

Martin Kontinen

PELIKEHITYS GRAFIIKKATUOTAN- NON NÄKÖKULMASTA

Opinnäytetyö
Tieto- ja Viestintäteknikka / Peliohjelmointi

2020



**Kaakkois-Suomen
ammattikorkeakoulu**

Tekijä/Tekijät	Tutkinto	Aika
Martin Kontinen	Insinööri (AMK)	Kesäkuu 2020
Opinnäytetyön nimi		55 sivua
Pelikehitys grafiikkatuotannon näkökulmasta		
Toimeksiantaja		
GameLab, XAMK		
Ohjaaja		
Pekka Vilpponen		
Tiivistelmä		
<p>Tämä opinnäytetyö dokumentoi kolmiulotteisen pelihahmon suunnittelun, toteuttamisen ja tuomisen pelimoottoriin. Työn tavoitteena oli tutkia kolmiulotteisen pelihahmon suunnitteluun ja toteuttamiseen tarvittavaa teoriaosaamista ja selventää käytännön toteutukseen vaadittavia työvaiheita.</p> <p>Kyseessä on Xamkin Gamelabin tilaama kehittämistyö, jonka tarkoituksena oli kartoittaa animoidun ja teksturoidun pelihahmon suunnitteluun ja toteutukseen tarvittavia vaiheita, työvälineitä ja prosessin kulkua.</p> <p>Hahmon toteuttamisessa käytettiin Autodeskin 3ds Max -mallinnusohjelmaa. Pelihahmon tekstuurikartan muokkaukseen käytettiin Paint.NET-kuvankäsittelyohjelmaa. Lopuksi valmis pelihahmo tuotiin Unreal Engine 4 -pelimoottoriin.</p> <p>Opinnäytetyön tuloksena syntyi animoitu ja teksturoitu kolmiulotteinen pelihahmo, joka on valmis käytettäväksi useimmissa pelimoottoreissa.</p>		
Asiasanat		
pelikehitys, grafiikka, 3D-mallinnus		

Author (authors)	Degree	Time
Martin Kontinen	Bachelor of Engineering	June 2020
Thesis title		55 pages
Graphic production in game development		
Commissioned by		
GameLab, XAMK		
Supervisor		
Pekka Vilpponen		
Abstract		
<p>This thesis documents the process of designing, creating and importing a three-dimensional game character into a game engine. The aim of this thesis was to study the theoretical knowledge needed in the designing and creating of a three-dimensional game character and to examine the work steps required for practical implementation.</p> <p>The thesis is a development study commissioned by GameLab, with a purpose of examining the needed steps, tools and process flow in the creation of a fully textured and animated game character.</p> <p>The character was created using Autodesk's 3ds Max modeling software. Paint.NET image editing software was used to edit the game character's texture map. Lastly, the finished game character was imported into Unreal Engine 4 game engine.</p> <p>The result of the thesis was an animated and textured three-dimensional game character that is ready for use in most game engines.</p>		
Keywords		
game development, graphics, 3D-modeling		

SISÄLLYS

1	JOHDANTO	6
2	PELIKEHITYS	7
3	GRAFIKKATUOTANTO	8
3.1	Käyttötarkoitukset.....	8
3.2	Suunnittelu	9
4	2D-GRAFIikka	11
4.1	Vektorigrafiikka.....	12
4.2	Rasterigrafiikka.....	12
5	3D-GRAFIikka	13
5.1	Monikulmioverkko.....	14
5.2	Mallinnusohjelmat.....	15
5.3	Apuohjelmat	16
6	3D-MALLINTAMINEN	17
6.1	Tyyli ja tekniikat.....	18
6.1.1	Editable Poly	18
6.1.2	Editable Mesh.....	18
6.1.3	Editable Patch	19
6.1.4	NURBS.....	19
6.2	Modifier-työkalut.....	19
6.3	Topologia	20
6.4	Animoimattomat mallit.....	21
7	PELIHAHMON TOTEUTTAMINEN	22
7.1	Suunnittelu	22
7.2	Referenssikuva.....	22
7.3	Esivalmistelut mallinnusohjelmassa	25
7.4	Mallintamisen aloittaminen	29
7.5	Teksturointi.....	33

7.6	Animoidut mallit.....	39
7.7	Luuobjektit.....	39
7.8	Skinning-prosessi.....	42
7.9	Animointi	44
7.10	Pelimoottoriin vieminen	47
7.11	Mallin tuominen pelimoottoriin.....	48
8	POHDINTA.....	50
	LÄHTEET	52
	KUVALUETTELO	

1 JOHDANTO

Videopeliteollisuus on tietokoneiden suorituskyvyn ja saatavuuden myötä kasvanut teollisuuden toimiala, joka on työllistänyt Suomessa vuonna 2019 yli 3 000 pelikehittäjää (Yrittäjät 2019). Noin kolmasosa maailman väestöstä käyttää videopelejä viihdekäyttöön. Videopeliteollisuutta hyödynnetään viihdekäytön ulkopuolella myös muun muassa opetuskäytössä sekä terveys- ja hyvinvointipalveluiden pelillistämiseen (Puhakka 2008, 24).

Tässä Kaakkois-Suomen ammattikorkeakoulun Kotkan kampuksen GameLabin toimeksiantamassa opinnäytetyössä syvennyttään pelikehityksen graafiseen puoleen. Peligrafiikan ollessa olennainen osa nykyaikaista pelinkehitystä, se on myös olennainen osa Gamelabissa tehtävässä pelikehityksessä. Kotkan kampuksen peliohjelmoinnin koulutuksen opiskelijoiden opetussuunnitelmaan kuuluu ohjelmointipainotteisten opintojen lisäksi vain yksi kurssi, jossa keskitytään peligrafiikkaan ja mallintamiseen. Koulutuksen ohjelmistokeskeisyyden vuoksi saattaa graafinen osaaminen jäädä osalla heikommaksi kuin mikä nykyaikaisen pelikehityksen kannalta voisi olla hyödyksi.

Kyseessä on kehittämistyö, jonka tarkoituksena on syventää pelikehityksen ja 3D-mallintamisen teoriaosaamista sekä vahvistaa aiempaa käytännön osaamista pelialan opiskelijoiden ja aiheesta kiinnostuneiden keskuudessa. Valmista opinnäytetyötä voidaan käyttää esimerkkinä yhdestä lähestymistavasta pelikehitykseen grafiikkatuotannon näkökulmasta joko sellaisenaan, osittain tai sovellettuna.

