

Matti Kempainen

**3D-HAHMOJEN TOTEUTUS MOBIILPELIIN**

Opinnäytetyö  
Kajaanin ammattikorkeakoulu  
Luonnontieteiden ala  
Tietojenkäsittely  
Syksy 2012



Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittely koulutusohjelma
Tekijä(t) Matti Kemppainen	
Työn nimi 3D-hahmojen toteutus mobiilipeliin	
Vaihtoehtoiset ammattiopinnot Peliohjelmointi	Ohjaaja(t) Raimo Mustonen
	Toimeksiantaja
Aika Syksy 2012	Sivumäärä ja liitteet 64
<p>Mobiilipelien suosio on kasvanut räjähdysmäisesti viime vuosina älypuhelinkehittämisen myötä. Tässä opinnäytetyössä selvitetään kolmiulotteisen pelihahmon toteutusprosessi mobiilipeliin. Lisäksi pohditaan maksullisten ja ilmaisten ohjelmien eroja toteutuksessa. Pelihahmojen toteutus perustuu mobiilipeliin peliyrityksessä, jossa työskentelin graafikkona. Ohjelmien vertailussa on mukana 3D Studio Max, Blender 3D, Photoshop ja GIMP.</p> <p>Käytännön osuudessa käydään läpi 3D-pelihahmon toteutuksen vaiheet yksityiskohtaisesti suunnitteluvaiheesta viimeistelyyn. Jokaisessa vaiheessa huomautetaan mahdollisista eri ohjelmien eroista, jotka vaikuttavat vaiheen toteutukseen merkittävästi. Hahmon toteutuksessa on noudatettu yleisesti käytettyä 3D-peliobjektien toteutusprosessia. Tärkeimmät käsitellyt vaiheet ovat suunnittelu, 3D-mallinnus, UV-mappaus, riggaus, teksturointi, viimeistely ja vieminen Unity 3D-pelimoottoriin.</p> <p>Työn arvioinnissa todettiin sekä ilmaisten että kaupallisten ohjelmien pääsevän samoihin lopputuloksiin mobiilipelihahmon toteutuksessa. Alkuperäiset hahmot toteutettiin käyttämällä kaupallisia 3D Studio Max ja Photoshop ohjelmia ja projektin jälkeen testasin samoja vaiheita ilmaisilla ohjelmilla.</p>	
Kieli	Suomi
Asiasanat	3D, pelihahmo, toteutus, mobiilipeli
Säilytyspaikka	<input type="checkbox"/> Verkkokirjasto Theseus <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto



School Business	Degree Programme Business Information Technology
Author(s) Matti Kempainen	
Title 3D-character Creation for Mobile Games	
Optional Professional Studies Game Programming	Instructor(s) Raimo Mustonen
	Commissioned by
Date Fall 2012	Total Number of Pages and Appendices 64
<p>The popularity of mobile games has increased exponentially in recent years. The purpose of this thesis is to explain the process of creating a three-dimensional game character for mobile games. It also discusses the difference between commercial and free software in the production of characters. The production process is based on a real mobile game project in a game company where I worked as a graphic artist. The programs to be compared are 3D Studio Max, Blender 3D, Photoshop and GIMP.</p> <p>In the practical part of the thesis the creation of a 3D-game character is explained step by step, from planning to the finished product. At each stage of the process there are reminders of possible differences between programs having a significant role in the implementation. The commonly used stages of 3D-game character creation have been followed, i.e. 3D-modeling, UV-mapping, rigging, texturing, finishing and importing to Unity 3D game engine.</p> <p>In the evaluation of the thesis it was noted that both commercial and freeware programs were able to achieve the same results. The original characters were created using commercial programs and were recreated using totally free software.</p>	
Language of Thesis	Finnish
Keywords	3D, game character, implementation, mobile game
Deposited at	<input type="checkbox"/> Electronic library Theseus <input type="checkbox"/> Library of Kajaani University of Applied Sciences

## SISÄLLYS

1 JOHDANTO	1
1.1 Aihevalinnan perustelu	1
1.2 Tutkimusongelma	1
1.3 Käytännön osuus	2
2 MOBIILIALUSTAT	3
2.1 Apple iOS	3
2.2 Android	5
3 2D- JA 3D-PELIGRAFIikka	6
3.1 2D-grafiikka	6
3.2 3D-grafiikka	7
4 GRAFIIKAN TUOTANNON VAIHEET	8
4.1 Konseptitaide	8
4.2 Pohjapiirrokset	9
4.3 3D-mallinnus	10
4.4 Optimointi ja viimeistely	13
4.5 UV-mappaus	14
4.6 Teksturointi	15
4.7 Riggaukset ja animaatio	16
5 PELIHAHMON SUUNNITTELU	19
6 3D-GRAFIikkaOHJELMAT	21
6.1 Blender 3D	21
6.2 3D Studio Max	22
7 2D-GRAFIikkaOHJELMAT	24
7.1 Adobe Photoshop	24
7.2 GIMP	25

8 PETS VS ORCS PELIPROJEKTI	26
8.1 Tietoa pelistä	27
9 HAHMON TUOTANTO	28
9.1 Suunnittelu ja konseptitaide	28
9.2 3D-mallinnus	30
9.2.1 Mallinnusprosessi	30
9.3 UV-mappaus	35
9.4 Animaatio ja riggaus	38
9.5 Teksturointi	42
9.6 Toimivuus Unityn kanssa	45
9.7 Käyttöliittymät	47
10 LOPPUTULOKSET JA ARVIOINTI	50
10.1 Ohjelmistojen muut erot	52
10.2 Omat kommentit	52
11 LÄHTEET	55

## SYMBOLILUETTELO

2D	Kaksiulotteinen
3D	Kolmiulotteinen
3D-malli	Matemaattinen esitys kolmiulotteisesta objektista 3D-ohjelmassa.
3DSMax	3D Studio Max, Autodeskin kehittämä 3D-mallinnusohjelma.
Edge	3D-mallin osa, joka yhdistää verteksit toisiinsa.
Face	Pinta, joka koostuu vähintään kolmesta edgestä.
GIMP	GNU Image Manipulation Program, ilmainen kuvankäsittelyohjelma.
Mesh	Objekti, joka koostuu vertekseistä, edgeistä ja faceista.
NURBS	Non-uniform rational basis spline, matemaattinen malli, jota käytetään käyrien ja pintojen luomiseen ja esittämiseen tietokonegrafiikassa.
Polygon	Koostuu useasta facesta.
Riggaus	Tekniikka, jolla mahdollistetaan 3D-mallin liikuttaminen animoinnissa.
UV-mappaus	Tekniikka, jolla tehdään kolmiulotteisesta objektista kaksiulotteinen versio. Käytetään teksturoinnissa.
Verteksi	Piste, jossa kaksi tai useampi linja yhdistyy. Polygonin kulma.

## 1 JOHDANTO

Opinnäytetyön aiheena on tutkia pelihahmojen tuottamista mobiilipeliprojektissa ja miten tuotanto eroaa eri grafiikkaohjelmien välillä. Käytännössä selvitetään kaikki grafiikan tuotannon vaiheet ja selvitetään, eroaako eri vaiheiden toteutus eri ohjelmien välillä. Grafiikan tuotanto keskittyy animoituihin 3D-pelihahmoin, joita on käytetty mobiilipeliprojektissa. Hahmoja käytetään käytännön osuudessa havainnollistamaan työvaiheita.

### 1.1 Aihevalinnan perustelu

Ajatus opinnäytetyöhön tuli työskennellessäni peliyrityksessä muutaman kuukauden ajan, jossa toteutin 3D-grafiikkaa. Tuotantomme keskittyi mobiilialustalle ja käytännön osuudestani voi olla paljon hyötyä mobiilipelialalle tähtäävälle graafikolle. Mobiilipelien suosio on kasvussa, joten tulevaisuudessa syntyy varmasti vielä monia alan yrityksiä Suomeen. Konsolipeliprojekteihin verrattuna, mobiilipeli-projektit ovat huomattavasti kevyempiä, joka houkuttelee myös alalle uusia tekijöitä. Mobiilipelien kasvava suosio on myös hyvä syy tutkia grafiikan tuotantomenetelmiä. Ohjelmien vertailu voi myös auttaa aloittelevaa pelialan yrittäjää päättämään käytettävät työkalut.

### 1.2 Tutkimusongelma

Tutkimusongelmana opinnäytetyössäni selvitetään, kuinka voidaan tuottaa tehokkaasti 3D-pelihahmoja mobiilipeliin. Samalla myös tutkitaan onko eri grafiikkaohjelmilla suuri merkitys esimerkiksi tuotannon nopeuteen. Tavoitteena on selvittää grafiikan tuotannon eri vaiheet ja auttaa käyttäjää pääsemään tehokkaasti eteenpäin projekteissa. Ohjelmien käyttöön vaikuttaa myös kokemus ja henkilökohtaiset mieltymykset, mutta tämä tutkimus antaa hyvän kuvan ohjelmien peruskäytöstä mobiilipelituotannossa. Ohjelmissa ei ole myöskään käytetty ohjelmistolaajennuksia.

### 1.3 Käytännön osuus

Tein peligrafiikkaa peliyrityksessä Pets Vs Orcs- mobiilipeliin, jonka parissa työskentelin muutaman kuukauden. Suurin osa grafiikoista on toteutettu projektimme toisen graafikon kanssa. Alustana toimii iOS ja Android, joka täytyi ottaa huomioon grafiikoiden tuotannossa. Peli toteutettiin Unity 3D -pelimoottorilla, joten grafiikan täytyi olla tehokkaasti hyödynnettävissä pelimoottorissa. Pienen tiimimme takia, lopputuloksen oli oltava näyttävää mahdollisimman pienillä resursseilla ja mahdollisimman nopeasti. Pelimme julkaistiin iOS- ja Android- alustalle, joista kerrotaan myös lyhyesti.

Opinnäytetyössä esittelee lyhyesti mobiilialustoja, sekä vertailua 2D- ja 3D-grafiikan käytettävyydessä peleissä. Lisäksi käydään läpi teoriaa peligrafiikasta, grafiikan tuotannosta sekä lyhyesti eri grafiikkaohjelmista. Lopuksi perehdytään käytännön osuuden grafiikan tuotantoprosessiin, sekä vertaillaan millä grafiikkaohjelmistolla olisi päässyt samoihin tavoitteisiin tai jopa suoriutumaan paremmin. On tärkeää myös tarkkailla, miten pelihahmoja voisi vielä parantaa.



## 2 MOBIILIALUSTAT

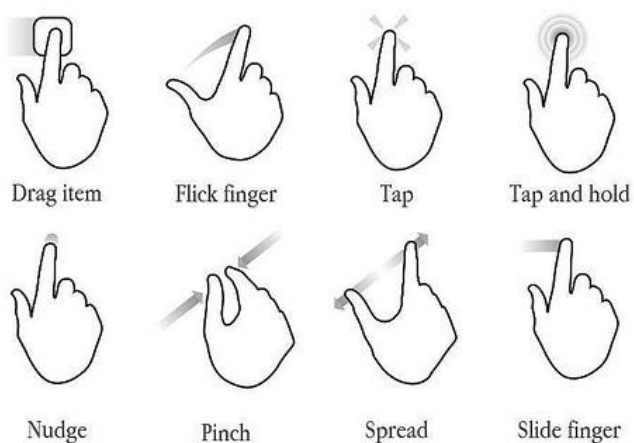
Viime aikoina älypuhelimet ovat yleistyneet räjähdysmäisesti ja tablet-tietokoneet ovat myös alkaneet vallata markkinoita. Älypuhelimet ovat vaikuttaneet ominaisuuksillaan ihmisten elämään ja myös pelimarkkinoihin. Eri puhelinvalmistajien suurimmat valttikortit ovat käyttöjärjestelmät, joista yleisempiä ovat Applen iOS ja Googlen Android. Nykyaikaisia mobiilikäyttöjärjestelmiä hyödynnetään älypuhelimissa, tableteissa sekä muissa mobiililaitteissa. Älypuhelimia ohjataan kosketusnäytön avulla, joka on vaikuttanut suuresti myös mobiilipelien kehitykseen. (AddictiveTips 2011.)

### 2.1 Apple iOS

Yleisimpiin mobiilialustoihin kuuluu Applen iOS, jota käytetään Applen kehittämässä iPhone-älypuhelimissa sekä iPad-tableteissa. Esimerkiksi iPhonella onnistuu kätevästi soittaminen, viestien lähettäminen, valokuvaaminen, kuvien ja videoiden katselu, internetin selaaminen sekä pelien pelaaminen. Pelit ladataan iOS-laitteisiin App Store -palvelun kautta, joka sisälsi syyskuussa 2012 yli 700 000 ladattavaa sovellusta ja peliä. Sovellukset ladataan laitteisiin nettiyhteyden avulla ja maksut tapahtuvat esimerkiksi luottokortin tai PayPal-tilin kautta. Sovellusten lataaminen onnistuu myös tietokoneelle, josta ne voidaan myöhemmin siirtää laitteisiin käytettäväksi. (Apple a 2012; Techcrunch 2012.)

IOs-käyttöjärjestelmän hallinta tapahtuu kosketusnäytön kautta sormilla ja käden eri liikkeillä (Kuvio 1.). Sormien liikkeillä on yleensä samoja tarkoituksia käyttöjärjestelmän hallinnassa ja pelien ohjaamisessa. Kosketusnäytöissä on myös yleensä multi-touch - ominaisuus, eli näyttö tunnistaa useamman kuin yhden sormen kerrallaan. iPhone- ja iPad-laitteissa on myös kiihtyvyyssanturit, jotka tunnistavat esimerkiksi laitteen kallistelun. (Dummies 2012.)

Vaikka älypuhelimia on ollut markkinoilla jo 90-luvulta lähtien, sai Applen vuonna 2007 julkistama iPhone vasta kuluttajat tosissaan siirtymään älypuhelin-aikakauteen. (Apple b 2012; AddictiveTips 2011.)



Kuvio 1. Kosketusnäytöllä käytettävät ohjausmenetelmät (Curioux Blog)

Pelien kehitys iOS-alustalle onnistuu Applen iOS Development Kit –työkalun avulla. Työkalu on ladattavissa Applen sivuilta ja hintaan 99 dollaria vuodessa. Eri pelimoottorit tukevat myös iOS-alustaa, kuten Unity 3D. (Apple c 2012; Idevgames 2011.)



Kuvio 2. Pelaamista iPhone 3G:llä (SlashGear)

## 2.2 Android

Android on Linux-pohjainen käyttöliittymä älypuhelimille ja tableteille. Sen on kehittänyt yhteistyössä Open Handset Alliance ja Google. Android on nykyään yksi menestyneimmistä älypuhelinlustoista. Android Market kilpailee rinta rinnan Applen App Storen kanssa ja tarjoaa myös kattavan valikoiman sovelluksia ja pelejä. Android on myös varteenotettava ja kilpailukykyinen alusta esimerkiksi pelikehittäjille. Android-alusta onkin jo kerännyt suuren määrän kehittäjiä ja oli vuonna 2010 eniten myynyt mobiilialusta. Android Marketissa kehittäjät voivat julkaista ja myydä pelejään vaivattomasti. Pelien julkaisemiksi käyttäjän täytyy luoda kehittäjäprofiili ja maksaa rekisteröintimaksu. (Canalys 2011; Android Developers 2012.)

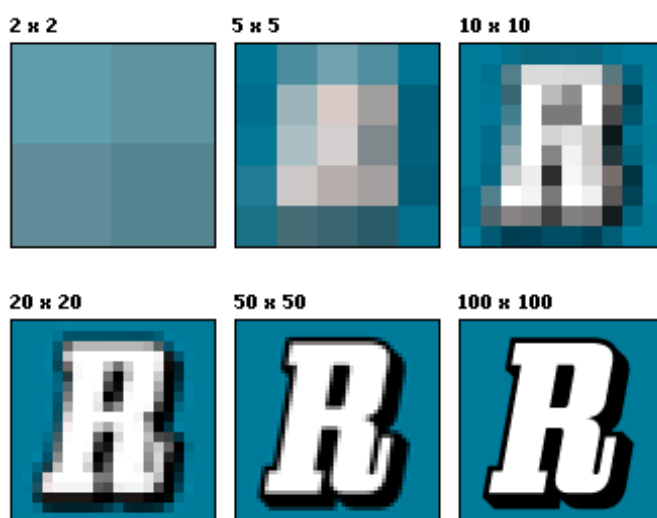
Android-kehitykseen vaaditaan Java- ja Android-kehitystyökaluja. Varsinaiseen ohjelmointiin käytetään Eclipse-ohjelmaa, johon liitetään Android-kehitykseen käytettävät ohjelmistolaajennukset. Kaikki tarvittava ohjelmisto voidaan ladata internetistä veloitusetta. Pelien testaamiseen tarvitaan myös siihen sopiva mobiililaite. (Skill-Guru 2011.)

### 3 2D- JA 3D-PELIGRAFIikka

2D- ja 3D-peligrfiikkaa tuotetaan digitaalisessa muodossa tietokoneella, käyttäen siihen erikoistuneita ohjelmia. Monet pelit käyttivät esirenderöityjä 3D-malleja grafiikkana, ennen kuin koneet kykenivät esittämään niitä reaaliajassa. Teknologian kehittyessä 3D-grafiikasta on tullut yleisempää, kuin 2D-grafiikasta. 3D-grafiikka tuli tunnetuksi 90-luvulla esimerkiksi Quake -pelisarjan ja Toy Story -animaation myötä. (Giant Bomb 2012.)

#### 3.1 2D-grafiikka

Digitaalista 2D-grafiikkaa käytetään tekstin ja kuvien esittämisessä tietokoneella. Nykyään lähes kaikki kuvat ovat jollain tapaa tietokoneella tuotettuja ja tietokonegrafiikkaa käytetään paljon myös digitaalisten valokuvien parantelussa. Tietokonegrafiikan tuotanto oli ennen hyvin monimutkaista ja kallista, mutta nykyään jokainen tietokoneen omistaja pystyy siihen esimerkiksi ilmaisten ohjelmistojen avulla. Grafiikkaa tuotetaan pääasiallisesti kuvankäsittelyohjelmilla, joilla onnistuu sekä piirtäminen, että kuvien parantelu. Digitaalisiin kuviin luokitellaan vektori- ja rasterikuvat, joista rasterikuvat ovat yleisemmin käytettyjä. Rasterikuvat koostuvat pikseleistä, joille voidaan määrittää esimerkiksi oma väriarvo. Kuvan yksityiskohtaisuuden määrää, kuinka monesta pikselistä se koostuu. (New World Encyclopedia 2008; Ezine Articles 2008.)



