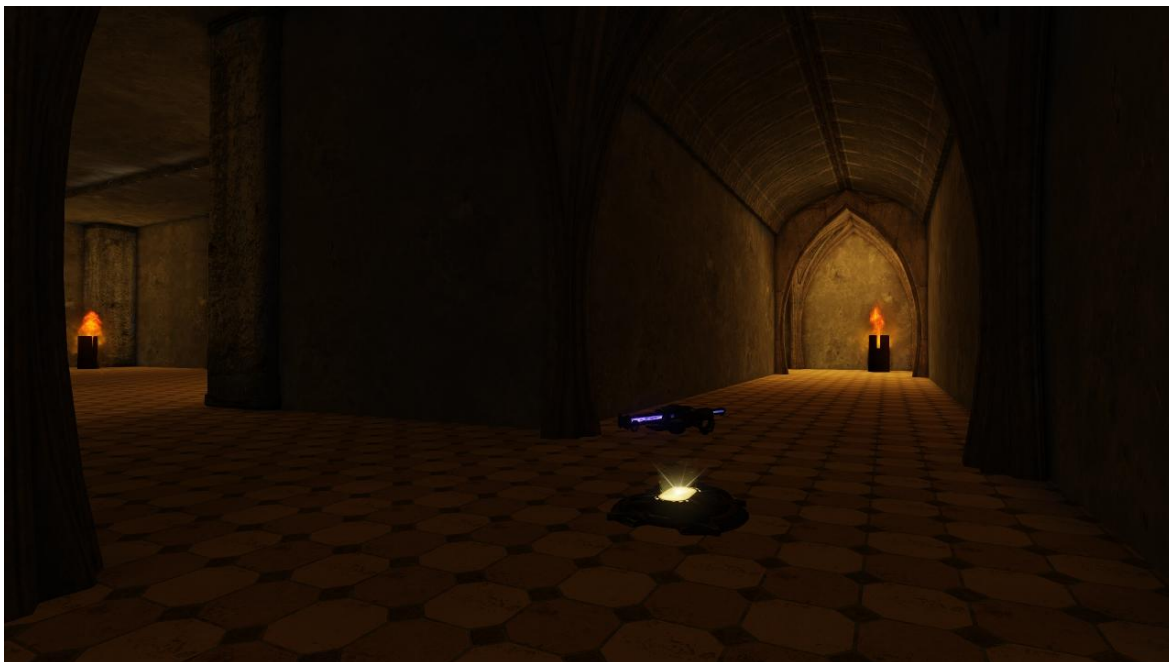
LIITE 2. Toteutettu pelikenttä DVD

LIITE 1 Kuvia valmiista kentästä 1

Kuva 1. Kentän keskialue



Kuva 2. Käytävä ulos keskialueelta

LIITE 1 Kuvia valmiista kentästä 2

Kuva 3. Näkymä käytävälle



Kuva 4. Pelaaja ja raketinheitin

LIITE 2 Toteutettu pelikenttä DVD