Opinnäytetyö on jaettu seitsemään päälukuun. Ensimmäisessä pääluvussa käsitellään pelikehitystä pintapuolisesti. Toinen pääluke keskittyy grafiikkatuotantoon. Kolmannessa pääluvussa käsitellään 2D-grafiikan historiaa ja perusasioita. Neljännessä pääluvussa syvennyttään 3D-grafiikkaan ja sen tuottamiseen käytettäviin mallinnusohjelmiin. Viidennessä pääluvussa käsitellään 3D-mallintamista ja sen tekniikoita. Kuudennessa pääluvussa käsitellään työn toteutuksen kulku vaiheittain. Seitsemäs pääluke sisältää pohdintaa opinnäytetyön suunnitelmanmukaisesta toteutumisesta.

2 PELIKEHITYS

Pelikehitys on ohjelmistotuotantoa, jossa yhdistyvät tiede ja taide. Pelikehityksellä tuotetaan videopelejä, jotka ovat ohjelmistoja, joissa on pelattava sisältö, kuva ja ääni. (Bethke 2003, 4.)

Videopelien kehityksellä on pitkä historia, joka on saanut alkunsa 1950-luvulla. Yksi ensimmäisistä videopeleistä on kanadalaisen insinöörin, Dr. Josef Katesin, Canadian National Exhibition -tapahtumaa varten kehittämä Bertie the Brain. Kyseessä on arcade-versio ristinollasta tietokonetta vastaan. Ensimmäinen reaaliaikainen graafinen tietokonepeli, Spacewar, ohjelmoitiin vuonna 1961. (Puhakka 2008, 25.)

Pelikehitys on prosessi, joka käsitteenä voi tarkoittaa yhtä tai useampaa pelikehityksen sisältämistä vaiheista. Pelikehityksessä voidaan hyödyntää monia valmiita ohjelmistotuotannon valmiita toimintamalleja.

Pelikehityksellä tarkoitetaan pelin valmistusprosessia. Kyseisen prosessin toteuttamiseen ei ole yhtä oikeaa toteutusvaiheiden sarjaa, mutta lyhykäisyydessään pelikehitys alkaa ideasta, jota jalostetaan. Ideaa kehitetään tapauskohtaisesti erilaisin menetelmin, joiden lopputuloksena on useimmiten valmis peli, joka pitää sisällään toiminnallisuuksia, grafiikkaa ja ääntä.

Pelien valmistusprosessiin käytetään nykyään useimmiten valmiita pelimoottoreita. Pelimoottorit ovat pelikehitystä nopeuttamaan kehitettyjä ohjelmistokehyksiä eli ohjelmia, joiden päälle pelinkehittäjät voivat rakentaa oman pelin. Pelimoottoreissa on yleisimmin valmis lähdekoodi, renderöintimoottori, fysiikkamoottori ja muita työkaluja. Yksi suosituimmista pelimoottoreista on Epic Gamesin Unreal Engine, jossa on myös pelimoottorin sisäisiä työkaluja 3D-mallintamiseen (Game Designing 2020). Useimmiten grafiikka kuitenkin tuotetaan pelimoottorin ulkopuolella.

3 GRAFIIKKATUOTANTO

Grafiikkatuotanto tarkoittaa grafiikan tuottamista. Tuotettu grafiikka voi olla mitä tahansa näkyvää tai ei-näkyvää, mutta useimmiten grafiikasta puhuttaessa tarkoitetaan jotakin visuaalista jollakin pinnalla. Tässä opinnäytetyössä käsitellään lähinnä tietokonegrafiikkaa ja sen tuotantoa. Tietokonegrafiikkaa on kaikki visuaalinen informaatio, jota tietokoneella käsitellään. Grafiikkatuotanto digitaalisesta grafiikasta puhuttaessa kattaa kaiken yhdestä pikselistä esimerkiksi animoituun 3D-malliin. (Takala & Savioja 2007.)

Tietokonegrafiikaksi rinnastettavaa grafiikkaa kehitettiin ensimmäistä kertaa sotilaskäyttöön 1940-luvun lopulla (Puhakka 2008, 25). Vuonna 1951 luodun Yhdysvaltalaisen SAGE-ilmapuolustusjärjestelmän tutkien signaali piirtyi vektorigrfiikkana käyttäjälle. Järjestelmän toimivuuden testausvaiheessa signaali tutkilta tietokoneille kuljetettiin puhelinlankoja pitkin (Lincoln Laboratory 2018).

Spacewar, yksi ensimmäisistä reaaliaikaisista graafisista tietokonepeleistä ohjelmoitiin vuonna 1961 (Puhakka 2008, 25). Syvyyspuskuritekniikan periaatteen ensimmäinen esiintyminen on dokumentoitu vuonna 1974. Syvyyspuskuritekniikalla (engl. Z-buffering) vertaillaan 3D-kuvan syvyyksiä ja piirretään renderöitävään kuvaan ainoastaan näkyvät kohteet (Computer Hope 2017).

Myöhemmin vektorinäytöt korvautuivat pikseleihin perustuvilla rasterinäytöillä. Tietokonegrafiikan käyttö teollisuudessa suunnittelun ja teknisen piirtämisen tukena yleistyi 1980-luvulla, kun Autodeskin AutoCAD tuli markkinoille. (Puhakka 2008, 26.)

3.1 Käyttötarkoitukset

Digitaalista tuotettua grafiikkaa valmistetaan maailmalla useita eri käyttötarkoituksia varten, muun muassa mainontaan, viihdekäyttöön ja opetustarkoitukseen. Tuotettua grafiikkaa voidaan käyttää sellaisenaan digitaalisena, mutta sitä voidaan myös painaa ja tulostaa käyttötarkoituksesta ja halutusta esitystavasta riippuen.

Opetustarkoituksessa digitaalisen grafiikan tarkoituksena on yleensä opettaa suoraan tai havainnollistaa esimerkein. Nykyaikainen oppimateriaali sisältää paljon kuvia, hyvänä esimerkkinä on oppikirjat, joissa usein tekstikappaleessa käydystä asiasta on jonkinlainen kuvallinen esitys.