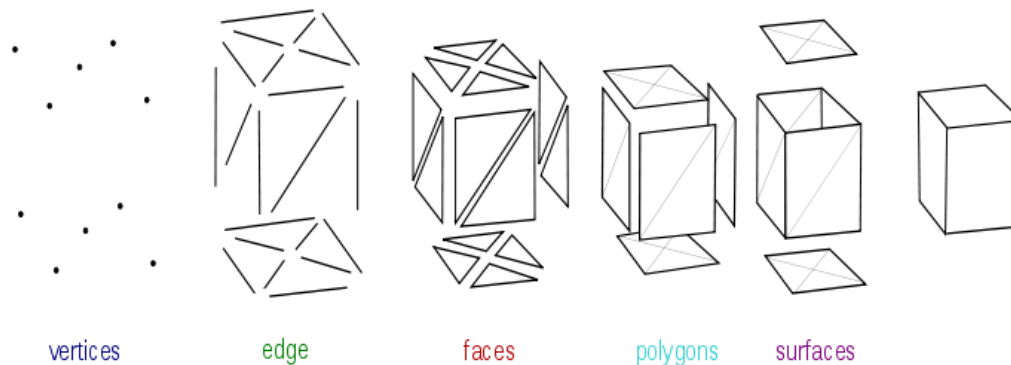
Kuvio 3. Pikselien määrän merkitys digitaalisissa kuvissa (Patriot Truck Leasing)

Vektorigrafiikkaan ei vaikuta kuvan resoluutio. Kuvan muodot ja ominaisuudet perustuvat koordinaatteihin sekä matemaattisiin funktioihin, joka mahdollistaa kuvan suurentamisen ilman, että sen tarkkuus kärsii. (Ezine Articles 2008.)

Kaksiulotteisissa peleissä objektit voivat liikkua vain horisontaalisella tai vertikaalisella akselilla. 2D-peliä suunniteltaessa on päätettävä lopullinen kuvakulma, koska kaikki grafiikka on piirrettävä noudattamaan kameran sijaintia. Kuvakulman avulla voidaan myös luoda illuusio kolmiulotteisuudesta, esimerkiksi isometrisissä peleissä pelilaatat on piirretty näyttämään kolmiulotteisilta. Hyvä esimerkki perinteisestä 2D-peleistä on Super Mario Bros., joka on vaikuttanut paljon tasohyppelypelien kehitykseen. (GiantBomb 2012; New World Encyclopedia 2008.)

### 3.2 3D-grafiikka

Kolmiulotteisessa tilassa käytetään kolmiulotteista koordinaatistoa, johon kuuluu vaaka-, pysty- ja syvyysakseli. Peleissä käytettävät 3D-mallit koostuvat pisteistä eli verteksistä, jotka sijaitsevat kolmiulotteisessa tilassa. Verteksit yhdistää toisiinsa viivat, joita kutsutaan edgeiksi. Monikulmio, eli polygon on struktuuri, joka koostuu vähintään kolmesta verteksistä. Yhdessä polygonit muodostavat 3D-mallin, joita peleissä käytetään. Mallin yksityiskohtaisuus määräytyy polygonien määrästä (Kuvio 4). (About.com b 2012.)

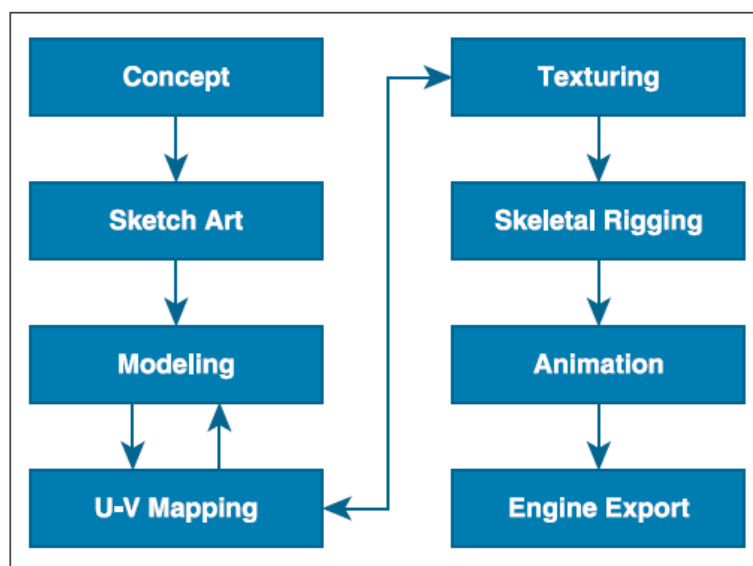


Kuvio 4. Palaset, joista 3D-objekti koostuu (Feedagg 2011.)

## 4 GRAFIIKAN TUOTANNON VAIHEET

Grafiikka on ensimmäinen asia, johon käyttäjä kiinnittää huomiota pelissä. Tässä tutkimuksessa käsitellään grafiikan tuotantoa mobiilialustojen rajoitusten mukaisesti. Lisäksi käydään läpi tuotantovaiheet 3D-hahmojen toteutuksessa.

Hahmojen tuotanto voi vaikuttaa monimutkaiselta, mutta prosessi on hyvin lineaarinen. Vaikka laitteiden tehot ja hahmojen ulkonäkö on kehittynyt vuosien varrella, tuotantoprosessi on pysynyt hyvin samanlaisena. (Franson & Thomas 2007, 2).



Kuvio 5. 3D-pelihahmon tuotantoprosessi (Franson & Thomas 2007, 2).

### 4.1 Konseptitaide

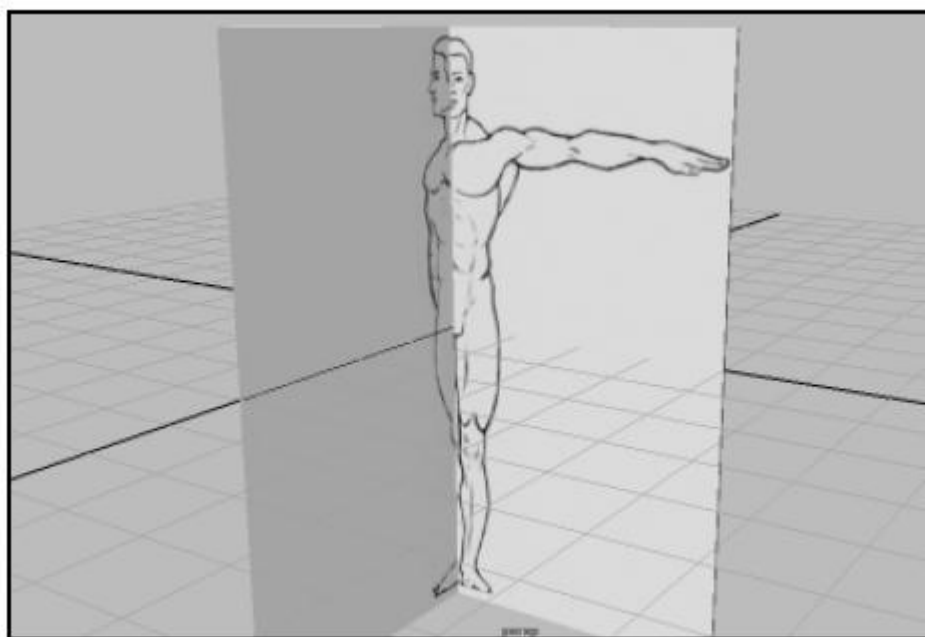
Konseptitaiteen tarkoitus on visualisoida peli-idea ennen lopullista tuotantoa. Konseptikuvat eivät ole välttämättä viimeistelyjä, eivätkä edusta pelin lopullista tyyliä. Alussa pyritään tuottamaan paljon karkeita hahmotelmia, joista valitaan sopivimmat tyylit käytettäväksi. Jokaista hahmotelmaa on turha tässä vaiheessa viimeistellä, koska se vie vain paljon aikaa. Hahmotelmissa on tärkeää kuvata pelin yleinen tyyli. Konseptitaiteen tuotanto riippuu myös paljon konseptitaiteilijan työtavasta. (LudoCraft 2006; ImagineFX 2007.)

Valmiit konseptit auttavat jakamaan yhteisen visualisoinnin pelistä koko tuotantotiimin kesken ja auttavat hahmottamaan, miltä peli lopulta näyttää. Konseptitaidetta tehdään pelin

kentistä, hahmoista ja esimerkiksi tavaroista. Lopulliset konseptit kerätään yhteen ja niitä käytetään ohjeistuksena, kun pelissä käytettävää grafiikkaa toteutetaan. Kuvat voivat toimia referenssimateriaalina 3D-graafikoille, animaattoreille sekä teksturoijille. Joskus on myös tapana tehdä konseptikuvista Art Design Document, jossa on kaikki pelin sisältämä visuaalisuus. (LudoCraft 2006; ImagineFX 2007.)

## 4.2 Pohjapiirroksiset

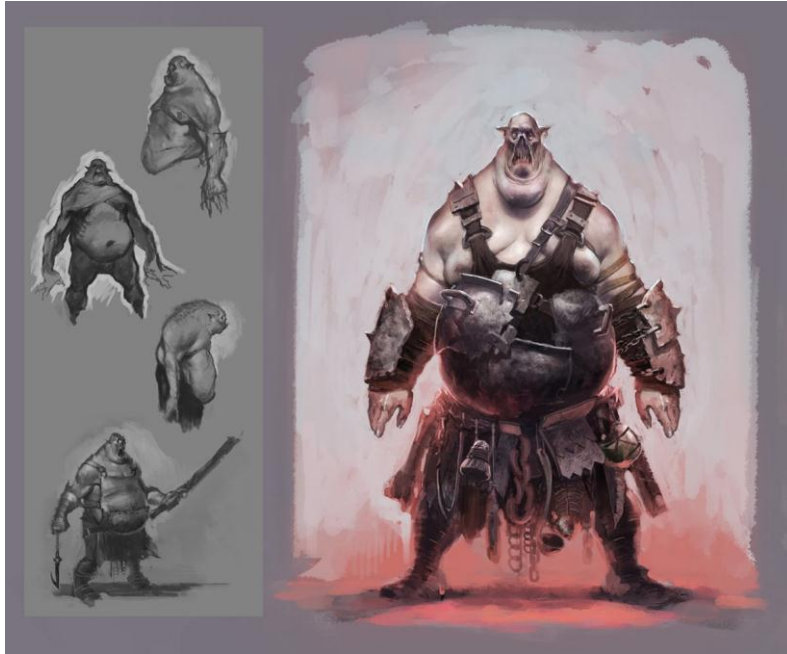
Pohjapiirros on ortografinen ja yksityiskohtainen kuvitus pelihahmosta, jota käytetään apuna hahmon toteutuksessa. Ne muistuttavat esimerkiksi talojen pohjapiirroksia, joissa kuvataan jokin asia monesta eri kuvakulmasta. Useimmissa piirroksissa on etu- ja takakuvakulmat, mutta joissakin tapauksissa on lisäksi sivu- ja yläkuvakulmat, varsinkin jos hahmo on nelijalkainen. (Pardew 2005, 10).



Kuvio 6. Pohjapiirroksen käyttö 3D-mallinnusohjelmassa. (Pardew 2005, 11)

Pohjapiirroksia käytetään yleensä pelihahmon 3D-mallin toteutuksessa. Mallintaja voi käyttää niitä pohjana tehdessään hahmon geometriaa. Pelin suunnitteluun sisältyy yleensä monta piirrosta, yksi jokaista hahmoa kohtia. Joskus niitä tehdään myös tärkeistä tavaroista. Nämä tavarat ovat yleensä kriittisiä pelissä etenemiselle tai niillä on muuten tärkeä rooli. Joissakin

peleissä pelattava hahmo muuttuu pelin edistyessä. Esimerkiksi roolipeleissä hahmot voivat saada uusia varusteita, joten tarvitaan useampia konseptipiirroksia. (Pardew 2005, 10).



Kuvio 7. Pelihahmon konseptitaidetta. Vasemmalla luonnoksia ja oikealla lopullinen hahmo (Pedronunezart)

#### 4.3 3D-mallinnus

3D-mallinnusvaihe on hahmon tuotannon monimutkaisin. Vaikka markkinoilla on useita eri mallinnusohjelmia, mallinnustekniikat ovat yleensä samoja. Mallinnusvaiheessa luodaan kolmiulotteisia objekteja peliin käyttämällä 3D-mallinnusohjelmaa. Pelituotannossa 3D-objektit voidaan karkeasti jakaa kahteen ryhmään: staattiset ja dynaamiset. (LudoCraft 2006.)

Kaikki objektit, joilla ei ole animaatiota, kutsutaan staattisiksi objekteiksi. Esimerkiksi, rakennukset, maasto ja dekoraatiot ovat staattisia. Dekoraatioihin luokitellaan kaikki objektit, joita käytetään pelin ympäristössä koristeena ja joihin pelaajalla ei ole mitään hallintaa. Staattisten objektien tuotanto on yleensä hyvin yksinkertaista, koska toteutuksessa tarvitaan vain 3D-malli ja tekstuuri. (LudoCraft 2006.)

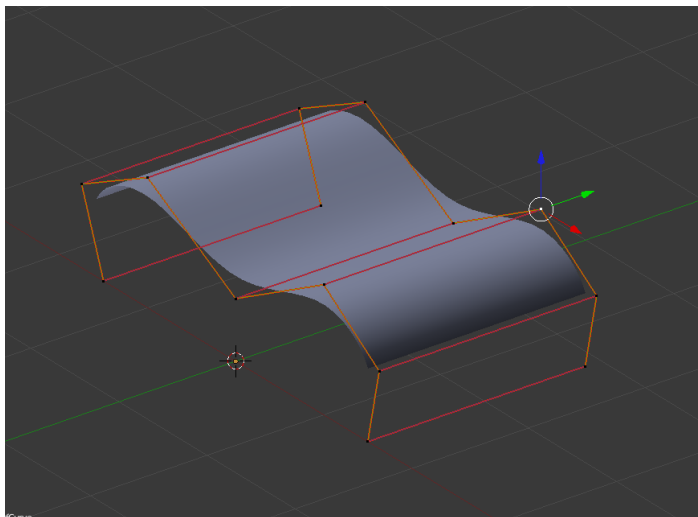


Staattisten objektien vastakohtana ovat dynaamiset objektit, eli kaikki mikä liikkuu ja animoidaan. Dynaamisia objekteja ovat usein hahmot, ajoneuvot ja tavarat joita pelaaja voi käyttää. Koska dynaamisissa objekteissa on usein animaatiota, on niiden tuotanto monimutkaisempaa. 3D-mallintamisen ja teksturoinnin lisäksi, objektit täytyvät myös rigata. Riggaamisessa objektiin tehdään mekanismi, jolla sitä voi liikutella animointivaiheessa. (LudoCraft 2006.)

Malleja tehdessä täytyy muistaa pelin tekniset rajoitukset, eli mallien yksityiskohtaisuus. Yleensä peliprojekteissa mallit saadaan näyttämään yksityiskohtaisemmilta hyvän teksturoinnin avulla, joka ei ole niin raskasta. (A Digital Dreamer 2012.)

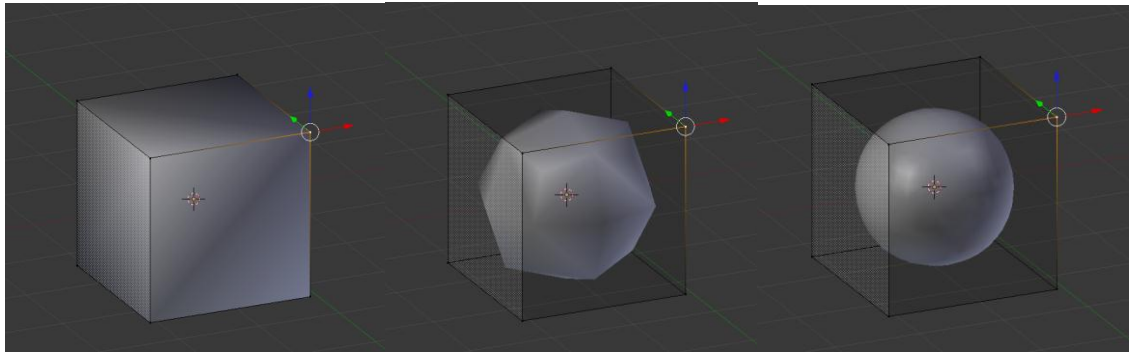
Suosituimpiin mallinnustekniikoihin kuuluvat NURB-mallinnus (Kuvio 8.), subdivision surface-mallinnus (Kuvio 9.), laatikkomallinnus eli box-modeling (Kuvio 10.) ja veistäminen eli sculpting (Kuvio 11.). (About.com d. 2012.)

NURBS, eli Non-Uniform Rational B-Splines on mallinnustapa, jonka toteutuksessa käytetään apuna Bézier-käyrää, jossa vähintään kahden pisteen avulla piirretään pisteiden välille käyrä. Tekniikkaa käytetään laajasti 3D-mallinnuksessa ja vektorigrafikassa. Mallinnuksessa mallin muotoa hallitaan pisteiden avulla. (RealWorld Graphics 2006.)



Kuvio 8. NURBS-mallinnusta Blender 3D:ssä.

Subdivision surface-mallinnustekniikassa tasoitetaan pohjameshin muotoa lisäämällä siihen lisää polygoneja. Subdivision surfacen vaikutusta voi myös lisätä, jolloin malliin tulee entistäkin enemmän pyöreyttä. (Holmes3D 2012.)



Kuvio 9. Subdivision surfacen vaikutus kuutioon. Oikeassa laidassa vaikutusta on lisätty niin paljon, että kuutio on tasoittunut palloksi.

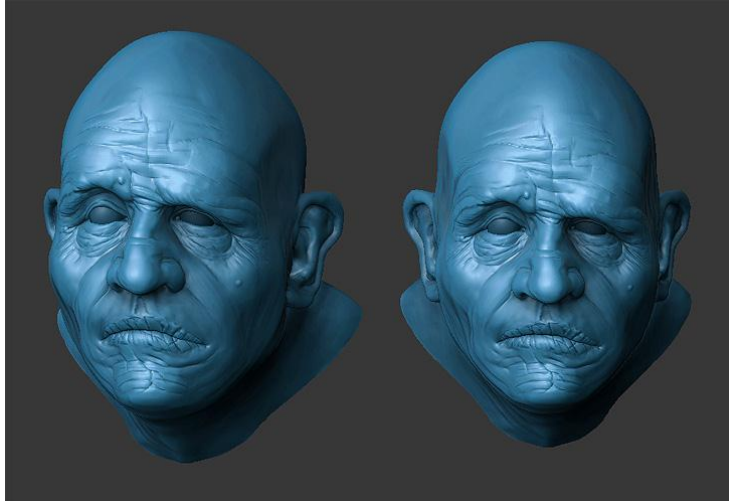
Laatikkomallinnuksessa malli luodaan muokkaamalla primitiivisiä muotoja kuten kuutiota. Tekniikka sopii erityisesti peliobjektien luontiin. Mallintaja voi lisätä helposti tarvittavan määrän yksityiskohtaisuutta objekteihin. Tekniikka on nopea ja helppo oppia, mutta monimutkaisten mallien luontiin se ei ole paras mahdollinen. (About.com d. 2012.)