Mainonnassa ja markkinoinnissa grafiikalla luodaan näkyvyyttä ja pyritään vetoamaan ihmisiin tavalla tai toisella. Yritysten mainokset ja logot voivat olla täysin digitaalisen grafiikan tuotantomenetelmillä toteutettuja. (Ford 2019.)

Viihdealalla digitaalisen grafiikan tarkoitus on yleisimmin viihdyttää näkijäänsä. Elokuva- ja filmitelollisuudessa digitaalisella grafiikalla voidaan luoda erilaisia visuaalisia tehosteita, joiden toteuttaminen oikeasti olisi mahdotonta tai kannattamatonta rahallisesti ja/tai turvallisuuden kannalta. (Okun & Zwerman 2010, 2–3.) Digitaalisella grafiikalla voidaan myös toteuttaa täyspitkiä animaatioelokuvia. Videopelitelollisuudessa kaiken visuaalisen ollessa digitaalista grafiikkaa muutamia poikkeuksia lukuun ottamatta on digitaalisesti tuotettu grafiikka olennainen osa videopelien kokonaisuutta (Gamespot 2001).

3.2 Suunnittelu

Menestyksekkäiden teosten ja asiakkaille toteutettujen graafisten teosten taustalla on useimmiten jonkin asteista suunnittelua. Graafisen suunnitteluprosessin vaiheet vaihtelevat tekijöittäin, mutta tapauskohtaisesti toimivaksi todettuja suunnitteluprosessin vaihemalleja on myös olemassa.

Suunnittelutyötä tehdään yleisimmin ongelman ratkaisemiseen tai tarpeen täyttämiseen. Asiakas tarvitsee esimerkiksi videopelin toteuttamista varten 3D-malleja. Edellä mainitun tarpeen täyttämisen suunnittelussa on hyvä ottaa huomioon tyyli ja ulkoasu, käyttötarkoitus ja tiedostoformaatti, kohdeyleisö ja sen tuomat edellytykset, käytettävissä oleva aika, tarvittavat resurssit sekä muut asiakkaan vaatimukset ja toivomukset. Tätä voidaan kutsua suppeaksi vaatimusmäärittelyksi. Vaatimusmäärittelyä käytetään paljon ohjelmistotuotannossa, mutta se on myös sovellettavissa muuhun asiakaslähtöisen työn suun-

nitteluun. Suunnittelun tärkeys korostuu entisestään 3D-grafiikkaa suunniteltaessa, koska 3D-grafiikan tuleva käyttötarkoitus tuo mukanaan tekijöitä, jotka kannattaa ottaa huomioon jo grafiikan luomisen aloitusvaiheessa.

Tyyli voi olla esimerkiksi taidesuuntaukseltaan realistinen tai surrealistinen. Varsinainen ulkoasu ja piirrostyylit määräytyvät monesti taiteilijan tai tarpeen mukaan. (Taimisto s.a.)

Käyttötarkoituksen vaikutukset suunnitteluvaiheessa vaihtelevat käyttötarkoituksittain. Edellä mainitussa skenaariossa, jossa asiakkaalle mallinnetaan 3D-malleja videopelin toteuttamista varten, olisi hyvä tietää alusta jolle peliä ollaan toteuttamassa. Pääsääntöisesti peliä varten mallinnettaessa on hyvä käytäntö pitää polygonien määrä mahdollisimman alhaisena. Alhaisen polygonien määrän ja tekstuurien resoluution huomioonottamisen tärkeys korostuu silloin, kun tehdään mobiilipeliä (Android developers 2018). Mobiililaitteilla ei ole ainaakaan vielä tänä päivänä vastaavanlaista suoritustehoa kuin esimerkiksi nykyaikaisilla Windows-tietokoneilla. Tämä johtuu osittain mobiililaitteiden asettamista rajoitteista, kuten tilasta, johon grafiikkasuoritin tulee saada mahtumaan. Tilan säästämiseksi mobiililaitteissa käytetään yleisimmin passiivisia jäähdytysratkaisuja komponenttien jäähdyttämiseen, jotka eivät ole optimaalisia korkeatasoisten videopeligrafiikoiden toistamiseen. Mobiililaitteissa suorittimet toimivat myös matalalla jännitteellä mahdollisten ylikuumenemistapausten ennaltaehkäisemiseksi.

Tiedostoformaatti määräytyy useasti kohdeohjelman mukaan, johon tiedostoa ollaan tekemässä. Kaikilla ohjelmilla ei pystytä luomaan kaikkia tiedostoformaatteja. On olemassa ohjelmia, jotka pystyvät muuntamaan tiedostoja formaatista toiseen, mutta usein on järkevintä luoda tiedosto alusta lähtien oikeaan formaattiin, jos se vain on mahdollista. Tiedostoformaatti vaikuttaa myös kuva- ja äänitiedostoissa pakkaukseen, joka vaikuttaa laatuun ja lopulliseen tiedostokokoon (Oracle 2020).

Kohdeyleisö kannattaa ottaa huomioon jo käytettävän tyylin valitsemisessa. Kohdeyleisössä huomioitavat tekijät voivat olla yleiset mieltymykset, ikä ja sukupuoli. Kohdeyleisön asettamat edellytykset voivat olla grafiikan sisällön sopivuus kohdeyleisön ikäryhmälle (Christensen 2015).

Käytettävissä oleva aika vaikuttaa esimerkiksi työn laatuun ja yksityiskohtien määrään, joita ehditään toteuttamaan määräaikaan mennessä. Joissakin tapauksissa voi olla järkevää käyttää osittain tai kokonaan valmiita asetteja käytettävissä olevan ajan riittämisen takaamiseksi asioihin, joihin ei ole olemassa valmiita ratkaisuja (Workerbee 2018).

Suunnittelussa kannattaa huomioida myös käytössä olevat ja tarvittavat resurssit. Tällaisia resursseja voi olla muun muassa tietokone, kamera, piirto-pöytä ja internetyhteys (Ellis, 2020).

Grafiikkatuotannon tuottavuuden näkökulmasta tulisi katsoa myös kokonaisuutta. Taiteessa voi olla järkevää panostaa yksityiskohtiin lähes loputtomiin, mutta esimerkiksi peliprojektissa se ei aina ole kannattavaa. Yksityiskohtiin tarkertuminen pelintekoprosessin alkuvaiheessa saattaa olla turhaa työtä, koska peliprojekteissa muutoksia tulee tapahtumaan. Pahimmassa tapauksessa peliprojektin edistyminen loppuu kokonaan (Blackman 2013, 12–13).

4 2D-GRAFIikka

2D-grafiikka on rasteri- tai vektorigrafiikka tekniikoilla tuotettua digitaalista kuvaa. 2D-grafiikkaa ja sen sovellutuksia näemme tänä päivänä kaikkialla. Mainikkain rasterigrafiikan muokkausohjelma on Adobe Photoshop ja vektorigrafiikalla taas käyttötarkoituksesta riippuen joko Autodesk AutoCAD tai Adobe Illustrator. (TechTerms 2009.) Kuvassa alhaalla (Kuva 1) on pisteen suurenos, jossa nähdään vektori- ja rasterigrafiikan rakenteellisia eroavaisuuksia.



Kuva 1. Vektorigrafiikkaa ja rasterigrafiikkaa vierekkäin niiden erojen havainnollistamiseksi

2D-grafiikkaa käytetään pelituotannossa kaiken suunnitteluun hahmoista kenttiin, konseptitaiteeseen ja mainontaan sekä joskus kokonaisten pelien toteuttamiseen (Bethke 2003, 47).

4.1 Vektorigrafiikka

Vektorigrafiikka koostuu ankkuripisteistä, joilla on jonkinlainen yhteys, kuten esimerkiksi viiva tai katkoviiva. Vektorigrafiikkaa käytetään useimmissa CAD- ja 3D-mallinnusohjelmissa sen ominaisuuksien, muun muassa tarkkuuden ja skaalattavuuden vuoksi. Tarkkuutensa ja skaalattavuutensa vuoksi vektorigrafiikkaa käytetään myös paljon yritysten logoissa ja erilaisissa mainosgraafikoissa. Vektorigrafiikan käyttäminen sellaisenaan vektorigrafiikan piirto-ohjelman ulkopuolella on hyvin harvinaista. Vektorigrafiikka muunnetaan useimmiten bittikarttagrafiikaksi ennen myöhempää käyttöä hyödyntämällä erilaisia rasterointitekniikoita (Vector Conversion s.a.).

4.2 Rasterigrafiikka

Rasterigrafiikka koostuu pikseleistä, joihin voi tallentaa värin. Rasterigrafiikkaa kutsutaan myös bittikarttagrafiikaksi. Useimmissa käyttötarkoituksissa käytetään nykyään pääsääntöisesti rasterigrafiikkaa. Lähes kaikki tietokonegra-

fiikka, jota me näemme, on joko rasterigrafiikkaa tai rasteroitua vektorigrafiikkaa (Puhakka 2008, 26, 153). Tämä johtuu siitä, että nykyaikaiset näytöt koostuvat kuvapisteistä, eli pikseleistä, eivätkä täten pysty toistamaan vektoreita sellaisenaan täydellisesti (Eck 2010, 1–2).

Rasterigrafiikan tuottamiseen soveltuvia ohjelmia Photoshopin lisäksi on esimerkiksi ilmaisohjelmat GIMP ja Paint.NET. (Lansdown 2019, 181.) Rasterigrafiikkaa käytetään myös pelien grafiikan toistamiseen.

3D-pelituotannossa rasterigrafiikkaa käytetään 3D-grafiikan värittämiseen tekstuurein, muihin tekstuurikarttoihin, erilaisiin partikkeliefekteihin ja valikoiden ulkoasuun. 3D-peleissä voidaan myös käyttää 2D-spritejä sellaisenaan (Bethke 2003, 48).

5 3D-GRAFIikka

Kolmiulotteinen tietokonegrafiikka eli 3D-grafiikka on tietokoneella tuotettua grafiikkaa, jonka geometria on esitettävissä kolmiulotteisessa tilassa eli 3D-avaruudessa.

3D-grafiikkaa käytettiin ensimmäisen kerran elokuvatuotannossa vuonna 1973 elokuvassa nimeltä Futureworld. Vuonna 1995 Pixar Animation Studios loi maailman ensimmäisen täyspitkän animaatioelokuvan, Toy Storyn. (Puhakka 2008, 26–27.)

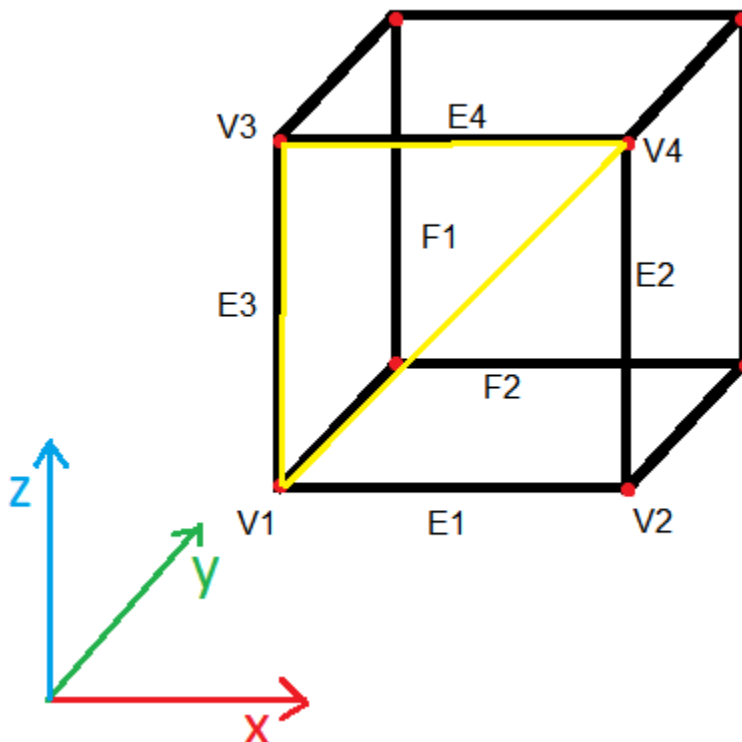
Atari julkaisi vuonna 1980 maailman ensimmäisen 3D-grafiikalla toteutetun arcade-pelin, Battlezonen. Battlezonessa grafiikka toteutettiin vielä vektoreilla, joka toistettiin vektorinäytöllä (Andys arcade 2018). Vaikka vuoden 1993 julkaistu Doom ei ollut 3D-grafiikalla toteutettu peli, sitä pidetään tärkeänä osana 3D-pelaamisen kehitystä ja osatekijänä 3D-grafiikan yleistymiseen videopelitaloudessa (Overmars 2012).