Kuvio 10. 3D- pään laatikkomallinnus. (Allan Brito)

Veistämisestä on tullut viimeaikoina suosittu mallinnustekniikka. Tekniikka muistuttaa paljon patsaan veistämistä tai muovailuvahan käyttöä. Veistotekniikka ei onnistu kaikilla 3D-mallinnusohjelmilla, vaan sitä varten täytyy käyttää siihen erikoistunutta ohjelmaa, kuten ZBrushia. Tekniikalla voidaan tehdä malleja, joita on hyvin vaikea tai mahdoton tuottaa perinteisillä mallinnustekniikoilla. Tätä tekniikkaa on paras hyödyntää jos halutaan saavuttaa mahdollisimman yksityiskohtaista tai hyperrealistista jälkeä. Veistettyjä malleja ei voi suuren

yksityiskohtaisuuden takia käyttää peleissä suoraan, mutta sitä voidaan hyödyntää esimerkiksi tekstuureissa ja normal mapeissa. Normal mapping on tekniikka, jolla saadaan pienen monikulmiomäärän malleihin lisää yksityiskohtaisuutta, mukaillemalla objektin pinnan muotojen käyttäytymistä valon kanssa. (About.com 2012.)



Kuvio 11. Veistotekniikalla toteutettu 3D-pää. (Gathering 2012.)

Vuonna 2006, pelien 3d-hahmoissa oli keskimäärin 5 000-7 000 facea ja määrä kasvaa aina tietokoneiden kehittyessä. Mobiilipeleissä määrä on kuitenkin vielä todella pieni. Hahmossa on keskimäärin 500 facea. (Franson & Thomas 2007, 8).

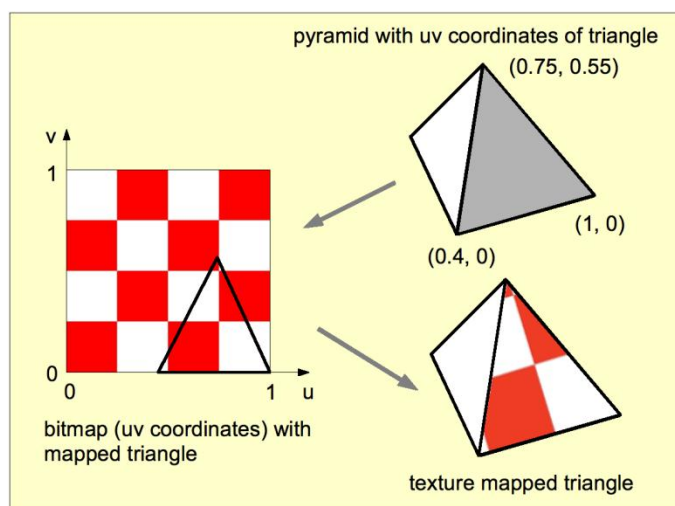
#### 4.4 Optimointi ja viimeistely

Kun 3D-malli on valmis, siihen täytyy tehdä mahdollisia korjauksia sen jatkokehitystä varten. Mallista etsitään reikiä, eli paikkoja jossa facet eivät ole yhdistyneenä toisiinsa. Tämä voi aiheuttaa renderöintiongelmia joissakin pelimoottoreissa. Optimointi auttaa myös minimoimaan kolmioiden määrän. Tarpeettomat verteksit yhdistetään toisiinsa ja yksittäiset verteksit poistetaan. Lisäksi joihinkin mallin kohtiin täytyy lisätä polygoneja takaamaan sulava liikkuminen animoitaessa. Näitä kohtia ovat polvet, kyynärpäät, hauis, olkapäät, kaula ja erityisesti kasvot. Kasvojen suurella polygonimäärällä saavutetaan sulava kasvoanimaatio. (Franson & Thomas 2007, 9).

## 4.5 UV-mappaus

Helpoin tapa ymmärtää UV-mappaus on ajatella, miltä auki leikattu pahvilaatikko näyttää. Pahvilaatikko on kolmiulotteinen objekti ja leikattuna sen muoto muistuttaa UV-mappia. Prosessissa muutetaan kolmiulotteinen objekti kaksiulotteiseksi kuvaksi ja kuvan värit kartoitetaan 3D-objektin pinnalle tekstuuriksi. UV-teksturoinnilla päästään parempaan lopputulokseen kuin proseduraalisilla materiaaleilla tai tekstuureilla ja se mahdollistaa yksityiskohtaisemman värityksen kuin pelkkä verteksen väritys. (BlenderWiki b 2012.)

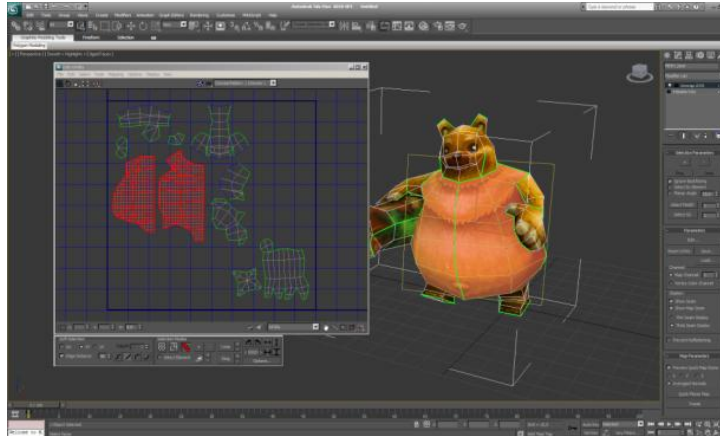
Kirjaimilla U ja V tarkoitetaan kaksiulotteisen tekstuurikartan koordinaatteja. U on horisontaalinen ja V vertikaalinen arvo, joita käytetään kolmiulotteisen koordinaatiston X, Y ja Z arvojen sijaan. Kun pahvilaatikko kootaan uudelleen siirretään U ja V koordinaatit oikeille paikoille kolmiulotteiseen koordinaatistoon. (BlenderWiki b 2012.)



Kuvio 12. UV-kartan koordinaatit. (blog.tartiflop)

UV-mappaus prosessin aikana 3D-malli on siis jaettu palasiin ”saumojen” kohdalta. Kun tekstuurikoordinaatit on määritetty UV-kartalle, ne eivät ole enää 3D-avaruudessa, eli niillä ei ole kolmatta ulottuvuutta. Tässä vaiheessa UV-kartasta voi ottaa kopion ja avata esimerkiksi Adobe Photoshopissa työstettäväksi. Kun teksturi on piirretty, voi sen avata jälleen 3D-mallinnusohjelmassa hahmon tekstuuriksi ja se kietoutuu hahmon ympärille UV-koordinaattien mukaisesti. UV-koordinaatiston muokkaaminen ei myöskään vaikuta 3D-mallin vertekseihin. (Franson & Thomas 2007, 10).

On tärkeää muistaa, että saumat joista UV-kartta leikataan, eivät ole näkyvillä. Jos sauma menee esimerkiksi keskeltä hahmoa, on vaarana että se näkyy tekstuurissa. Saumat kannattaa määrittää aina hahmon sivuille, jotta ne eivät ole näkyvissä pelaajalle. (Franson & Thomas 2007, 10).



Kuvio 13. Hahmon UV-mappaus 3D Studio Maxissa.

Joskus UV-kartta saattaa olla väärän muotoinen, joka aiheuttaa venymistä tekstuurissa. Tämän voi tarkistaa shakkilautatatekstuurilla, jonka ruutujen tasaisuus kertoo mistä kohti UV-mappaus on pielessä. (Franson & Thomas 2007, 10).

#### 4.6 Teksturointi

Tekstuurit ovat peleissä käytettäviä 2D-kuvia, joilla voidaan lisätä 3D-objekteille esimerkiksi yksityiskohtia, materiaaleja ja värejä. Dynaamiset ja staattiset 3D-objektit maalataan tekstuureilla, jotka noudattavat pelin visuaalista tyyliä. Dynaamisia objekteja teksturoidessa täytyy ottaa huomioon, että animaatio muuttaa 3D-mallien geometriaa, joten tekstuureiden täytyy olla mahdollisimman joustavia. (LudoCraft 2006.)



Kuvio 14. Teksturoidintia Adobe Photoshopissa

Pelien kehittyessä, hahmojen ja tavaroiden tekstuureilla on muitakin ominaisuuksia ja kerroksia kuin pelkkä tekstuurikartta. Näitä ominaisuuksia voi yhdistää antamaan tekstuurille esimerkiksi kiiltoa, heijastuksia tai läpinäkyvyyttä. Tekstuurit voivat sisältää myös bump mappoja tai normal mappoja. Näiden ominaisuuksien yhdistely saa pelihahmon näyttämään aidommalta ja nopeuttaa renderöimistä, toisin kuin käyttämällä suuren polygonimäärän malleja. (Franson & Thomas 2007, 28).

#### 4.7 Riggaus ja animaatio

3D-hahmon tuottamisen viimeinen vaihe on tehdä sille ”luut”, joilla voi liikuttaa mallia ja tehdä sille animaatioita. Hahmolle tehdään kokonainen luuranko, joka koostuu luista ja nivelistä, jotka vaikuttavat niiden ympärillä olevaan meshiin. (Franson & Thomas 2007, 15; About.com c 2012.)

Riggaamiseen kuuluu sopivan luurangon toteuttaminen hahmolle ja sen yhdistäminen 3D-malliin. 3DSMaxissa siihen käytetään skin modifieria ja Blenderissä tehdään yksinkertaisesti luurangosta hahmon aliobjekti, jolloin luut vaikuttavat ympärillä olevaan meshiin automaattisesti. On tärkeää, että luut liikuttavat vain niille määrättyjä alueita 3D-mallista. Jälkeenpäin on tehtävä mahdollisia muutoksia luihin ja 3D-malliin hyvän liikkuvuuden takaamiseksi. (Franson & Thomas 2007, 15).

Luiden asettaminen on ehkä riggausprosessin helpoin vaihe. Jotta riggaus toimisi hyvin, täytyy luiden ja nivelten noudattaa loogista hierarkiaa. Kun pelihahmon luurangon toteutus aloitetaan, ensimmäinen luu on hahmon kantaluu, josta voidaan hallita koko luurankoa.

Myöhemmät luut ovat yhteydessä kantaluuhun suoraan tai epäsuorasti muiden luiden kautta. Nivelet kannattaa sijoittaa samoihin kohtiin, kuin oikeassa luurangossa. (About.com c 2012.)

Pelihahmon kasvojen riggaus eroaa yleensä muista liikkumisen hallinnasta. Tavallisen nivel/luu struktuurin käyttäminen kasvojen riggauksessa on tehotonta ja hyvin vaikeaa, joten blend shapes-toiminnon käyttäminen on yleensä parempi vaihtoehto. Blend shapes-toiminnolla voidaan määrittää hahmolle valmiita kasvojen ilmeitä muokkaamalla esimerkiksi kasvojen verteksejä ja tallentamalla ilmeet käytettäväksi animoinnissa. Kasvojen riggaus on oma aiheensa ja eroaa paljon normaalista pelihahmon riggauksesta. (About.com c 2012.)

Kun hahmon luut ovat toimintakunnossa, on aika tehdä yksittäisiä animaatioseksenssejä, joita pelimoottori käyttää riippuen pelaajan toiminnasta pelissä. Tavallisella pelihahmolla voi olla jopa 50 - 100 eri animaatiota. Animaatioita voi tehdä manuaalisesti tai käyttäen liikkeen kaappausta. Loput animaatiot voidaan tehdä fysiikkamoottoreilla, kuten räsynuket. (Franson & Thomas 2007, 15).

Hahmojen animointiin käytetään aikajanaa, jonne asetetaan avainkehyksiä. Aikajanalta valitaan animaation kannalta merkitykselliset pisteet ja asetellaan hahmon tai objektin sijainti sekä asento sopivaksi. Tämän jälkeen sanotaan ohjelmalle, että piste on avainkehys ja ohjelma muistaa missä asennossa hahmo siinä hetkessä animaatiota on. Kun animaatio toistetaan, ohjelma interpoloi muutetut parametrit avainkehysten välillä ja toistaa liikkeen sulavana. Esimerkiksi jos tehdään animaatio, jossa hahmo kyykistyy, tehdään vain kaksi eri asentoa ja tehdään niille avainkehukset. Ohjelma tekee automaattisesti loput kehykset ja animaatiosta tulee sulavaa. (Animation Arena 2012; EuclideanSpace 2012).

Liikkeiden kaappauksessa nauhoitetaan ihmisenäyttelijöiden liikkumista erityisillä välineillä. Tämä tekniikka on todella kallista sillä se vaatii tilat ja kamerat, joilla liikettä nauhoitetaan. Mocap-tiedostoja voi käyttää mallinnusohjelmissa omissa 3D-hahmoissa. (Animation Arena 2012).

Jotkut pelit käyttävät fysiikkamoottoria lisänä animaatioille. Se mahdollistaa dynaamisen ja realistisen hahmojen liikkumisen. Tästä yleisin esimerkki on räsynukke-efekti, joka saa hahmon käyttäytymään kuollessaan kuin nukke joka tippuu lattialle. (Franson & Thomas 2007, 15).

Kun animaatiot ovat valmiina, ne otetaan käyttöön pelimoottorissa. Kun pelaaja painaa tiettyä nappia, otetaan tietty animaatiosekvenssi käyttöön pelissä. 3D-malli täytyy muuttaa myös pelimoottorin tukemaan tiedostomuotoon. (Franson & Thomas 2007, 15).



## 5 PELIHAHMON SUUNNITTELU

Hahmon suunnittelussa tehdään luonnoksia pelissä käytettävistä hahmoista. Pelihahmot ovat ihmisiä tai olentoja, joita pelaaja tai tekoäly ohjaa. Jotkut hahmot ovat pelin kannalta tärkeämpiä kuin toiset, mutta jokainen täytyy silti suunnitella alusta loppuun. (Pardew 2005, 10; Franson & Thomas 2007, 1).

Pelihahmon suunnittelun alussa täytyy miettiä, millainen peli on ja mikä tarkoitus hahmolla on siinä. Onko kyseessä toimintapeli, strategiapeli vai roolipeli? Kuvataanko peli ensimmäisestä vai kolmannelta persoonasta? Onko suunniteltava hahmo pelattava vai tekoälyn ohjaama? Millaisessa ympäristössä hahmo toimii? Onko kyseessä ihminen, eläin vai joku olio? Onko hahmolla aseita tai muita tavaroita? Nämä kysymykset auttavat luomaan onnistuneen pelihahmon, koska ne antavat esimakua hahmon ulkonäköön ja myös myöhemmin käytettäviin mallinnus- ja animaatiotekniikoihin. (Pardew 2005, 10; Franson & Thomas 2007, 1).

Pelin tyyli vaikuttaa melkein kaikkiin suunniteltavan hahmon komponentteihin. Esimerkiksi strategiapeleissä hahmot ovat kuvaruudulla niin pieniä, että myös 3D-mallien yksityiskohtaisuus ja tekstuureiden koko ovat myös pieniä. Tämä on tärkeää tiedostaa, jotta ei tee hahmoista liian monimutkaisia ja vältetään turhaa työtä. Jos peli on kuvattu esimerkiksi kolmannelta persoonasta, täytyy hahmojen olla yksityiskohtaisia ja panostaa enemmän animaatioihin. (Pardew 2005, 10; Franson & Thomas 2007, 1).



Kuvio 15. Strategiapelin kuvakulma. (Gameogre 2012.)

Jo suunnitteluvaiheessa on hyvä miettiä millaisia ominaisuuksia hahmoilla on ja miten se vaikuttaa toteutukseen. Jos suunniteltava hahmo on ihminen, on sen riggaaminen ja animointi yksinkertaisempaa, kuin esimerkiksi olion, jolla on kahdeksan jalkaa. Jos hahmolla on ase tai muu objekti kädessä, täytyy sen pystyä kantamaan sitä luonnollisesti, joka vaikuttaa 3D-mallin nivelten toteutukseen. Tavaroiden pitäminen vaatii myös erilaisia asentoja hahmolta. Mahdollinen taustatarina vaikuttaa myös pelihahmon ulkonäköön, liikkumiseen ja puheääneen. (Pardey 2005, 10; Franson & Thomas 2007, 1).

## 6 3D-GRAFIIKKAOHJELMAT

Grafiikkaohjelma on tietokoneohjelma, jota käytetään esimerkiksi kuvien ja muun visuaalisen multimedian tuotantoon. Pelituotannossa käytettäviin grafiikkaohjelmistoihin kuuluu kuvankäsittely- ja 3D-grafiikkaohjelmistot. (Webobedia 2012.)

3D-grafiikkaohjelmilla voidaan tuottaa 3D-malleja. Ohjelmat mahdollistavat 3D-mallien luonnin ja muokkauksen eri tavoilla, kuten lisäämällä tai vähentämällä polygonien määrää. Malleja voidaan myös pienentää, suurentaa tai venyttää. Ohjelmien käyttöliittymä mahdollistaa myös mallien tarkastelun eri kuvakulmista ja zoomaamisen. Markkinoilla on kymmeniä maksullisia ja ilmaisia mallinnusohjelmia. (Kioskea 2012.)

### 6.1 Blender 3D

Blender on ilmainen ja myös maailman suosituin avoimen lähdekoodin 3D-mallinnusohjelma. Blenderillä voi luoda laadukasta 3D-visuaalisuutta, still-kuvia ja jopa animaatioita. Koska se on ilmainen ja aktiivisesti päivittyvä, on se varteenotettava kilpailija ohjelmistomarkkinoilla. Jokainen Blenderin versio ladataan noin 200 000 kertaa. (BlenderWiki 2012.)

Blender sai alkunsa vuonna 1988, kun Ton Rosendaal oli perustamassa hollantilaista animaatiostudiota nimeltään NeoGeo. Ton oli vastuussa yrityksen sisäisestä ohjelmistokehityksestä ja myöhemmin päätettiin kehittää täysin uudet 3D-työkalut tyhjästä. Uudesta työkalusta syntyi ohjelma, joka tunnetaan nykyään Blenderinä. Vuonna 2002 Ton Rosendaal perusti Blender Foundationin, jonka tarkoituksena oli jatkaa Blenderin kehitystä ja markkinointia avoimen lähdekoodin ohjelmana. Projektiin saatiin sijoittajia ja lokakuussa 2002 Blender julkaistiin GNU-lisenssin alla. Blenderin kehitystä on jatkettu menestyksekkäästi tähän päivään asti. (Blender.org 2011.)

Blenderin ominaisuuksiin kuuluu laajat työkalut mallinnukseen, UV-mappaukseen, teksturointiin, riggaukseen sekä animointiin, jotka ovat tärkeitä pelien tuotannossa. Ohjelma sisältää myös monia muita työkaluja kuten erilaisia simulaatioita, renderöintiä ja jopa oman sisäisen pelimoottorin. Uusia ominaisuuksia tulee lisää melkein jokaisen version päivityksen yhteydessä. Sculpting, eli veistäminen on myös mahdollista Blenderillä jota voi tehdä yleensä

vain siihen erikoistuneilla ohjelmilla. Pienen kokonsa ansiosta ohjelman voi asentaa vaikka muistitikulle ja se pysyy aina mukana. (BlenderWiki 2012.)