5.1 Monikulmioverkko

Tyypillisesti 3D-grafiikan eli kolmiulotteisten objektien pinnat esitetään monikulmioverkkoina. Monikulmioverkko koostuu vertekseistä eli pisteistä ja edgeistä eli särmistä. Verteksillä on sijainti 3D-avaruuden $[x, y, z]$ -koordinaatissa, esimerkiksi verteksi $V_1 = [0, 0, 0]$.

Kolmiulotteisen kuution toteuttamiseksi tarvitaan monikulmioverkkoon yhteensä 8 verteksiä. Olkoon kuution verteksien koordinaatit: $V_1 = [0, 0, 0]$, $V_2 = [50, 0, 0]$, $V_3 = [0, 50, 0]$, $V_4 = [50, 50, 0]$, $V_5 = [0, 0, 50]$, $V_6 = [50, 0, 50]$, $V_7 = [0, 50, 50]$ ja $V_8 = [50, 50, 50]$.

Yksi tapa monikulmioverkon esittämiseen on määrittää jokainen särmä E erikseen reunaviivalistaan. Kuution reunaviivalista olisi esimerkiksi: $E_1 = [V_1, V_2]$, $E_2 = [V_2, V_4]$, $E_3 = [V_3, V_1]$, $E_4 = [V_4, V_3]$, ja niin edelleen, kunnes kaikki kuution reunaviivat ovat listassa (Kuva 2). (Puhakka 2008, 50–53.)



Kuva 2. Kuution kylki, jossa on monikulmioverkon verteksipisteet, särmät ja tahkot havainnollistettuna

Tapa, jolla Autodesk 3ds Maxin MAXScript listaa 3D-objektin geometrian, sisältää verteksilistan lisäksi face- eli tahkolistan reunaviivalistan sijaan. Tahkot

ovat pääsääntöisesti kolmen tai neljän verteksipisteen välille muodostettava taso tai pinta. Tahkot muodostavat 3D-objektin näkyvän pinnan, jonka pinnalle voi heijastaa esimerkiksi tekstuurikarttoja. Kuutiossa on yhteensä 6 sivua. MAXScript käsittelee geometrian tahkoja kolmioina jakaen kaikki sivut automaattisesti kolmioiksi, eli kuutiossa on yhteensä 12 tahkoa. Esimerkki MAX-Scriptin tahkojen, F -listasta: $F_1 = [V_1, V_3, V_4]$, $F_2 = [V_4, V_2, V_1]$, $F_3 = [V_5, V_6, V_8]$, $F_4 = [V_8, V_7, V_5]$, $F_5 = [V_1, V_2, V_6]$, $F_6 = [V_6, V_5, V_1]$, $F_7 = [V_2, V_4, V_8]$, $F_8 = [V_8, V_6, V_2]$, $F_9 = [V_4, V_3, V_7]$, $F_{10} = [V_7, V_8, V_4]$, $F_{11} = [V_3, V_1, V_5]$ ja $F_{12} = [V_5, V_7, V_3]$ (Autodesk 2013).

5.2 Mallinnusohjelmat

3D-mallinnusohjelmilla voidaan luoda uusia, muokata tai poistaa olemassa olevia monikulmioverkkoja. Suurimmalla osalla mallinnusohjelmista on mahdollista toteuttaa vähintään jonkin asteista teksturointia. Useimmilla 3D-mallinnusohjelmilla on myös mahdollista toteuttaa animaatioita, ja sen edellyttämiä toimenpiteitä. Tässä opinnäytetyössä käytetään Autodeskin ammattikäyttöön tarkoitettua 3ds Max -mallinnusohjelmaa.

3ds Max on monikäyttöinen ja suosittu mallinnusohjelma niin arkkitehtuurissa sovellutuksissa, kuin videopelien 3D-mallien toteuttamisessa. Ohjelma valikoitui käytettäväksi opinnäytetyössä, koska ohjelman käyttöä on opetettu Kaakkois-Suomen Ammattikorkeakoulun Peligrafiikka ja mallinnus -kursilla. Ohjelma sisältää kaiken välttämättömän monikulmioverkkojen mallintamisesta renderöintiin. Lisäksi käyttäjän on mahdollista luoda uusia liitännäisohjelmia 3ds Maxin MAXScript -komentosarjakieltä käyttäen. Ohjelma tukee myös useimpia 3D-tiedostoformaatteja. 3ds Maxia pidetään oppimiskynnykseltään helpommin lähestyttävänä 3D-mallinnusohjelmana kuin esimerkiksi Blenderiä.

Blender on avoimen lähdekoodin 3D-mallinnusohjelma. Blender sisältää mallintamiseen ja animoimiseen liittyvien toiminnallisuuksien lisäksi pätevät veistämiseen ja tekstuureiden maalaamiseen tarkoitetut työkalut.

Veistämällä tarkoitetaan monikulmioverkon geometrian muokkausta erilaisin työkaluin, joilla voi esimerkiksi kohottaa, laskea ja siirtää osoittimen määrittämän alueen geometriaa. Veisto-ohjelmia on mahdollista käyttää monikulmiomallintamisen sijaan tai sen rinnalla. Veistäminen on verrattavissa saven muotoilemiseen. Veisto-ohjelmista tunnetuin kaupallinen ohjelma on Pixologicin ZBrush.

ZBrush sisältää kaikki tarvittavat työkalut mallin aloittamisen muotoilusta viimeistelyyn ja lopulta teksturoimiseen. ZBrushissa on mahdollista lisätä yksityiskohtia ja monikulmioverkon tahkojen määrää kerroksittain.

Sculptris on myös Pixologicin veisto-ohjelma, joka on ilmainen. Sculptris sisältää suppeamman valikoiman työkaluja ja toimintoja kuin ZBrush. (Lansdown 2019, 179.)

Autodesk MudBox on Autodeskin kaupallinen veisto-ohjelma, jota verrataan usein Pixologicin ZBrushiin. MudBoxissa ja ZBrushissa on pääpiirteittäin samat toiminnallisuudet. ZBrushissa on hieman laajempi valikoima oletustyökaluja ja siveltimiä, mutta MudBoxissa käyttäjäystävällisempi käyttöliittymä. (Lansdown 2019, 176.) MudBox toimii myös erittäin hyvin yhdessä muiden Autodeskin tuoteperheeseen kuuluvien ohjelmistojen kanssa (All3DP 2020a).