Ohjelman lähdekoodi on GNU GPL lisenssin alla. GNU-lisenssin alla oleviin ohjelmistoihin pätee seuraavat asiat:

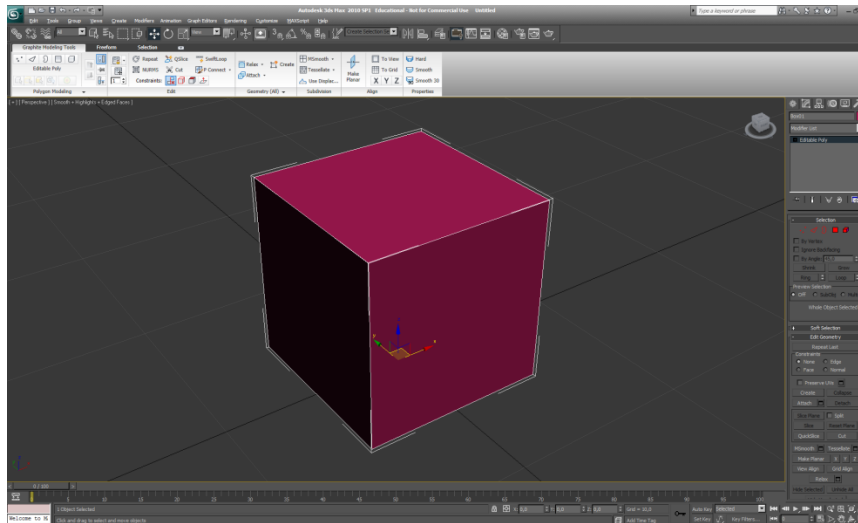
- ohjelmaa saa käyttää mihin vain
- ohjelmaa saa muokata ja muuttaa sen lähdekoodia
- ohjelmaa saa kopioida ja jakaa
- ohjelmaa saa parannella ja julkaista omia versioita

Lisenssin oikeuksien vastamaksuna käyttäjällä on velvollisuutena julkaistaessa ohjelmistoa seuraavat asiat:

- tekijän tulee toimittaa kopio GPL:stä ohjelman mukana, niin että vastaanottaja on tietoinen oikeuksistaan lisenssissä
- lähdekoodi täytyy olla mukana tai julkaista ilmaiseen jakoon
- jos haluat muokata koodia ja levittää muokattua versiota, sinun täytyy lisensoida muutoksesi GPL:n alla ja julkaista muutoksien lähdekoodit. (Et voi käyttää GPL koodia osana omaa ohjelmaa.)
- et voi rajoittaa ohjelman lisensointia GPL-ehtojen vastaisesti. (Et voi muuttaa GPL-lisensoitua ohjelmaa omaksi tuotteeksi). (BlenderWiki 2012; GNU 2012.)

## 6.2 3D Studio Max

Autodesk 3D Studio Max on suosittu 3D-mallinnusohjelma, joka on lähes standardi monessa peliyrityksessä, Tv-mainos -studioissa, arkkitehtuuria visualisoivissa yrityksissä sekä elokuvien erikoistehostestudioissa. Kuten Blenderissä, 3D Studio Maxissa on myös kattavat työkalut mallinnukseen, teksturointiin ja hahmojen animointiin. (Autodesk.com 2012.)



Kuvio 16. 3D Studio Max 2010:n käyttöliittymä.

Vuonna 1988 Autodesk alkoi kehittämään ohjelmaa nimeltään THUD, joka oli mallinnus- ja renderöintiohjelma. Nimi viittasi Thom Hudsoniin, joka oli projektin ainoa ohjelmoija. Projektiin liittyi myös Dan Silva, joka toi ohjelmaan avainkehysominaisuudet ja THUD muuttui 3D-animaatio-ohjelmaksi. Vuonna 1990 Autodesk julkaisi ohjelman nimellä 3D Studio, joka oli ensimmäinen edullinen 3D-mallinnusohjelma PC:lle. Ohjelman julkaisuhinta oli 3495 dollaria, joka ei ole siitä koskaan noussut. 3D Studio oli merkittävä myös työkalujen monipuolisuuden ansiosta. 3D Studio-ohjelman nimi muuttui myöhemmin 3D Studio Maxiksi ja ohjelmasta on julkaistu uusi versio joka vuosi. (Area 2012.)

Ohjelma ei ole ilmainen, joten jokainen julkaiseva peliyritys joutuu maksamaan vaadittavan lisenssimaksun. Kaupallisen lisenssin korkean hinnan takia, Autodesk tarjoaa myös opiskelijaversioon, joka on tarkoitettu vain ohjelman opettelemiseen. (Autodesk.com 2012.)

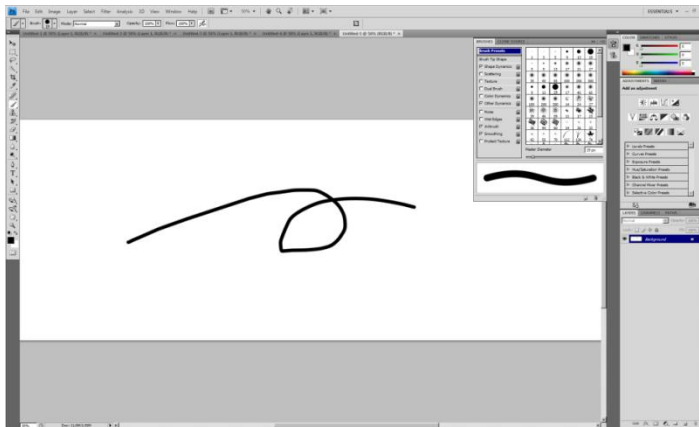
## 7 2D-GRAFIKKAOHJELMAT

2D-grafiikkaohjelmistoilla tuotetaan kaksiulotteista digitaalista grafiikkaa, kuten tekstiä, illustraatioita ja kuvien muokkaamista.

### 7.1 Adobe Photoshop

Photoshop on Adoben kehittämä kuvankäsittelyohjelma, jolla on kaupallisessa käytössä markkinajohtajuus. Ohjelma sisältää laajan valikoiman erilaisia työkaluja kuvien muokkaamiseen ja myös kuvien piirto-ominaisuudet ovat monipuoliset. Eri versioiden myötä on julkaistu myös uusia innovaatioita, joita monet muutkin ohjelmistot käyttävät. (Adobe 2012.)

Photoshopin kehitys alkoi vuonna 1987, kun Thomas Knoll teki ohjelman nimeltään Display, jonka tarkoitus oli näyttää harmaasävyiset kuvat selkeämmin mustavalkoisilla monitoreilla. Thomaksen veli työskenteli Industrial Light & Magic -erikoistehostestudiossa ja ehdotti, että ohjelma muutettaisiin täysiveriseksi kuvankäsittelyohjelmaksi. Vuonna 1988 ohjelman nimi muutettiin ImagePro:ksi ja siinä oli ominaisuuksia niin paljon, että veljekset päättivät markkinoida sitä kaupallisesti. BarneyScan skanneriyritys osti ohjelman lisenssin väliaikaisesti ja ohjelmaa levitettiin nimellä BarneyScan XP. Sopimuksen loputtua Adobe osti ohjelman myyntioikeudet ja vuonna 1990 julkaistiin Photoshop 1.0. (LowEndMac 2008.)

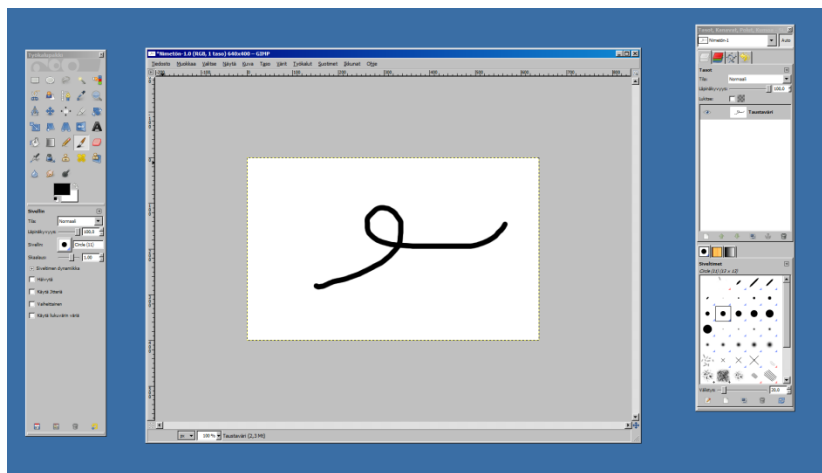


Kuvio 17. Adobe Photoshop CS4.

Photoshop on siis hyvä työkalu peliyrityksen graafikoille, koska sillä voi tuottaa esimerkiksi konseptitaidetta, kaupallisia piirroksia, grafiikkaa suoraan peliin tai tekstuureja 3D-malleihin. Se tukee myös monia eri kuvatiedosto-formaatteja. Ohjelman suosiosta kertoo mm. termi ”photoshoppaus”, jolla tarkoitetaan yleisesti kuvien digitaalista muokkausta. (Adobe 2012.)

## 7.2 GIMP

GIMP (GNU Image Manipulation Program) on ilmainen ja avoimen lähdekoodin kuvankäsittelyohjelma. Se suunniteltiin Adobe Photoshopin ilmaiseksi korvaajaksi ja sisältää myös kuvankäsittelyyn sekä piirtämiseen tarvittavat työkalut. Photoshopin tavoin myös GIMP-ohjelmaa on mahdollisuus laajentaa PlugIneillä, eli ohjelmistolaajennuksilla. Ohjelmaa on saanut näkyvyyttä myös ammattilaisten piireissä. (Gimp.org a 2012.)



Kuvio 18. GIMP 2.6.8.

GIMP:n on luonut kaksi opiskelijaa, Spencer Kimball ja Peter Mattis, jotka halusivat ohjelmoida kuvankäsittelyohjelman, vaikka kummallakaan ei ollut kokemusta graafisesta suunnittelusta. He saivat projektiinsa tukea myös opettajilta ja ohjelma sai nimekseen General Image Manipulation Program, eli lyhennettynä GIMP. Ohjelma julkaistiin vuonna 1996 ja siinä oli monia toimintoja, kuten tuki Plug-ineille, joiden avulla kehittäjät voivat tehdä ohjelmaan omia laajennuksiaan. (Gimp.org b 2012.)

## 8 PETS VS ORCS PELIPROJEKTI

Käytännön osuudessa tarkastellaan pelin 3D-hahmojen tuotantoprosessia, sekä vertaillaan eri kuvankäsittely- ja mallinnusohjelmia. Ohjelmien vertailun tarkoituksena on selvittää onnistuuko pelihahmon toteutus myös ilmaisohjelmistoilla. Tämä on hyödyllistä peliyritykselle jolla ei ole varaa maksaa kalliita kaupallisia lisenssejä.

Grafiikan tuotannossa käytettiin 3D Studio Max 2012 –mallinnusohjelmaa sekä Photoshop CS 5 –kuvankäsittelyohjelmaa. 3D Studio Maxia voidaan verrata Blenderiin ja tutkia mitä eroja on maksullisella ja ilmaisella ohjelmalla.

Kuvankäsittelypuolella voidaan verrata Photoshop- ja GIMP-kuvankäsittelyohjelmaa, miten ne soveltuvat esim. teksturointiin ja kuinka hyviä piirtotyökalut ovat. Ohjelmista tulee löytyä myös tarvittavat tuet eri kuvaformaateille.

Käytännössä jokaiseen suunnitelmassa mainittuun ohjelmaan perehdytään ja testataan samat vaiheet läpi, kuin Pets vs Orcs- projektin hahmojen tuotannossa. Työvaiheiden esimerkkikuvissa ei aina käytetä Pets vs Orcs- pelin hahmoja, mutta niiden toteutustapa on identtinen.



Kuvio 19. Pets Vs Orcs aloitusruutu.



## 8.1 Tietoa pelistä

Pets Vs Orcs oli free-to-play –peli, eli se oli täysin ilmainen ja kaikki pelin tuotto tulee sen sisäisistä mikromaksuista. Itse pelin genreä voitaisiin kuvailla strategia/kylänrakenteluksi, koska siinä oli tarkoituksena rakentaa omaa kylää ja taistella örkkejä vastaan eri kentissä. Taistelukentässä on sekä örkkien, että eläinten tukikohdat ja tavoitteena on saada tuhottua örkkien tukikohta. Taistelukentälle voi asettaa kylänrakentelussa koulutettuja eläimiä taistelemaan örkkien joukkoja vastaan. Kun eläimet on asetettu taistelukentälle, niitä ei voi hallita vaan ne taistelevat omatoimisesti. Taistelussa pärjääminen riippuu siitä, millaisia hahmotyyppejä laittaa örkkien hahmoja vastaan.

Kylänrakentelussa tehdään eläinarmeijalle kylä, jossa voi resursseja käyttämällä kouluttaa taistelussa käytettäviä eläimiä sekä avata uusia hahmoluokkia. Tiettyjä rakennustyypppejä rakentamalla saa myös lisää pelissä käytettävää valuuttaa. Pelin sisäistä valuuttaa voi ostaa lisää oikealla rahalla, joka on suosittu free-to-play –peleissä käytetty tapa saada siitä tuottoa,

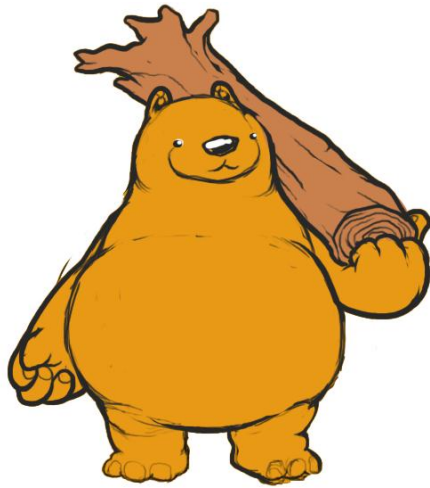
## 9 HAHMON TUOTANTO

Toimivan 3D-hahmon tuotanto peliin vaatii osaamista monelta eri osa-alueelta, kuten 3D-mallinnuksesta, 2D-grafiikasta ja animaatioista. Käytännön osuudessa keskitytään hahmon tuotantoprosessiin mobiilipeliin ja verrataan ohjelmistoja.

### 9.1 Suunnittelu ja konseptitaide

Hahmot suunnitellaan noudattaen pelin tyyliä ja genreä. Projektiamme varten suunnittelimme hahmoista värikkäitä ja tyyliltään piirroselokuvamaisia, joka sopii hyvin mobiilipelimarkkinoille ja laajalle kohderyhmälle. Suunnittelun aikana otettiin myös huomioon, että pelitilanteessa hahmot ovat ruudulla hyvin pieniä, joten niiden muodot täytyy olla erityisen selkeitä. Esimerkiksi örkkihahmon pää on muuhun vartaloon verrattuna tavallista suurempi, joka erottuu hyvin jopa kaukaisesta näkymästä. Suunnitteluun käytetyt luonnokset ja konseptitaide piirrettiin Adobe Photoshop–kuvankäsittelyohjelmalla. Kun hahmon ulkomuoto oli lopullinen, siitä tehtiin myös pohjapiirrokset auttamaan 3D-mallinnusta. Mahdolliset muutokset tehtiin jälkikäteen lopulliseen 3D-malliin.

Pets vs Orcs-pelissä erilaiset eläinhahmot sotivat örkkejä vastaan, joten suunnittelimme sitä varten neljä erilaista eläin- ja örkkihahmoa. Kaikki hahmoista tehdyt piirrokset koottiin yhteen ja niistä valittiin parhaimmat toteutettavaksi. Halusimme hahmoista mahdollisimman persoonallisia sekä muodoiltaan runsaita, joka auttaa myös hahmoluokkia erottumaan toisistaan pelitilanteessa.



Kuvio 20. Pvo-karhun konseptitaidetta.

Suunnitteluvaiheessa ohjelmien eroilla ei ole suurta merkitystä, koska kuvat voivat olla vain karkeita hahmotelmia tai perinteisesti lyijykynällä piirrettyjä. Myös suunnittelijan henkilökohtaiset taidot vaikuttavat tähän vaiheeseen suuresti ja se tekee ohjelmien vertaamisesta vaikeampaa. Tämä vaihe on mahdollista toteuttaa ilman mitään maksuja. Ainoa merkittävä ero suunnittelussa voi olla kuvankäsittelyohjelmien piirtämiseen käytettävien siveltimien muokkausmahdollisuus, joka on Photoshopissa hieman monipuolisempaa ja se tukee myös useampaa eri tiedostomuotoa.



Kuvio 21. Lopullinen pelihahmo. Hahmossa on selviä eroja konseptitaitteeseen.

## 9.2 3D-mallinnus

Hahmosta tehtyjen suunnitelmien pohjalta tehdään pelissä käytettävä 3D-malli. Mallinnusvaiheessa toteutetaan myös kaikki hahmon teksturointiin ja animointiin vaadittavat asiat, kuten UV-mappaus ja riggaus. 3D-mallin yksityiskohtaisuudessa noudatetaan alustan rajoituksia ja rakenteessa otetaan huomioon raajojen hyvä liikkuvuus. Mobiilipelejä varten, mallin polygonien lukumäärä pyritään pitämään mahdollisimman pienenä. Yksityiskohtien rajoitus tuo uusia haasteita täydellisen muodon ja hyvän teksturoinnin saavuttamiseksi.

3D-mallinnusvaiheessa, eri ohjelmien väliset erot voivat olla hyvin pieniä ja merkityksettömiä, koska 3D-mallit koostuvat kaikissa samoista komponenteista. Suurin ero ohjelmistoissa on käyttöliittymän sopivuus mallintamiseen, hyödyllisten työkalujen määrä ja yhteensopivuus muiden ohjelmistojen välillä.

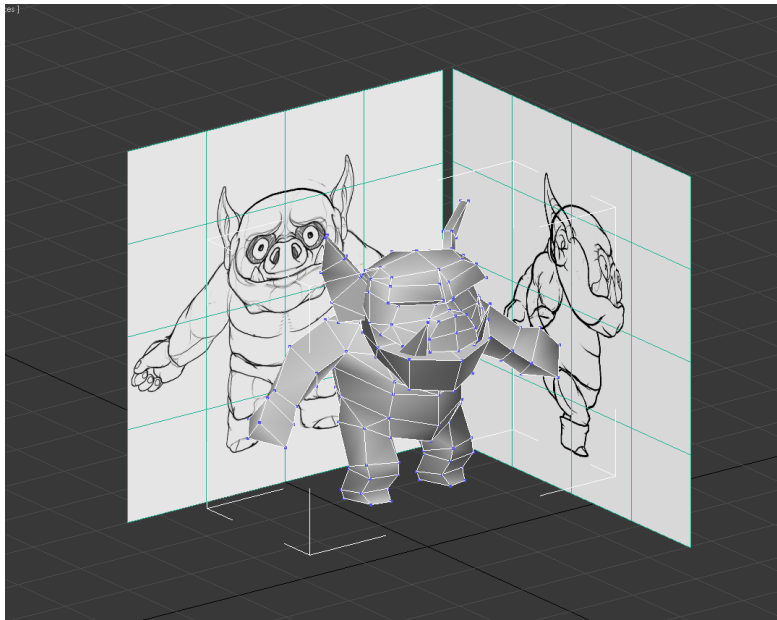
### 9.2.1 Mallinnusprosessi

3D-mallinnusohjelmassa luodaan 3D-malleja muokkaamalla niiden verteksejä, faceja ja edgejä, joista mallit koostuvat. Mallin yksityiskohtaisuus riippuu siitä, kuinka monta polygonia se sisältää. Useimmat 3D-mallinnusohjelmat sisältävät myös omat renderöinti- ja animaatiotyökalut, mutta mobiilipelien toteutuksessa hyvät mallinnus-, teksturointi- ja animaatiotyökalut ovat tärkeimmät.