5.3 Apuohjelmat

Hyödyllisiä apuohjelmia 3D-mallin valmistamisen eri vaiheisiin on paljon. SubstancePainter on lähes kaikkien isojen pelifirmojen käyttöön omaksuma tekstuurien luomiseen ja muokkaamiseen tarkoitettu ohjelma (Mobsby 2019).

ArmorPaint on Armoryn kehittämä vaihtoehto SubstancePainterin kaltaisille teksturoimiseen tarkoitetuille ohjelmille. Ohjelma perustuu avoimeen lähdekoodiin, joka on saatavissa GitHubista. Ohjelmasta on myös mahdollista ostaa valmiiksi käännetty versio.

Diffuusiokarttojen elävöittämiseen voidaan käyttää normaalikartoitusten luomiseen tarkoitettuja apuohjelmia, kuten CrazyBump tai MindTex 2. Apuohjelma

tallentaa normaalikartan pikseleihin suuntatiedon, joka toteutetaan RGB-väreillä. Normaalikartta voidaan luoda esimerkiksi 3D-objektin diffuusiokartasta. Normaalikartalla voidaan lisätä yksityiskohtia, kuten kohoumia tai uurteita 3D-mallien pinnoille. (Polycount 2018.)

Mixamo on Adoben selainpohjainen apuohjelma, jolla on mahdollista luoda 3D-mallinnetusta humanoidista luurankomalli, johon ohjelma sitoo automaattisesti monikulmioverkon vertekspisteet. Mixamossa on myös lukuisia valmiita animaatioita, joita pystyy muokkaamaan ja lataamaan omaan käyttöönsä. Mixamo on ilmainen ohjelma. Ei-selainpohjaisia vaihtoehtoja Mixamolle on SmartBody ja Rigify -ohjelmat (All3DP 2020b).

6 3D-MALLINTAMINEN

3D-mallintaminen on luomistyötä, jolla tarkoitetaan kolmiulotteisten mallien luomista tai muokkaamista. Kolmiulotteinen malli voi koostua yhdestä tai useammasta monikulmioverkosta. (Murdock 2004, 335.)

Valmiiden 3D-mallien käyttötarkoitukset lisääntyvät 3D-mallinnus- ja tulostustekniikoiden kehittyessä. 3D-malleja on käytetty vuosikymmeniä tuotesuunnitteluun ja arkkitehtuurillisiin käyttötarkoituksiin. Ensimmäisiä suuria harppauksia 3D-mallintamisen historiassa on ollut 1960-luvulla markkinoille tulleet CAD-järjestelmät (Computer Aided Design).

Nykyään 3D-mallintaminen on tuotu harrastelijoidenkin käden ulottuville ilmaisten 3D-mallinnusohjelmien, kuten Blenderin myötä. 3D-mallinnusohjelmien sekä 3D-tulostimien kehittyminen on saanut aikaan jonkin asteista 3D-mallinnusosaamisen yleistymistä, joka näkyy saatavilla olevien 3D-tulostettavien objektien laajassa kirjossa (UFO 3D 2019).

Pelien 3D-grafiikka käy läpi monivaiheisen luomisprosessin, johon on useita erilaisia toteutustapoja. Peleissä käytetyt 3D-mallit voivat olla joko animoimattomia tai animoituja. Tässä opinnäytetyössä käydään läpi vaiheittain animoidun 3D-mallin tekoprosessin kaikki vaiheet opinnäytetyötä varten valittuja toteutustapoja noudattaen.

6.1 Tyylit ja tekniikat

Tulevan 3D-mallin mukaan on tehokkainta valita tapauskohtaisesti oikea lähestymistapa ja tekniikka. Autodeskin 3ds Maxissa on mahdollista käyttää useita eri mallinnustekniikoita. Mallinnustekniikalla tarkoitetaan monikulmion muokkaamista sen tietyssä tilassa, joita on editable poly, mesh, patch ja NURBS. (Murdock 2004, 863.)

6.1.1 Editable Poly

Editable Poly -objekti on muokattava monikulmio-objekti, jolla on 5 erilaista valintatilaa. Käytettävissä olevat valintatilat ovat verteksi-, särmä-, reuna-, tahko- ja elementtivalintatila. Editable Poly -objektia ei luoda, vaan se tuodaan tai muutetaan jostain olemassa olevasta objektista. (Murdock 2004, 408.)

Editable Poly -tilassa työskennellään pääsääntöisesti nelikulmioilla, jotka koostuvat kahdesta kolmiosta, jotka koostuvat yhteensä neljästä verteksipisteestä. Editable Poly -tilassa ei ole kuitenkaan rajoitteita tahkojen enimmäisverteksimäärälle.

Editable Poly -tilassa on paljon erilaisia työkaluja, joilla voidaan muokata monikulmion geometriaa. Editable Poly -tila on myös mallinnustiloista käytetyin sen monikäyttöisyyden ja muunneltavuuden ansiosta. Polygon -objektit on rakenteeltaan ”joustavampia” sen neliöistä koostuvan rakenteen ansiosta kuin esimerkiksi Editable Mesh -objektit. Tästä syystä se on helpompi hallittava esimerkiksi animoitaessa niin, että objekti taipuu luonnollisesti ja sulavasti halutulla tavalla. (Murdock 2004, 390.)

6.1.2 Editable Mesh

Editable Mesh -tila on peruseräiltään samankaltainen Editable Poly -tilan kanssa. Kun Editable Poly -tilassa työskennellään pääsääntöisesti nelikulmioilla, työskennellään Editable Mesh -tilassa kolmioilla, jotka koostuvat kolmesta verteksipisteestä. Editable Mesh -objektia ei luoda, vaan se tuodaan tai muutetaan jostain olemassa olevasta objektista.

Editable Mesh -tilassa on vähemmän käytettävissä olevia työkaluja kuin Editable Poly -tilassa. Smoothaukseen liittyvät modifier-työkalut eivät saa aikaan yhtä hyviä lopputuloksia Editable Poly -tilaan verrattuna. (Murdock 2004, 391.)

6.1.3 Editable Patch

Editable Patch -objektit ovat helppokäyttöisiä, mutta niiden litteän olemuksen takia niillä on erittäin rajallinen toiminnallisuus. Paras tapa käyttää Editable Patch -toimintoja on muuttaa jokin olemassa oleva objekti Editable Patch -muotoon. Patchit perustuvat splini-käyriin, joita hallitaan käyrillä olevista kontrollipisteistä. Editable Patch -objekteilla on paljon erilaisia käytettäviä työkaluja, joilla muokata niitä. (Murdock 2004, 432.)

6.1.4 NURBS

NURBSilla on yhtäläisyyksiä patch-tilan kanssa. NURBS-objektia muokataan B-käyrällä sijaitsevia kontrollipisteitä hallitsemalla. NURBS tulee sanoista Non-Uniform Rational Basis Spline. NURBS-objektit on helppokäyttöisiä kontrollipisteistä muokattavien splinien hyvästä hallittavuudesta ja toisiinsa sulautuvuudesta johtuen. NURBS-pinnat ovat sileitä ja käytännöllisiä mallittaessa orgaanisia kohteita. (Murdock 2004, 336.)

6.2 Modifier-työkalut

Modifier-työkaluilla voi muokata 3D-objektia. Modifier-työkaluilla on omat tehtävänsä, niin kuin oikeillakin työkaluilla. Modifier-työkaluilla pystyy muokkaamaan objektin geometriaa, lisäämään materiaalikartoituksia, vääristää pintoja, ja niin edelleen.

Modifier-työkaluja voidaan lisätä muokattavalle objektille esimerkiksi Command panel -valikon Modifier List -pudotusvalikosta valitsemalla. Objektilla voi olla useita modifier-työkaluja samaan aikaan aktiivisena objektin modifier-pinossa (eng. modifier stack). Modifier-työkalujen järjestys modifier-pinossa määrittää järjestyksen, jossa modifier-työkalut vaikuttavat, jolla on merkitystä objektin lopullisen ulkomuodon kannalta. (Murdock 2004, 287–288.)

6.3 Topologia

Monikulmioverkon topologialla tarkoitetaan rakennetta, josta mallin pinta muodostuu. Pinta voi muodostua kolmioista, nelikulmioista, n-kulmioista tai niiden yhdistelmästä. Hyvän monikulmioverkon topologia sisältää ainoastaan kolmioita ja nelikulmioita. Tämä johtuu siitä, että grafiikkasuorittimet käsittelevät monikulmioverkkoja verteksipisteinä ja kolmioina. Kaikki monikulmioverkon nelikulmiot ja n-kulmiot muutetaan automaattisesti renderöimisen yhteydessä kolmioksi. N-kulmiot muuntuvat kolmioiksi arvaamattomalla tavalla, josta voi aiheutua ongelmia animoimisessa ja sävyttimien toiminnassa. (Thorn 2017, 13–14.)

Nelikulmioista koostuva topologia on usein helppolukuinen ja siisti. Nelikolmioista koostuva topologia muutetaan myös automaattisesti kolmioiksi renderöimisen yhteydessä, mutta ei niin arvaamattomasti, koska nelikulmion puolittava automaattisesti muodostettu särmä voi jakaa nelikulmion kolmioiksi vain kahdella tavalla. Nelikulmiot voi muuttaa kolmioiksi myös etukäteen käsin tai käyttämällä siihen tarkoitettuja modifier-työkaluja. Nelikulmiot jakautuvat tarvittaessa siististi pienemmiksi nelikulmioiksi suuremman tarkkuuden saavuttamiseksi säilyttäen samalla alkuperäisen monikulmioverkon muodot. Nelikulmioista koostuvat monikulmioverkot vääntyvät animoitaessa pääsääntöisesti siistimmin kuin esimerkiksi kolmioista koostuvat monikulmioverkot.

Kolmioista koostuvat monikulmioverkot kelpaavat grafiikkasuorittimelle sellaisenaan, joten mitään automaattisia muunnoksia ei tapahdu. Kolmiosta koostuvia monikulmioverkkoja on kuitenkin hankalampi työstää kuin esimerkiksi nelikulmioista koostuvia monikulmioverkkoja. Animoitaessa kolmioista koostuvilla monikulmioverkoilla on taipumusta muodostaa teräviä kulmia taivekohtiin. Tästä syystä animoitavissa monikulmioverkoissa käytetään yleisimmin nelikulmioita mallin pinnan muodostamiseen. (Turbosquid 2007.)

6.4 Animoimattomat mallit

Pelikehityksessä käytettävistä malleista suurin osa on animoimattomia. Animoimattomissa 3D-malleissa on useimmiten vain geometria ja tekstuuri. Tällaisia malleja ovat esimerkiksi pelikenttä ja kentälle sijoitetut paikallaan olevat objektit. Vaikka 3D-malli on animoimaton itsessään, se ei poissulje mahdollisuutta, etteikö sitä voisi manipuloida pelimoottorissa monin tavoin. Myös tekstuureilla voi luoda animoimattomiin malleihin lisää eloa tai illuusion liikkeestä.

3D-mallintaminen aloitetaan jostakin yksinkertaisesta monikulmiosta, johon käytetään yhtä tai useampaa mallinustekniikkaa halutun lopputuloksen saavuttamiseksi. Mallintaa voi esimerkiksi referenssikuvan pohjalta tai ilman. (Murdock 2004, 127.) Tässä opinnäytetyössä mallintaminen tapahtuu referenssikuvan pohjalta.

7 PELIHAHMON TOTEUTTAMINEN

Kolmiulotteisen pelihahmon toteutuksen vaiheet ovat yksinkertaistetusti monikulmioverkon luominen, teksturoiminen, luuanimaatioiden toteuttaminen ja edellä mainittujen vaiheiden edellyttämät toimenpiteet. Tässä opinnäytetyön osuudessa käydään läpi tätä opinnäytetyötä varten luodun animoidun pelihahmon toteuttamisen vaiheet suunnittelusta pelimoottoriin viemiseen.

Kolmiulotteisen pelihahmon mallintaminen on hyvä aloittaa jostain ideasta. Idea voi olla esimerkiksi jotain mitä näkee edessään tai jokin mielikuvituksen tuote. Pelialalla on tyypillistä visualisoida ja kehittää ideaa muun muassa konseptitaidteen muodossa. Konseptitaidetta ovat visuaaliset suunnitelmat ja luonnokset, joita voidaan myös kutsua konseptikuviksi.