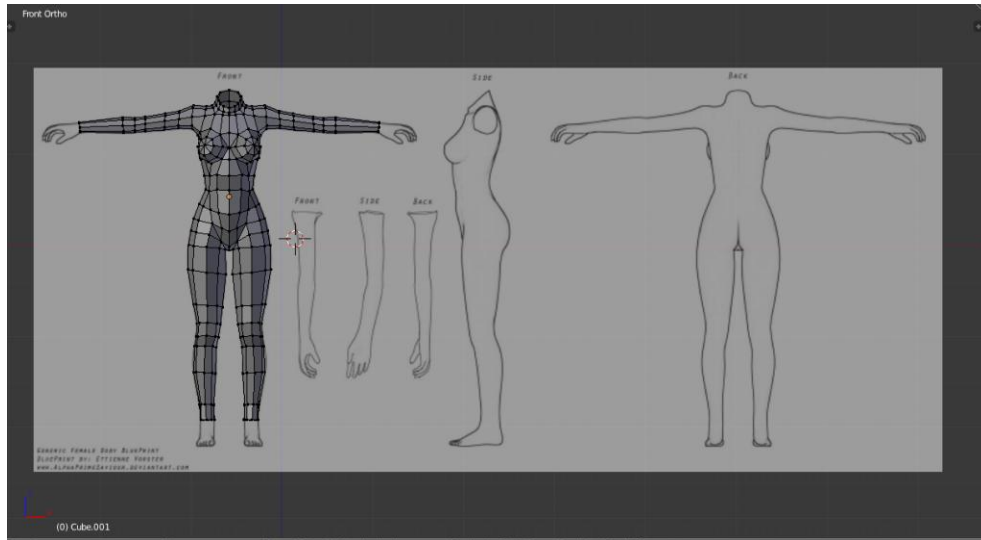
Kuvio 22. Örkkihahmon mallinnusta helpottavat suunnitelmat.

Mallinnusvaiheen alussa otetaan ohjelmassa käyttöön mahdolliset pohjapiirrokset (Kuvio 25.). Sekä Blenderissä, että 3D Studio Maxissa on mahdollisuus asettaa hahmosuunnitelmat näkyville ohjelman 3D-näkymän taustalle. Tämä auttaa nopeuttamaan mallinnusta ja noudattamaan hahmoista piirrettyjä suunnitelmia. 3DSMaxissa pohjapiirrosten valmistelu tapahtuu asettamalla kaksi yhdestä polygonista koostuvaa tasoa mallin ympärille. Toinen taso osoittaa mallin eteen ja toinen sivulle. Eteenpäin osoittavaan tasoon lisätään materiaali ja tekstuuriksi mallinnettavan hahmon kuva edestä. Vastaavasti toiseen tasoon laitetaan hahmon kuva sivulta. Kun 3DSMaxin kuvakulman näkyväksi asettaa sivu- tai etunäkymän, pystytään tarkkailemaan pohjapiirroksia mallinnuksen aikana (Kuvio 23.). On tärkeä ottaa huomioon että pohjapiirrokset ovat näkyvässä samankokoisia suhteessa toisiinsa.



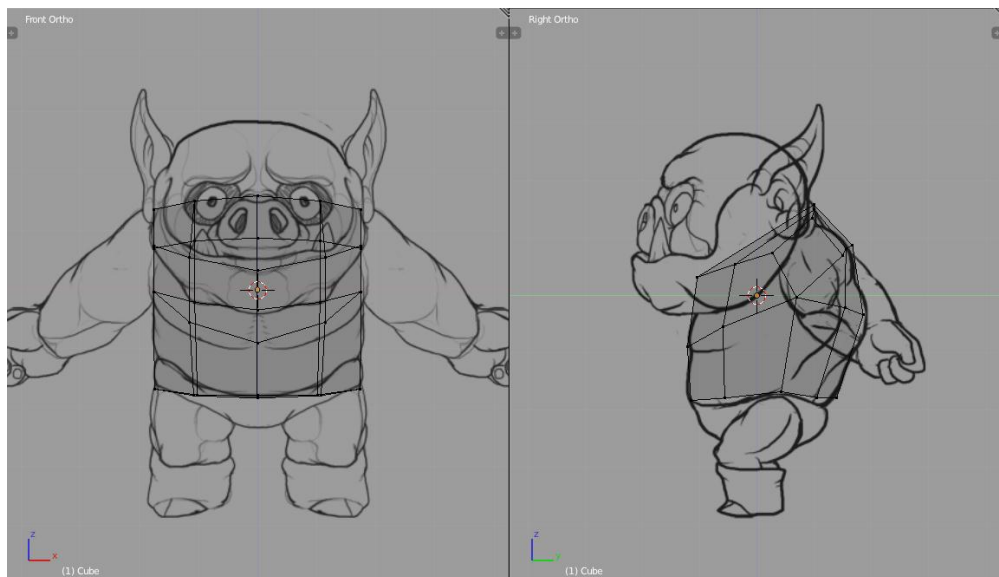
Kuvio 23. Pohjapiirrosten asettelu 3D Studio Maxissa.

Blenderissä saman asian voi saavuttaa esimerkiksi jakamalla 3D-näkymän kahteen osaan. Toinen kuvaa hahmoa edestä ja toinen sivulta. Molempiin näkymiin asetetaan taustakuvaksi hahmon pohjapiirrokset (Kuvio 24.). On myös mahdollista käyttää vain yhtä 3D-näkymää ja asettaa malli etu- ja sivukuvakulmissa oikeaan kohtaan pohjapiirrosta, mutta tämä tekniikka vaikeutuu jos pohjapiirroksessa on enemmän kuin kaksi kuvakulmaa, koska mallia joudutaan liikuttamaan.



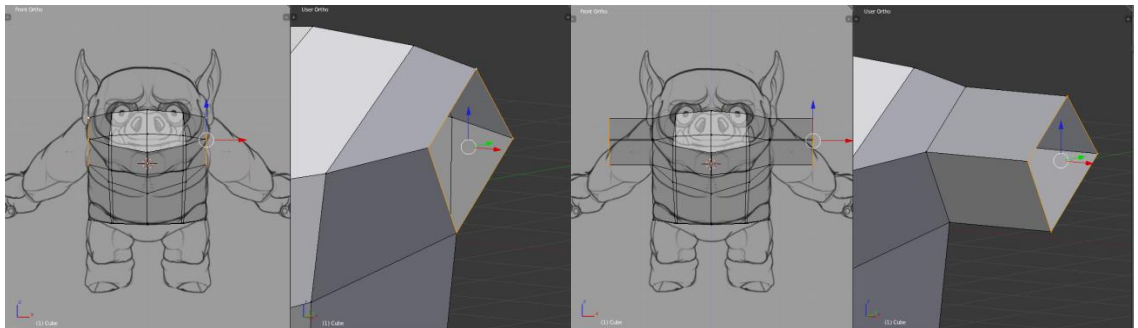
Kuvio 24. Pohjapiirroksen asettelu Blender 3D:ssä.

Tässä työssä kaikki mallit on tehty käyttäen laatikkomallinnustekniikkaa, eli hahmo mallinetaan valmiista 3D-laatikosta, johon lisätään tarvittava yksityiskohtaisuus. Laatikkomallinnus on tarkka ja säästeliäs tapa, joka sopii täydellisesti pelihahmoille. Sillä saa nopeasti tehtyä hahmon, joka on juuri suunnitelmien mukainen ja säästää polygoneja enemmän kuin muut tekniikat. Kuution koko ja verteksit asetetaan vastaamaan taustalla olevaa hahmosuunnitelmaa. Tarvittavia yksityiskohtia saadaan malliin esimerkiksi lisäämällä verteksejä. Kuutiosta voidaan aluksi tehdä hahmon torso, josta on helppo jatkaa raajoihin ja kaulaan.



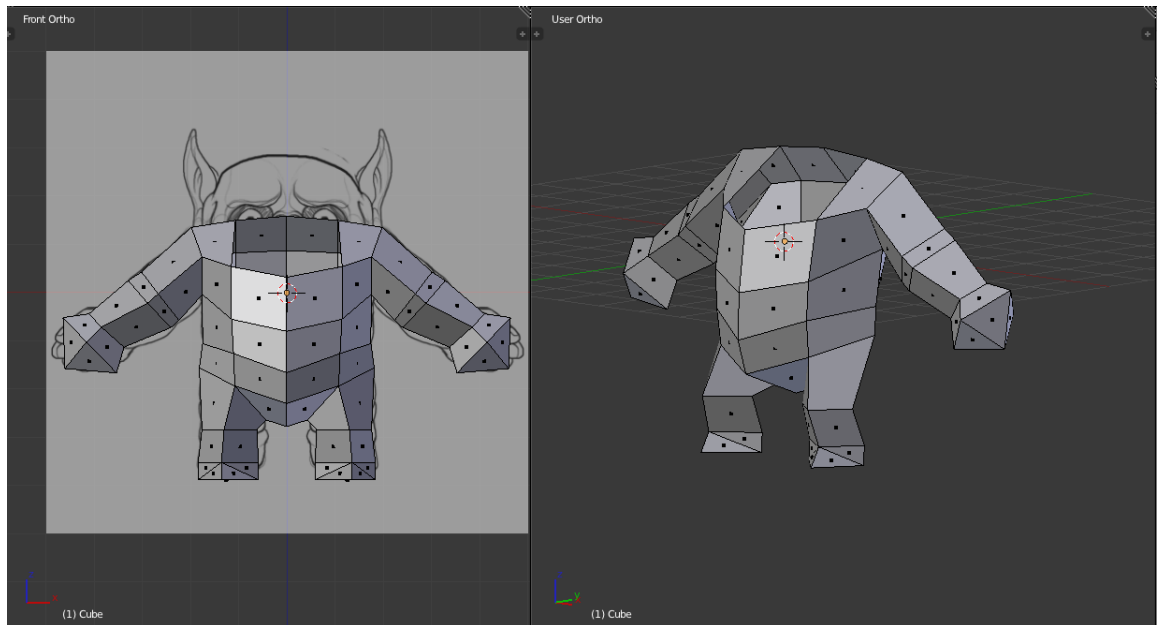
Kuvio 25. Mallinnuksen hyvät alkuasetelmat. Taustalla hahmon pohjapiirroksat ja malli, jossa mirror modifier. Mallinnus on aloitettu kuutiosta, johon on lisätty jo hieman yksityiskohtia.

Jos hahmon muoto on symmetrinen, toisen puolen erikseen mallintaminen on turhaa. Symmetrisyyden säilyttämiseksi voidaan käyttää useimmista ohjelmista löytyvää mirror-modifieria, joka luo hahmon toisen puolen samanlaisiksi automaattisesti. Mirror-modifier-työkalu otetaan käyttöön poistamalla toinen puoli hahmosta ja asettamalla se kopioimaan hahmon muotoa tietyllä akselilla. Kun työkalu on päällä, ohjelma näyttää kokonaisen hahmon vaikka todellisuudessa mallinnetaan vain toista puolta. Mirror-modifier löytyy Blenderin ja 3DSMaxin lisätyökalulistoista. 3DSMaxissa mirror-modifieria vastaa symmetry-modifier, koska sen mirror-modifierilla on hieman erilainen käyttötarkoitus. Kun mallista on poistettu puolet, valitaan modifier ja määritetään akseli, jonka suunnassa mallin peilikuva syntyy.



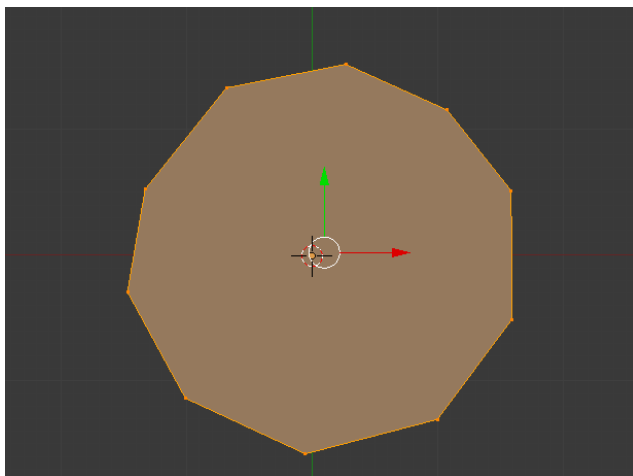
Kuvio 26. Extrude-työkalun käyttö hahmon raajojen toteutuksessa.

Hahmon raajojen luomiseksi, voidaan torsoon jättää aukot, joista raajat saadaan esiin käyttämällä extrude-ominaisuutta (Kuvio 26.). Extrude-työkalulla voidaan kopioida haluttu osa mallia ja kopioiden välille syntyy face. Hahmojen käsiä ja jalkoja mallintaessa täytyy ottaa huomioon niiden liikkuvuus. Väärällä tavalla mallinnetut nivelet saattavat animoitaessa vääristää hahmon muotoja. Liikkuvien osien ympärille täytyy lisätä tarpeeksi polygoneja, jotta se kestää liikkumisen. Nivelten kohdalla täytyy olla siis vähintään yksi verteksisilmukka ja liikkuvuutta voi parantaa lisäämällä niitä useampia. Hahmojen pää mallinnettiin erillisestä laatikosta, samalla tekniikalla ja yhdistettiin lopulta torsoon yhdistämällä verteksit.



Kuvio 27. Hahmon ruumis on melkein valmis. Pää tehdään erillisestä kuutiosta.

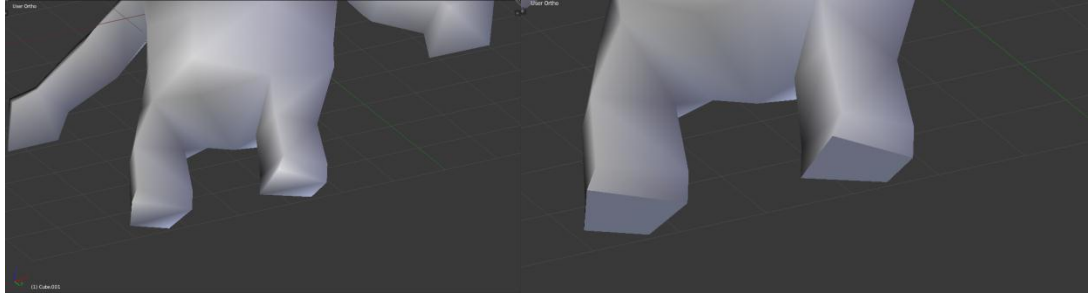
Blenderin rajoituksena on aina ollut, että enintään neljä verteksiä voi olla yhden facen ympärillä, toisin kuin 3D Studio Maxissa, joka on ollut kenties mallinnuksen suurin ero ohjelmissa. Mutta tähänkin on tullut muutos Blenderin uusimpien versioiden myötä ja yhdellä pinnalla voi olla rajoittamaton määrä verteksejä. Tämän muutoksen myötä knifetoolista on tullut myös hyödyllisempi ja 3DSMaxin kaltainen työkalu. Pelien tuotannossa täytyy muistaa, että hahmot muutetaan moottorissa kolmioiksi ja useamman kuin neljän verteksin pinnat täytyy rajata leikkaustyökalulla, että se jakautuu kolmioiksi halutulla tavalla, eikä aiheuta ongelmia esimerkiksi liikkussa.



Kuvio 28. Esimerkki facesta, jossa enemmän kuin 4 verteksiä.

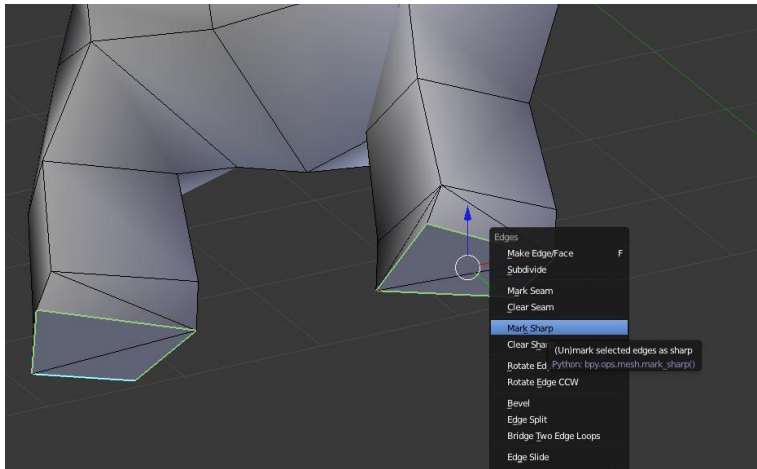


Mallien viimeistelynä voidaan määrittää jotkin reunat teräviksi. Tällä tavoin voidaan korostaa joitakin muotoja ja välttää liikaa pyöreyttä. Tämä voidaan toteuttaa esimerkiksi Blenderin edgesplit työkalulla, mutta se jakaa verteksit kahteen osaan, jolloin malliin jää päällekkäisiä verteksejä. 3D Studio Maxin vastaava työkalu smoothing groups on tuettuna Unityn kanssa ja soveltuu siis paremmin tarkoitukseen.



Kuvio 29. Hahmon jalkapohjien edget on määritetty teräviksi.

Smooth groups työkalulla voi jakaa facet eri numeroihin. Samannumeroiset facet tasoittuvat toisiinsa ja erinumeroisten väliin tulee terävä reuna.

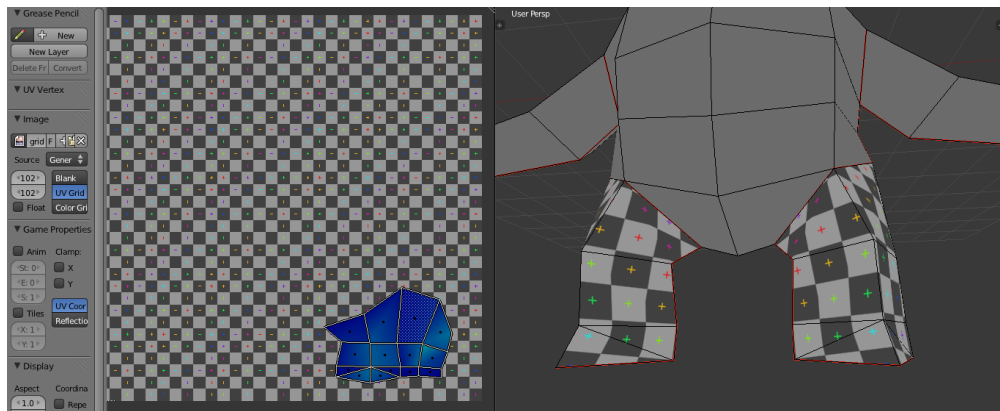


Kuvio 30. Blenderissä reunojen hallinta löytyy painamalla ctrl+e.

### 9.3 UV-mappaus

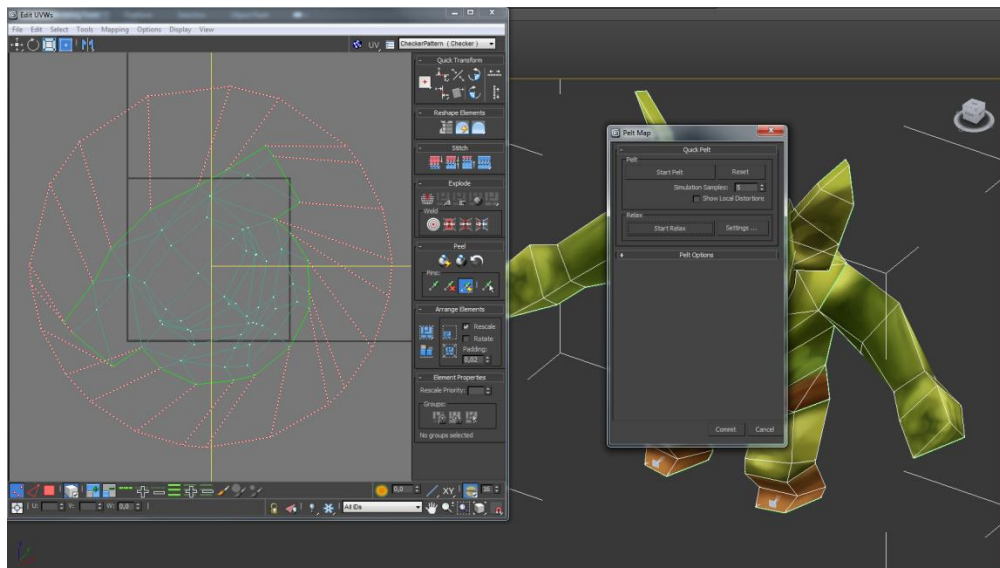
Ennen UV-mappauksen aloittamista on suunniteltava, mitkä osat hahmon tekstuurista ovat symmetrisiä. Kuten 3D-mallinnuksessa, symmetriset osat UV-mapataan vain kerran ja kopioidaan esimerkiksi mirror modifier -työkalun avulla, joka nopeuttaa tekstuurien

piirtämistä huomattavasti. Pets Vs Orcs -projektissa käytimme tätä tekniikkaa hahmojen raajoissa. Päästä ja torsoa ei peilattu, jotta niihin voidaan piirtää epäsymmetrisiä kuvioita.



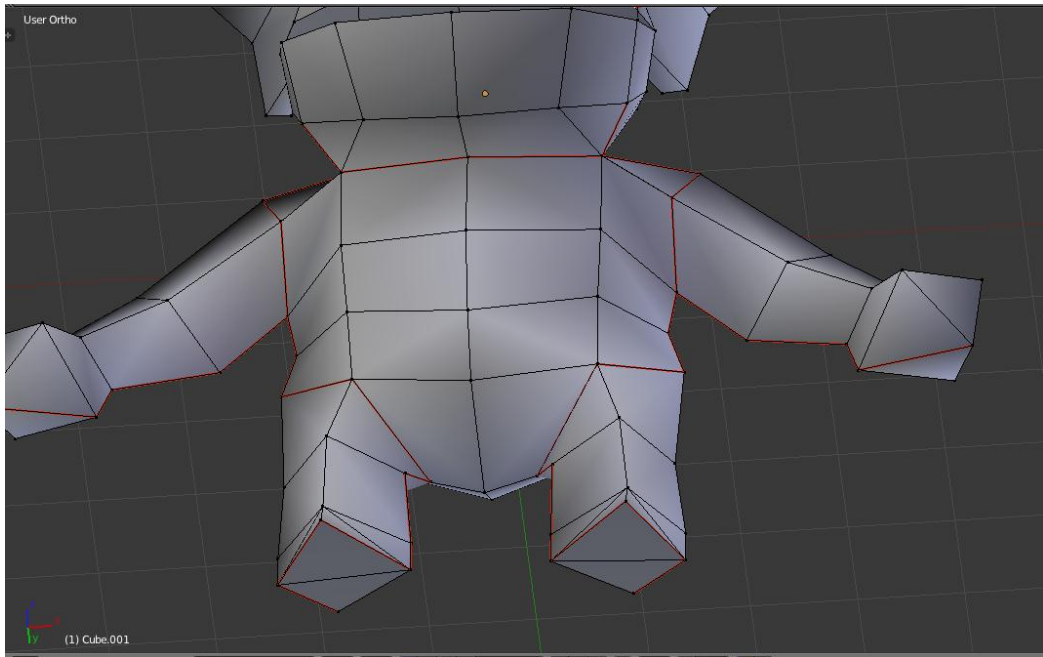
Kuvio 31. Blender 3D: UV-mappaunäkymä.

UV-mappauksessa täytyy tietyistä mallin edgeistä määrittää sauma, jotka toimivat samalla periaatteella kuten oikeat saumat vaatteissa (Kuvio 35.). Hahmon käsivarret ovat muodoltaan putkimaisia, joten niihin on määritetty pitkittäin menevät saumat, josta UV-map menee poikki. Näin käsivarren UV-map saadaan levitettyä ja siihen on helppo piirtää tekstuuri. Sama tapa toimii myös jalkoihin. Kun kädet ja jalat on kartoitettu, saadaan niistä symmetrinen kopio mirror modifierin avulla ja toinen puoli kopioituu entisen UV-mapin päälle (Kuvio 31.).



Kuvio 32. 3D Studio Maxin Relax-työkalu toiminnassa.

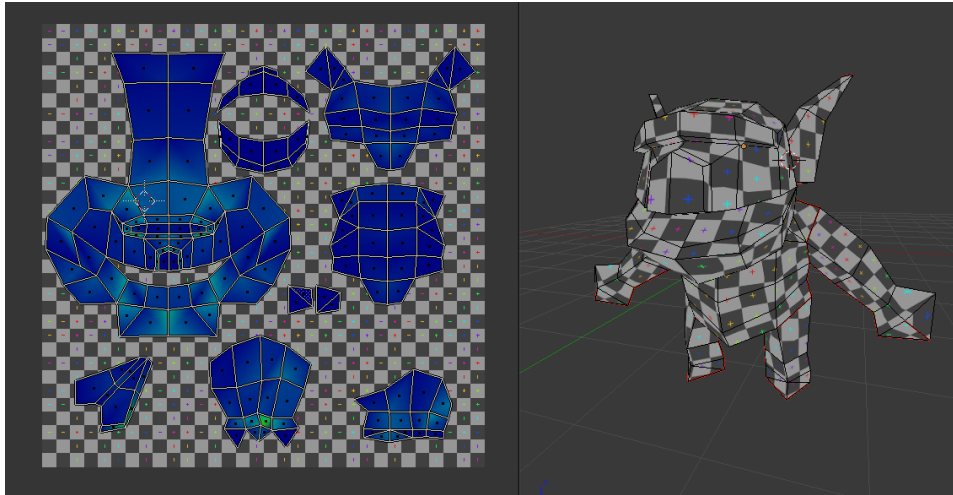
Hahmon torso voidaan mapata helposti jakamalla se etu- ja takapuoliin. Erityisesti orgaanisten hahmojen UV-mappauksessa täytyy ottaa huomioon pyöreät muodot ja ettei niiden tekstuuri veny liikaa. Venymisen voi tarkistaa ohjelmista löytyvällä shakkilautatekstuurilla. Kun tekstuurin ruudut ovat mahdollisimman neliön muotoisia, UV-map ei veny. Blenderistä löytyy myös erityinen työkalu, joka visualisoi kartoituksen venymisen väriarvoilla. 3DSMaxissa on erityisesti orgaanisten mallien kartoitukseen relax-työkalu (Kuvio 32.), joka auttaa säilyttämään muodot sekä vähentämään venymistä.



Kuvio 33. UV-kartoituksessa käytettävät saumat näkyvät mallissa punaisella.

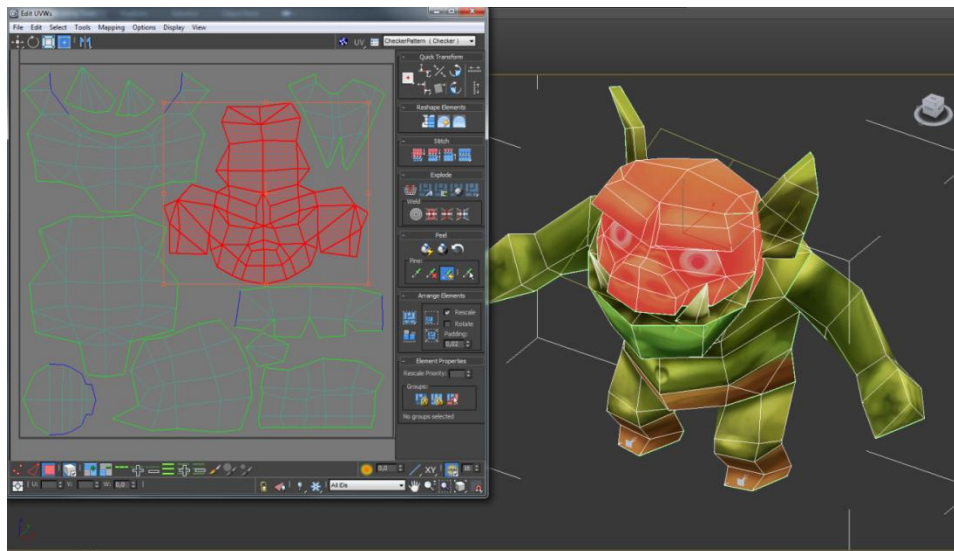
Kun kaikki hahmon osat ovat UV-kartassa, palaset täytyy asetella sopivaan järjestykseen teksturoijalle. Jos halutaan esimerkiksi kasvoihin suurempi resoluutio, eli pikseleiden määrä, voidaan sitä skaalata suuremmaksi UV-kartalla.

Blenderissä UV-kartoitustyökalut saadaan näkyviin editor type -valikosta, valitsemalla UV/image-editor. 3DSMaxissa UV-kartoitustyökalut löytyvät modifier-valikosta, jolloin aukeaa erillinen ikkuna.



Kuvio 34. 3D-malli on kokonaan UV-mappattu ja valmiina teksturoitavaksi.

Prosessin lopuksi UV-kartasta tallennetaan mallikuva, jota käytetään pohjana kuvankäsittelyohjelmassa. Kuvan koon ja tiedostomuodon voi itse määrittää halutuksi. Mallikuvan ansiosta hahmon tekstuurit voidaan piirtää suoraan oikeisiin kohtiin.

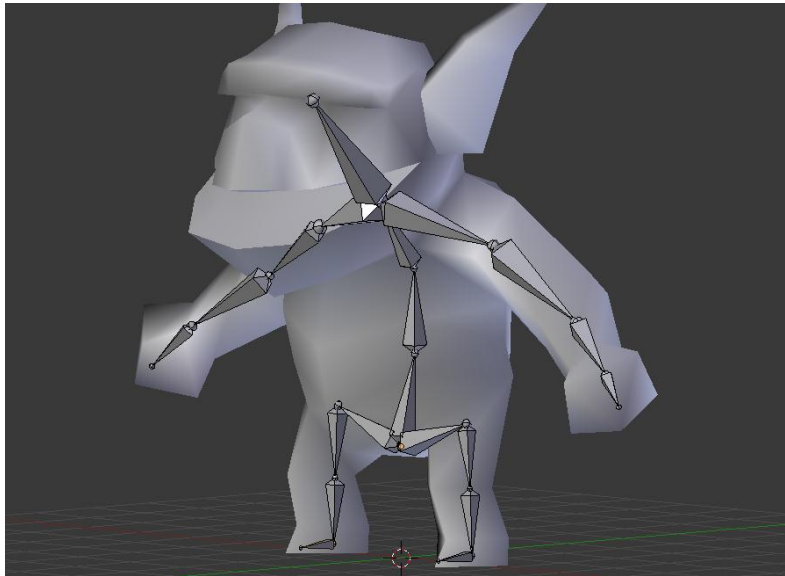


Kuvio 35. 3D Studio Maxin UV-mappausnäkyvä.

#### 9.4 Animaatio ja riggaus

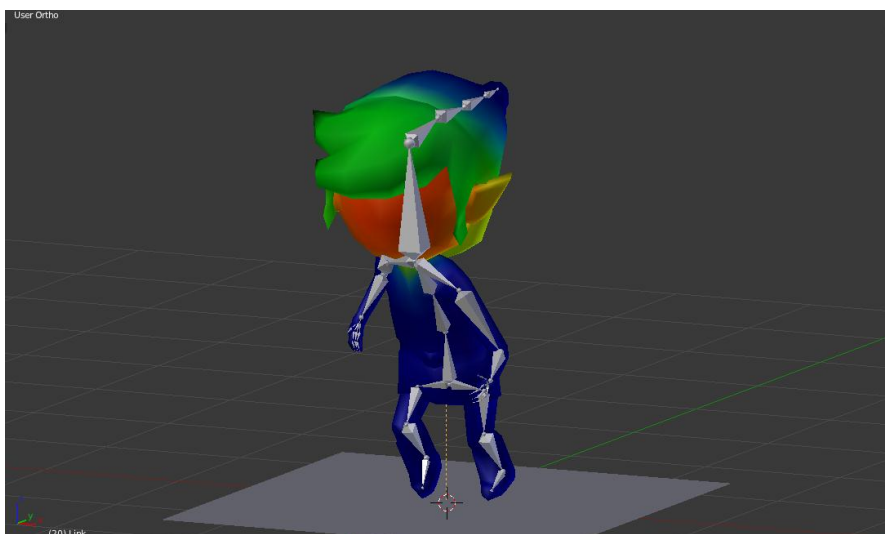
Tässä projektissa hahmoille tehdään luihin perustuva riggaus, koska se on yhteensopiva Unityn kanssa. En ollut itse PVO-projektissa vastuussa hahmojen riggaamisesta ja animoinnista, mutta käyn silti läpi tavan jolla se olisi mahdollista tehdä. Riggaustapa on myös

todettu toimivaksi Unity-pelinmoottorin kanssa, mutta ei ole välttämättä mobiilialustalle optimaalisin.



Kuvio 36. Örkkihahmon luuranko Blender 3D:ssä.

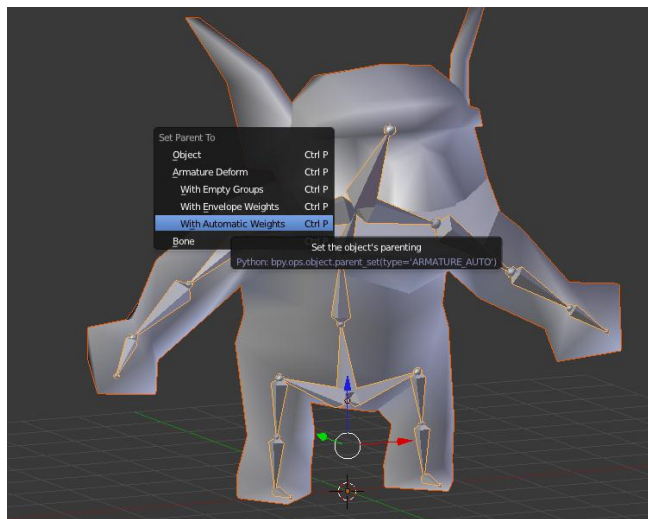
3D-hahmojen animointia ja liikuttelua varten täytyy toteuttaa riggaus, eli toteutetaan mekanismi niiden liikuttamiseksi. Hahmojen sisään tehdään luut (Kuvio 36.), jotka liikuttavat niille määrättyjä osia. Luut saadaan toimimaan esimerkiksi käyttämällä weight paint toimintoa, jossa luun hallitsema kohta maalataan väriasteikolla sinisestä punaiseen. Sininen väri edustaa arvoa 0, eli niitä kohtia joita luu ei pysty liikuttamaan. Punainen väri vastaa arvoa 1, jolloin luun vaikutus vertekseihin on vahvin.



Kuvio 37. Weight paint Blenderissä. Päässä oleva luu ei vaikuta sinisinä näkyviin kohtiin.

Kun hahmo on valmis rigattavaksi, sille tehdään ensin yksittäinen luu, josta lähdetään rakentamaan koko luurankoa. Mitä enemmän hahmossa on luita, sitä enemmän myös niveliä ja liikkuvuutta. Mobiilipeleissä hahmoissa on niin vähän polygoneja, että myöskään luita ei tarvita paljon. Luiden suuri määrä tekee myös pelin toimivuudesta raskaampaa. Hahmon selkärangassa voi olla esimerkiksi kolme luuta. Luiden välinen nivel täytyy sijoittaa aina verteksisilmukan kohdalle. Blenderissä uusia luita on helppo luoda lisää extrude-toiminnolla, joka toimii samalla tavalla kuin mallintaessa. Selkärangasta saadaan päähän ja olkapäihin menevät luut valitsemalla selkärangan kärkiosa ja käyttämällä extrudea. Olkapäiden luut jatkuvat käsivarsiksi ja lantion luut jaloiksi.

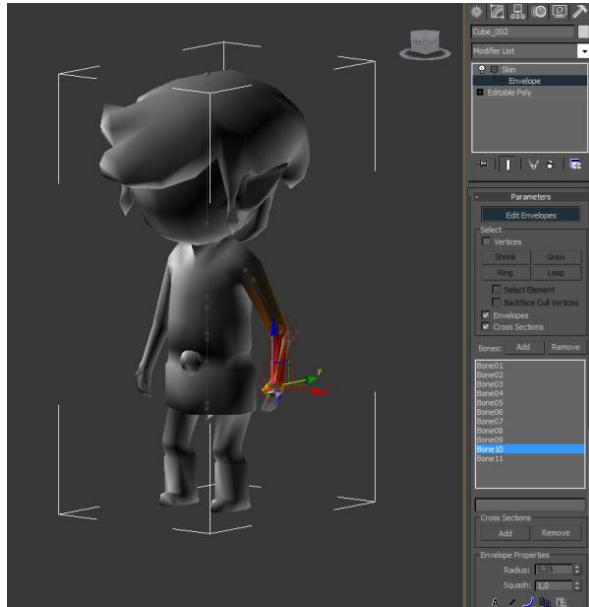
Kun hahmolla on luuranko, täytyy se vielä saada toimimaan yhdessä hahmomallin kanssa. Blenderissä yksinkertainen tapa on object-modessa valita ensin hahmomalli sitten luut ja painamalla ctrl+p, jolloin luista tehdään mallin parent-objekti. Kun parent-valikosta vielä valitaan kohta ” with automatic weights” (Kuvio 38.), saadaan luut vaikuttamaan ympärillä olevaan meshiin automaattisesti ja niillä voidaan liikuttaa mallia pose modessa. Automaattinen luiden vaikutus toimii yksinkertaisissa malleissa yleensä hyvin, mutta halutessaan voi parannella luiden toimivuutta weight paint modessa. On myös tärkeää nimetä luut uudelleen vastaamaan niiden hallinnassa olevia kohtia.



Kuvio 38. Luiden yhdistäminen malliin Blenderillä on helppoa.

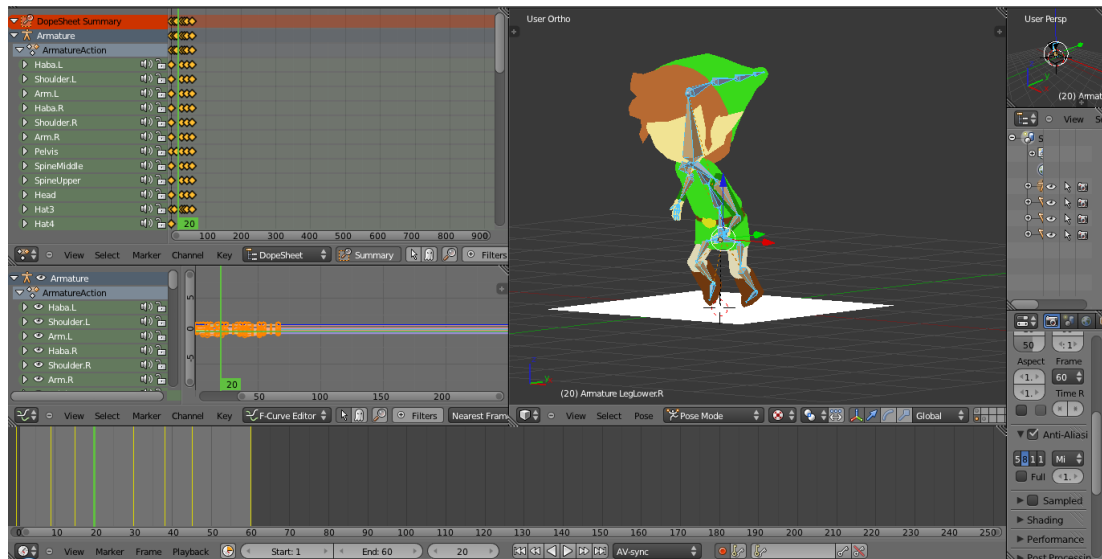
3DSMaxissa luut tehdään lähes samalla periaatteella. Käsien ja jalkojen luut on helppo kopioida valitsemalla kaikki luut ja valitsemalla tools-valikosta mirror. Työkalu tekee käsien

luista peilikuvan toiselle puolelle. Kun uusia luita lisätään, ohjelma luo automaattisesti uuden luun, joka on yhdistyneenä edelliseen, joten extrude-työkalua ei tarvita. Kun tehdään hahmolle käsien ja jalkojen luut, täytyy ne yhdistää erikseen selkärankaan käyttämällä select and link-toimintoa. Ohjelmassa on myös mahdollisuus yhdistää luurankoon dummy-objekti, jolla voi helposti liikuttaa koko hahmoa.



Kuvio 39. Luun vaikutusalueen muokkaaminen 3D Studio Maxissa.

Kun luuranko on valmis, se saadaan toimimaan käyttämällä Skin-modifieria, joka löytyy modifier-valikosta. Skin modifierin valikosta valitaan Add, jolloin avautuu lista olemassa olevista luista. Listasta valitaan kaikki hahmon luut ja valitaan select, jolloin luut toimivat, mutta vaativat vielä säätämistä. Skin-modifierin alavalikosta löytyy envelope-valikko, josta luiden vaikutusalueita voi säätää (Kuvio 39.). Luiden vaikutusalue näkyy hahmomallissa punaisella värillä ja sen kokoa voi muuttaa luiden ympärillä olevista kehikoista.



Kuvio 40. Blenderin animaatiotyökalut. Alla näkyvissä aikajana ja avainkehukset.

3DSMaxissa on lisäksi myös oma sisäinen riggaus-menetelmä nimeltään biped, jolla yhdistetään valmis ihmisen luuranko 3D-malliin. Biped-luurankoa on mahdollista muokata monin tavoin sopimaan 3D-hahmon ruumiin muotoon. Luurangolla on myös valmis toiminto, jolla se matkii tapoja miten ihmisruumis liikkuu, käyttäen käänteistä kinematiikkaa. Esimerkiksi, jos luurangon kämmenosaa vetää, kyynärvarsi ja olkavarsi liikkuvat sen mukana luonnollisesti, aivan kuin vetäessä oikean ihmisen kättä.

## 9.5 Teksturointi

Teksturointi tehdään kuvankäsittelyohjelmilla. Tekstuurit voidaan piirtää kokonaan käsin tai käyttää valokuvia, joita käsitellään. Tätä vaihetta auttamaan, tehdään 3d-mallinnusohjelmassa kuva UV-kartasta, jota voi käyttää pohjana teksturoinnissa.

Koska Pets vs Orcsin tyyli ei ole realistista, kaikki tekstuurit piirrettiin perinteisesti käsin ja käyttäen Photoshopin suodattimia tehosteena. Teksturoinnissa kannattaa tehdä jokainen vaihe eri kerrokselle, jolloin niiden muokkaaminen on kätevempää. Kun UV-kartta on avattu kuvankäsittelyohjelmalla, tehdään uusi kerros ja aletaan suunnittelemaan pohjaväriä hahmolle. Pohjaväri peittää koko tekstuurin ja on sen yleisin väri. Tässä tapauksessa käytämme örkille sopivaa vihreää väriä ja sitä voi myös tarvittaessa muuttaa.

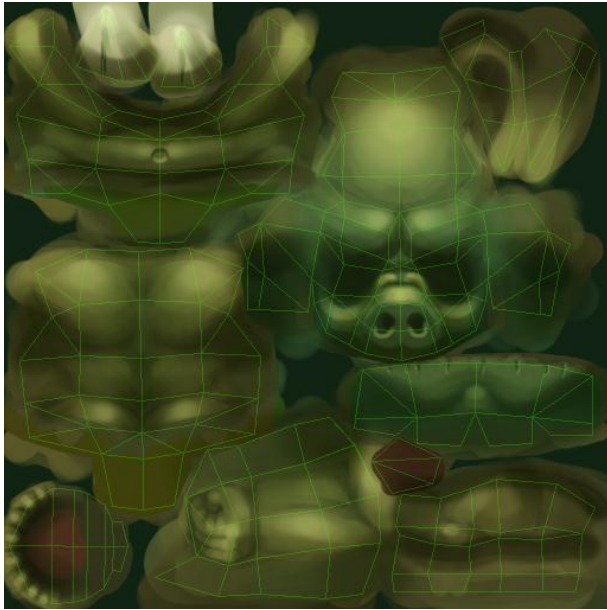


Sekä Gimpissä että Photoshopissa teksturoinnin peruseriaate on sama. Molemmissa ohjelmissa voi tehdä tasoja ja muokata tason tyyliä sopivaksi esimerkiksi varjojen piirtämiseen. Suurin rajoite Gimpillä teksturoimisessa on sivellinasetusten vähyys verrattuna Photoshopiin. Molemmissa ohjelmissa on kuitenkin mahdollisuus tehdä omia sivellintyyliä esimerkiksi kuvista.



Kuvio 41. Teksturoinnin pohja ja UV-kartta apuna pohjalla.

Koska Pets vs Orcsin hahmoissa ei käytetty shadereita eli pelimoottorissa ei laskettu valaistuksia, jotka mahdollistavat reaaliaikaisen varjostuksen, varjot piirrettiin suoraan tekstuuriin. Varjot piirretään omalle tasolle ja sen tyyliksi valitaan multiply, joka tekee varjon väristä oikean sävyistä. Koska peli on kuvattu ylhäältä päin, tekstuurien valaistus suunnattiin myös tulemaan yläsuunnasta.



Kuvio 42. Tekstuureihin lisätty varjostusta ja valaistusta.

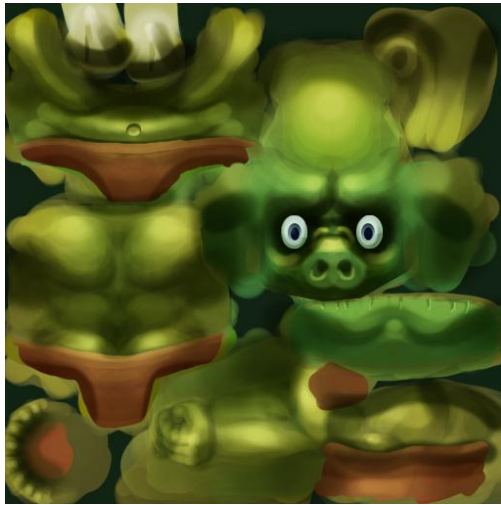
Valaistuksen jälkeen tekstuuriin on lisätty yksityiskohtia, kuten silmät ja vaatekappaleita. Hahmoilla ei ole erillistä kasvoanimaatiota, joten silmät voidaan piirtää suoraan tekstuuriin.



Kuvio 43. Tekstuuriin lisätty yksityiskohtaisuutta.

Tekstuuriin piirtämisen aikana, niiden ulkonäköä voi tarkkailla testaamalla niitä 3D-mallin kanssa 3D-mallinnusohjelmassa tai pelimoottorissa. Näin selvitetään myös miten hyvin hahmo erottuu muusta pelin grafiikasta ja että sen tyyli on konsistentti. Kun kaikki tarvittava on piirretty tekstuuriin, siihen voidaan tehdä vielä viimeistelyjä, kuten kontrastin ja

värikylläisyyden muuttamista. Lopullinen örkin tekstuuri on saanut kirkkaammat värit, joka edesauttaa sen erottamista pelikentältä.



Kuvio 44. Tekstuurin kontrastia ja värikylläisyyttä muokattu.



Kuvio 45. Lopullinen hahmo ja teksturointi.

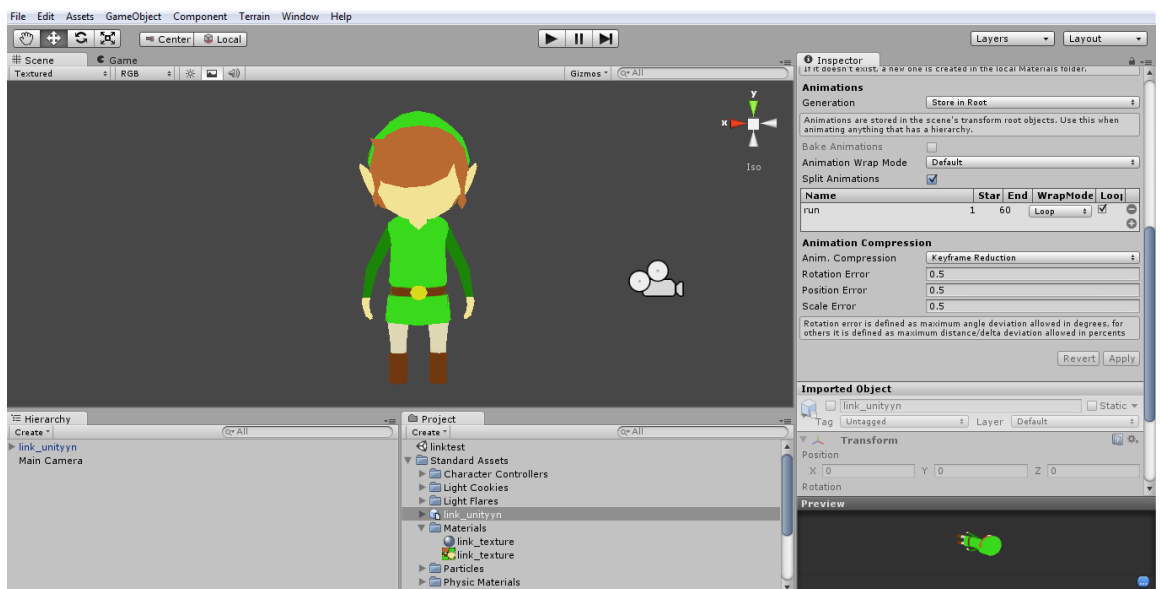
## 9.6 Toimivuus Unityn kanssa

Kun kaikki hahmon luonnin vaiheet on toteutettu, täytyy se saada vielä toimimaan pelimoottorissa. Unity 3D on suoraan yhteensopiva useimpien 3D-mallinnusohjelmien kanssa, joten mallien siirtäminen on todella vaivatonta. 3DSMaxin ja Blenderin tiedostot voi siirtää suoraan Unityn projektikansioon tai halutessaan tiedostot voi muuttaa ensin Fbx-

muotoon. Unityyn tuodaan tiedostoista mallit ja niiden verteksit, polygonit, kolmiot, UV-kartat sekä rigatut hahmot ja niiden luut sekä animaatiot toimivat myös. Lisäksi tuodaan tietoja mallien koosta, rotaatiosta ja skaalasta sekä objektien nimistä.

Kun hahmo on valmis sen voi tallentaa joko Blender- tai 3DSMax-tiedostomuotoon ja siirtää Unity-projektin kansioon. Unityssä 3d-objektille tehdään materiaali ja valitaan sille oikea tekstuuri. Hahmon animaatiot voidaan jakaa osiin, esimerkiksi juoksuun, kävelyyn, hyppyyn määrittämällä kehysten oikea alku- ja loppukohta.

Jotta molempien ohjelmien tiedostomuodot toimisivat Unityssä, täytyy ohjelmat olla myös asennettuna koneelle. Jos ohjelmia ei ole saatavilla voi käyttää vaihtoehtoisesti fbx-tiedostoja, joiden tekeminen onnistuu sekä Blenderissä että 3DSMaxissa.



Kuvio 46. Blenderissä tehty 3D-malli sirrettyinä Unityyn. Animaatioksi on määritetty ”run”, joka alkaa kuvasta 1 ja loppuu kuvaan 60.

## 9.7 Käyttöliittymät

Ohjelmistojen käyttöliittymien toimivuus on myös tärkeässä osassa, kun vertaillaan niiden käyttöä ja sopivuutta grafiikan tuotantoon.

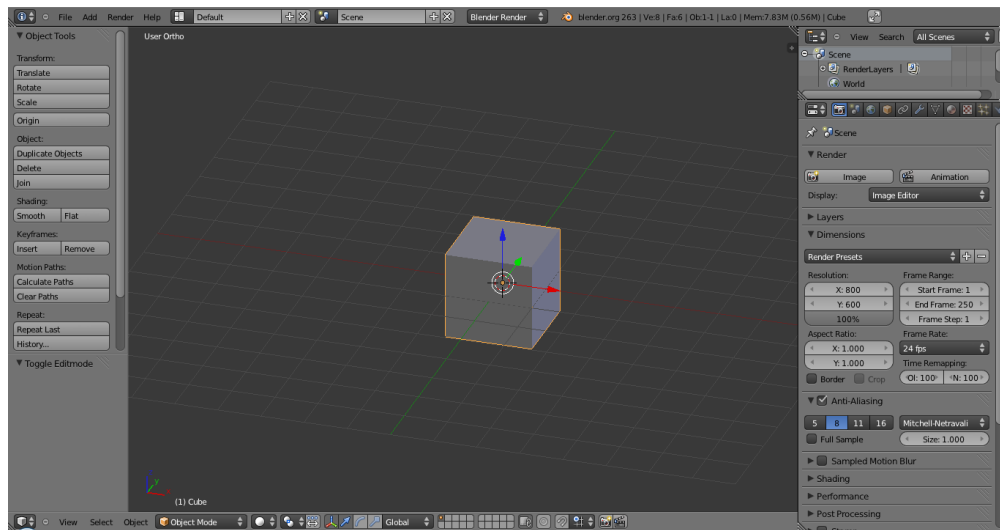
3D Studio Maxin käyttöliittymä on pysynyt suunnilleen samanlaisena versiosta toiseen. Lisätoimintojen myötä käyttöliittymä on vain saanut lisää uusia painikkeita ja yleensä moni toiminto on piilossa välilehden tai vierityspalkin takana. Käyttöliittymän tyyli on tarvittavan selkeää ja eri painikkeissa on laajat ohjeet kun kursori viedään sen päälle. Käyttöliittymän pikanäppäinten kustomointimahdollisuudet ovat myös hyvät. Ohjelma on saanut kritiikkiä käyttöliittymän sekavuudesta ja joidenkin toimintojen turhasta monimutkaisuudesta. 3D Studio Max 2013 on saanut hieman parannuksia käyttöliittymään, tarjoamalla esimerkiksi erilaisia näyttöasetelmia, joilla voi kätevästi muokata mm. 3D-näkymien asettelua. Tämä ominaisuus on tosin ollut Blenderissäkin jo usean version ajan.



Kuvio 47. 3DSMax 2012.

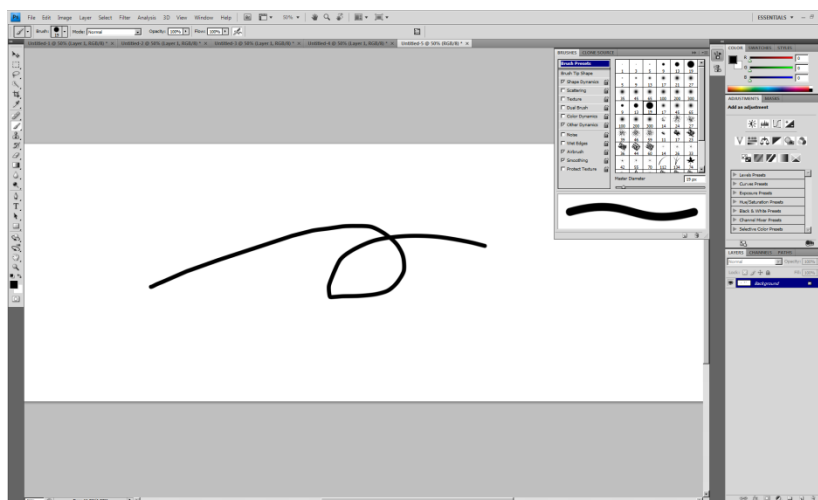
Blender 3D:n käyttöliittymä uudistettiin kokonaan versiossa 2.5, joka on parantanut käyttökokemusta huomattavasti. Perustyökalut siirrettiin alapalkista sivulle, joka antaa enemmän tilaa 3D-näkymälle. Tärkeimmät painikkeet ovat löydettävissä helposti ja ohjelman eri toimintoja voi myös etsiä, painamalla välilyöntiä ja kirjoittamalla hakusana työkalulle.

Oletusasetuksessa näkymän oikealla puolella sijaitsee asetukset esimerkiksi materiaaleille, renderöintiin ja modifier-työkaluille. Vasemmalla puolella on object-tools, eli objektien hallintaan käytettävät perustyökalut kuten kääntely, skaalaus ja kopiointi. Blenderin käyttö on pikanäppäinpainotteista, joten objektienhallintatyökaluja ei usein tarvitse käyttää valikon kautta.



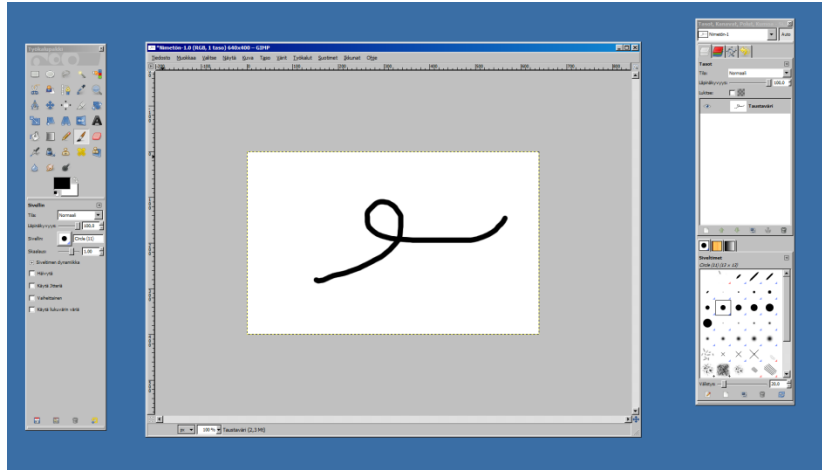
Kuvio 48. Blender 3D 2.63.

Photoshopin käyttöliittymä on myös pysynyt samantyyllisenä usean vuoden ajan. Perustyökalut ovat vakiona ruudun vasemmassa laidassa ja tasojen hallinta ruudun oikeassa laidassa. Työkalujen sijaintia voi halutessaan muuttaa helposti. Käyttöliittymän väritys on muuttunut versiossa CS6 tummaksi, kuten 3DSMaxissa.



Kuvio 49. Photoshop CS4.

GIMP:n käyttöliittymä on hyvin samanlainen kuin Photoshopissa, mutta kaikki ikkunat ovat vakiona irrallaan toisistaan. Kuten Photoshopissa, perustyökalut löytyvät vasemmasta laidasta ja tasot oikeasta.



Kuvio 50. GIMP 2.

## 10 LOPPUTULOKSET JA ARVIOINTI

Lopputuloksena saatiin pelissä käytettäviä 3D-hahmoja, sekä todettiin myös ilmaisten ohjelmien pääsevän samaan lopputulokseen ja sopivan hyvin pelinkehitykseen. Alkuperäinen Pets vs Orcs kehitettiin käyttäen maksullisia grafiikkaohjelmistoja, mutta myöhemmin samaa prosessia testattiin myös ilmaisilla ohjelmilla ja hahmot saatiin toimimaan Unity 3D-pelimoottorissa. Aloitteleva mobiilipeliyritys voi siis harkita myös ilmaisohjelmistoa kalliiden lisenssien sijaan. Unity 3D:n kanssa työskentely mobiililaitteille vaati tosin maksullisen version, mutta kustannus on pienempi verrattuna esimerkiksi 3D Studio Maxin lisenssiin. Ilmaiset ja pätevät kehitystyökalut houkuttelevat myös kokeilemaan taitojaan ja saamaan uusia kehittäjiä pelimarkkinoille.



Kuvio 51. Pets Vs Orcs-pelin lopullisia pelihahmoja.

Toteutetut hahmot olivat selkeitä, toisistaan erottuvia ja sopivat pelin kuvakulmiin. Vaikka hahmot ovat pelin näkymässä todella pieniä, hahmoluokat olivat tunnistettavia. Tarvittaessa kuvakulmaa voi siirtää lähemmäksi hahmoja, jonka toteutus on kätevää 3D-grafiikkaa



käyttäessä. Kentän tausta ei myöskään estä pelaajaa näkemästä hahmoja. Hahmojen polygonimäärä oli sopiva mobiililaitteisiin ja peli saatiin pyörimään tarpeeksi hyvin. Pelin grafiikka vastasi suunnitelmia, vaikka konseptitaiteen tasolle on vaikea päästä 3D-grafiikalla. Pelin värikylläisyys ja tyyli oli myös sopivaa casual-pelille.

Hahmoista löytyy myös parannettavaa. Joitakin muotoja olisi voinut korostaa enemmän, jotta hahmot olisivat entistäkin selkeämpiä ja erottuisivat hyvin pienemmältäkin älypuhelimien näytöltä. Esimerkiksi pään ja raajojen kokoa voisi suurentaa vielä reilusti. Tiukan aikataulun puitteissa ei ollut aikaa kokeilla erilaisia graafisia tyylejä tai useampia eri hahmomalleja, vaan hahmojen ulkonäkö täytyi hyväksyä konseptipiirrosten pohjalta. Tärkeää oli myös hahmojen toimivuus Unity 3D:n ja mobiililaitteiden kanssa ja siihen tavoitteeseen päästiin.



Kuvio 52. PVO:n pelinäkö.

Ohjelmien vertailussa löytyi pieniä eroja, mutta mobiilipelihahmojen kehityksessä erot eivät ole merkittäviä. Monimutkaisemmassa tuotantoprosessissa ohjelmistojen erot tulevat paremmin näkyviin. Ohjelmilla työskentely ei ole kovin erilaista ja tuotantoprosessi pysyy aina samana, riippumatta mitä ohjelmaa käytetään. Haasteellisinta oli selvittää miten tietyt asiat toteutetaan eri ohjelmissa. Aiempi Blenderin ja Photoshopin käyttökokemus vaikutti myös lopulliseen tulokseen. Esimerkiksi ohjelman käyttöliittymien hallitseminen ja

työskentelynopeus riippuu täysin käyttäjästä. Ilmaisia ohjelmia kannattaa ehdottomasti kokeilla ja opetella ennen lopullisen ostopäätöksen tekemistä.

Grafiikkaohjelmia on hyvin vaikea vertailla keskenään ja tässä tapauksessa kaikista merkittävin ero ohjelmien välillä on hinta. Jos käyttäjällä on tarpeeksi taitoa, kaikkiin toteutusvaiheisiin löytyy vastaavat työkalut kaikista ohjelmista. Kannattaa siis ottaa selvillä mitä hyötyä maksullisten ohjelmien hankinnassa oikeasti on.

## 10.1 Ohjelmistojen muut erot

Vertailussa olevilla ohjelmilla on muitakin käytännön eroja kuin sisäiset työkalut. Ohjelmien laitevaatimukset ja koko vaihtelee myös todella paljon. Käyttöliittymän toimivuus ja ohjelmien sulava toiminta voi edistää työskentelyä huomattavasti.

Blender, 3DSMax, Photoshop, GIMP ovat yhteensopivia Windowsin ja Maccien kanssa, mutta Blender ja GIMP tukevat myös Linuxia.

Ohjelmien laitevaatimukset ovat sopivia nykypöydille, mutta hieman vanhemmilla komponenteilla sekä kannettavilla tietokoneilla voi tulla ongelmia erityisesti 3DSMaxin kanssa. Kuvankäsittelyohjelmassa laiteistovaatimukset riippuvat siitä, kuinka useata kuvaa käsitellään samanaikaisesti ja kuinka suuri on kuvien resoluutio.

## 10.2 Omat kommentit

### Blender 3D

Blender on laitevaatimuksiltaan kevyt mallinnusohjelma ja sen hankkiminen on todella vaivatonta ja nopeaa. Käyttöliittymän uudistuksen myötä ohjelma on noussut uudelle tasolle ja on varteenotettava työkalu peliprojekteissa ja myös animaatioelokuvissa. Blender toimii hyvin myös hieman vanhemmilla laitteilla. Avoin lähdekoodi mahdollistaa ohjelman vapaan muokkaamisen, uusien ohjelmistolaajennusten runsaan kehityksen ja sopii erityisesti 3D-mallinnusta aloitteleville henkilöille.

Blenderin ja 3D Studio Maxin yhteensopivuudessa on pieniä ongelmia, jotka voivat osoittautua ongelmallisiksi jos projektissa käytetään molempia ohjelmia. Blender tukee FBX-tiedostojen vientiä, mutta ei tuontia. FBX-tiedostoja joudutaan myös mahdollisesti korjaamaan vielä 3DSMaxissa. Tämä ongelma on mahdollista ratkaista ohjelmistoliitännäisillä.

Erikoisuuksina Blenderissä on myös oma sisäinen pelimoottori, valmius digitaaliseen veistämiseen sekä videoeditointityökalut. Ohjelma tukee myös Python Script –ohjelmointikieltä, joka ei ole rajoittunut vain Blenderin sisäiseksi kieleksi vaan on yleisesti käytetty.

Blender 3D on vähemmän käytetty ohjelma kuin 3DSMax, joten ei sovi välttämättä isojen studioiden tuotantotapoihin, mutta sopii hyvin mobiilipeliprojekteihin. Ilmaisenä ohjelmana käyttäjätuki ja takuu ohjelman toimivuudesta ei ole yhtä varma kuin 3DSMaxissa, jota voidaan pitää riskinä suurissa projekteissa.

### 3D Studio Max

3D Studio Max on suosittu mallinnusohjelma peliteollisuudessa, jonka perusteet kannattaa opetella välttääkseen mahdollisia eri ohjelmien välisiä yhteensopivuusongelmia suurissa peliprojekteissa. Se sisältää myös enemmän työkaluja kuin Blender, mutta kaikki eivät ole tarpeellisia mobiilipeliteollisuudessa. Mallinnukseen käytettävät työkalut ovat monipuolisia ja myös UV-mappausominaisuudet ovat laajat. Käyttöliittymä on hieman sekava ja kaipaisi mielestäni uudistusta.

3DSMaxin laitevaatimukset ovat suuremmat kuin Blenderissä. Olen vertaillut toimivuutta omalla tietokoneellani ja huomannut selvän eron toiminnassa. Ohjelma käyttää omaa MaxScript –ohjelmointikieltä, joka on rajoittunut vain 3DSMaxin käyttöön.

Ohjelman lisenssi on kallis, mutta 30-päivän kokeiluversion lataaminen on mahdollista. Maksullinen lisenssi tuo myös etuja verrattuna ilmaiseen, kuten paremman tuotetuen mahdollisia ongelmia varten.

## Photoshop

Photoshopissa on monipuoliset piirto-ominaisuudet ja siveltimen muokkausmahdollisuudet. Kokeilemistani kuvankäsittelyohjelmista paras tekstuuriin piirtämiseen. Myös Photoshop kuuluu monien yritysten standardiohjelmiin ja kannattaa siis opetella. Ohjelman kallis lisenssimaksu saattaa pelottaa aloittelevia kuvankäsittelijöitä. Teksturoinnin lisäksi, ohjelmalla onnistuu myös korkealaatuisten konseptipiirrosten ja myynninedistämismateriaalin teko. Ohjelman heikkoutena GIMPiin verrattuna voidaan pitää yhteensopimattomuutta Linux-käyttöjärjestelmien kanssa. Photoshop sisältää enemmän eri ominaisuuksia kuin GIMP, mutta läheskään kaikkia ei tarvita esimerkiksi teksturoinnissa.

## GIMP

Avoin lähdekoodi ja soveltuu myös hyvin teksturointiin. Työkalut eivät ole yhtä laajoja kuin Photoshopissa, mutta monipuoliset ilmaiseksi ohjelmaksi ja tarpeeksi kattavat mobiilipeliinprojektiin. Sopii hyvin harjoitustyökaluksi ja myös oikeisiin projekteihin. Ohjelman käytön peruseriaate on hyvin samankaltainen, kuin Photoshopissa. GIMP sisältää monia samoja työkaluja kuin Photoshop, mutta käyttöliittymä on vain hieman erilainen. GIMP ei vaadi niin suuria laitevaatimuksia kuin Photoshop ja sen koko on myös huomattavasti pienempi, joten sen voi tallentaa vaikka muistitikulle. Vähemmän kehittyneitä ominaisuuksia kuin Photoshopissa, mutta peruskäyttäjä selviää hyvin GIMPin ominaisuuksillakin. Helposti laajennettavissa ohjelmistolaajennuksilla, avoimen lähdekoodin ansiosta.

## 11 LÄHTEET

A Digital Dreamer. <http://www.adigitaldreamer.com/articles/video-game-texture-mapping.htm> (Luettu 14.10.2012).

About.com. <http://3d.about.com/od/3d-101-The-Basics/a/Introduction-To-3d-Modeling-Techniques.htm> (Luettu 1.5.2012).

About.com b. <http://3d.about.com/od/3d-101-The-Basics/a/Anatomy-Of-A-3d-Model.htm> (Luettu 14.10.2012).

About.com c. <http://3d.about.com/od/Creating-3D-The-CG-Pipeline/a/What-Is-Rigging.htm> (Luettu 10.11.2012).

About.com d. <http://3d.about.com/od/3d-101-The-Basics/a/Introduction-To-3d-Modeling-Techniques.htm> (Luettu 14.10.2012).

Addictive Tips. <http://www.addictivetips.com/mobile/an-introduction-to-modern-mobile-operating-systems/> (Luettu 1.5.2012).

Adobe. <http://www.adobe.com/products/photoshopextended/faq.html> (Luettu 1.5.2012).

Allan Brito. <http://www.allanbrito.com/wp-content/uploads/2008/07/box-modeling.jpg> (Luettu 14.10.2012).

Android Developers. <http://developer.android.com/guide/publishing/publishing.html> (Luettu 1.5.2012).

Animation Arena. <http://www.animationarena.com/introduction-to-3d-animation.html> (Luettu 29.11.2012).

Apple a. [http://support.apple.com/kb/HT2001?viewlocale=fi\\_FI](http://support.apple.com/kb/HT2001?viewlocale=fi_FI) (Luettu 1.5.2012).

Apple b. <http://www.apple.com/iphone/built-in-apps/app-store.html> (Luettu 1.5.2012).

Apple c. <https://developer.apple.com/programs/ios/> (Luettu 1.5.2012).

Area. <http://area.autodesk.com/maxturns20/history> (Luettu 1.5.2012).

Autodesk.com. [http://images.autodesk.com/adsk/files/3ds\\_max\\_2011\\_help.pdf](http://images.autodesk.com/adsk/files/3ds_max_2011_help.pdf) (Luettu 1.5.2012).

Blender Wiki. <http://wiki.blender.org/index.php/Doc:2.6/Manual> (Luettu 1.5.2012).

Blender.org. <http://www.blender.org/education-help/> (Luettu 1.5.2012).

BlenderWiki b.

<http://wiki.blender.org/index.php/Doc:2.6/Manual/Textures/Mapping/UV> (Luettu 10.11.2012).

blog.tartiflop <http://blog.tartiflop.com/wp-content/uploads/2008/11/texturemapping.png> (Luettu 19.11.2012).

Canalys. <http://www.canalys.com/newsroom/google%E2%80%99s-android-becomes-world%E2%80%99s-leading-smart-phone-platform> (Luettu 1.5.2012).

Curious Blog. [http://curiousblog.files.wordpress.com/2010/08/idevice\\_gestures.jpg](http://curiousblog.files.wordpress.com/2010/08/idevice_gestures.jpg) (Luettu 1.5.2012).

Dummies. <http://www.dummies.com/how-to/content/how-to-use-the-ipads-touchscreen-interface.html> (Luettu 12.10.2012).

Euclideanspace. <http://www.euclideanspace.com/threed/animation/keyframing/index.htm> (Luettu 29.11.2012).

Ezine Articles. <http://ezinearticles.com/?2D-and-3D-Computer-Graphics&id=1462411> (Luettu 1.5.2012).

Feedagg. <http://www.feedagg.com/feed/14909199/Graphic-Designing> (Luettu 14.10.2012).

Franson D. & Thomas E. 2007. Game Character Design Complete. Boston: Thomson Course Technology PTR.

Gameogre. <http://www.gameogre.com/starcraft2c.jpg> (Luettu 14.10.2012).

Gathering.

<http://www.gathering.org/tg12/files/content/images/creativia/seminars/3Dillustrasjon.jpg> (Luettu 14.10.2012).

- Giant Bomb. <http://www.giantbomb.com/2d/92-1427/> (Luettu 15.6.2012).
- Gimp a. <http://www.gimp.org/about/introduction.html> (Luettu 1.5.2012).
- Gimp b. [http://www.gimp.org/about/ancient\\_history.html](http://www.gimp.org/about/ancient_history.html) (Luettu 1.5.2012).
- GNU. <http://www.gnu.org/> (Luettu 1.5.2012).
- Holmes3D. <http://www.holmes3d.net/graphics/subdivision/> (Luettu 14.10.2010).
- Idevgames. <http://www.idevgames.com/articles/ios-overview> (Luettu 1.5.2012).
- ImagineFX. <http://www.imaginefx.com/02287754330842797110/tutorial.pdf> (Luettu 1.5.2012).
- Kioskea. <http://en.kioskea.net/faq/245-3d-graphics-software> (Luettu 29.11.2012).
- Low End Mac. <http://lowendmac.com/software/p/photoshop.html> (Luettu 1.5.2012).
- Ludocraft. [http://ludocraft.oulu.fi/elias/dokumentit/game\\_production\\_process.pdf](http://ludocraft.oulu.fi/elias/dokumentit/game_production_process.pdf) (Luettu 1.5.2012).
- New World Encyclopedia.  
[http://www.newworldencyclopedia.org/entry/Computer\\_graphics](http://www.newworldencyclopedia.org/entry/Computer_graphics) (Luettu 1.5.2012).
- Pardew L. 2005. Beginning Illustration and Storyboarding for Games. Boston: Thomson Course Technology PTR.
- Patriot Truck Leasing. [http://patriottruckleasing.com/body-images/Resolution\\_illustration.png](http://patriottruckleasing.com/body-images/Resolution_illustration.png) (Luettu 1.5.2012).
- Pedronunezart.  
[http://3.bp.blogspot.com/\\_WMKJHmpmUcM/TNyZJ1ed59I/AAAAAAAAAQY/z2k45QIDw4s/s1600/CASTLEVANIA\\_BUCHER.jpg](http://3.bp.blogspot.com/_WMKJHmpmUcM/TNyZJ1ed59I/AAAAAAAAAQY/z2k45QIDw4s/s1600/CASTLEVANIA_BUCHER.jpg) (Luettu 1.5.2012).
- RealWorld Graphics. <http://www.rw-designer.com/NURBS> (Luettu 12.10.2012).
- Skill-Guru <http://www.skill-guru.com/blog/2011/01/10/getting-started-with-android-development/> (Luettu 10.11.2012).

SlashGear. [http://cdn.slashgear.com/wp-content/uploads/2009/02/iphone\\_gaming.jpg](http://cdn.slashgear.com/wp-content/uploads/2009/02/iphone_gaming.jpg)  
(Luettu 1.5.2012).

Techcrunch. <http://techcrunch.com/2012/09/12/ios-app-store-boasts-700k-apps-90-downloaded-every-month/> (Luettu 29.11.2012).

Webobedia. <http://www.webopedia.com/TERM/G/graphics.html> (Luettu 29.11.2012).