7.1 Suunnittelu

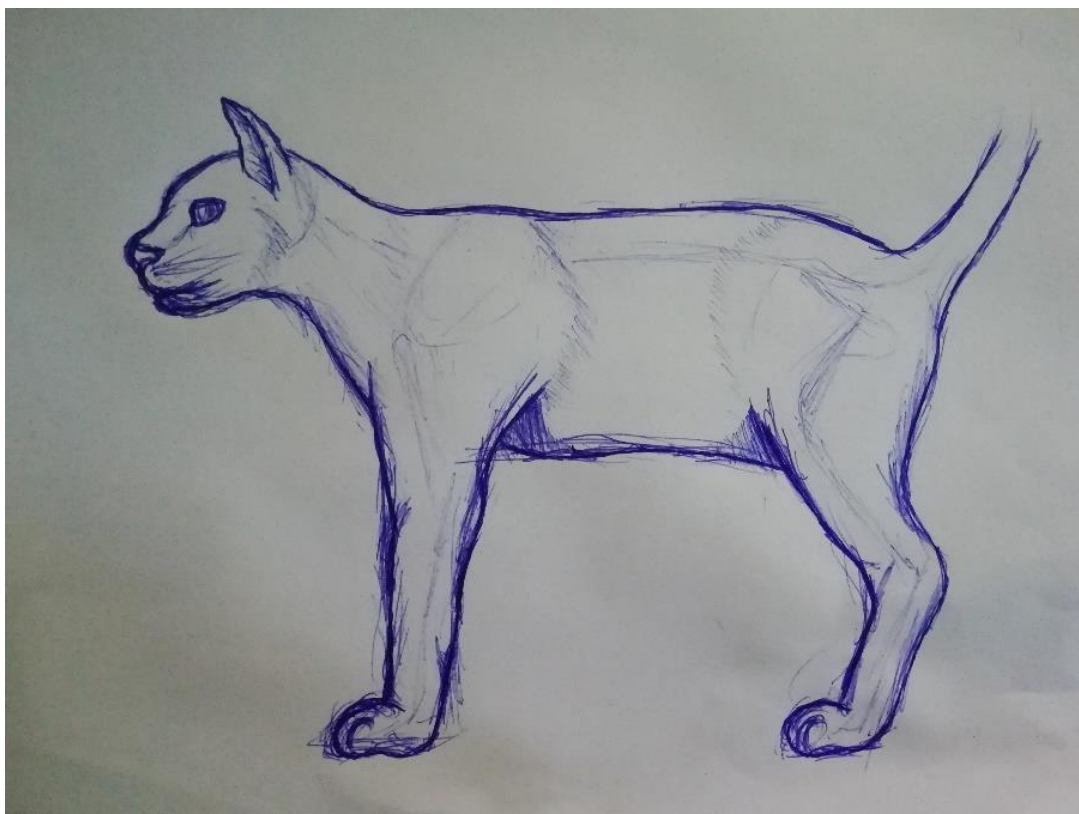
Opinnäytetyön suunnitteluvaiheessa päädyttiin käyttämään mallinnusesimerkinä teksturoitua ja animoitua nelijalkaista hahmoa, sillä tällaisen hahmon mallintamiseen ja animointiin ei ole olemassa yhtä kattavia valmiita työkaluja kuin bipedaalisen hahmon mallintamiseen. Työn aikana mallinnettavaksi hahmoksi valittiin kotikissa, jonka kompleksisuutta mallinnettavana ja animoituna hahmona lisäävät nelijalkaisuus, taipuisa luurakenne ja häntä.

7.2 Referenssikuva

3D-mallintamisen tukena käytettävä referenssikuva on kuva mallinnettavasta kohteesta, jota voidaan käyttää tarkkojen kohdetta vastaavien muotojen saavuttamiseksi tai inspiraation lähteenä. Referenssikuva voi olla muun muassa valokuva tai piirros. (Murdock 2004, 84.)

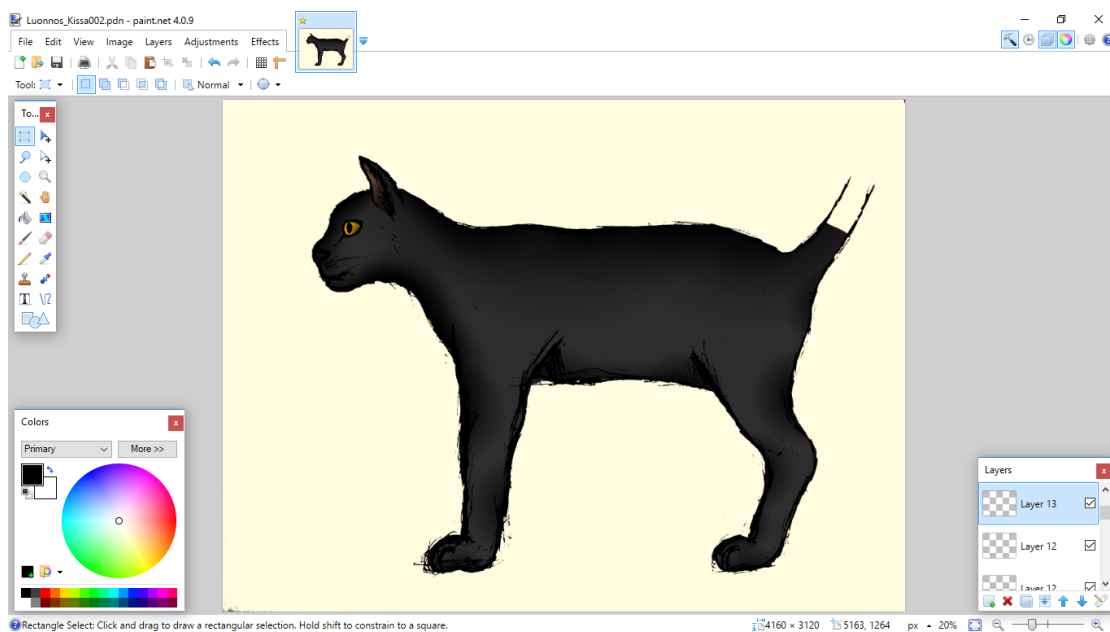
Työn toteutusvaihe alkoi luonnostelemalla fiktiivinen kuva kissasta tavalliselle A4-kokoiselle tulostuspaperille (Kuva 3). Tässä vaiheessa luonnos eli viimeistelemätön konseptikuva toimii pohjana referenssikuvalle, jota voidaan myöhemmin käyttää 3D-mallin mallintamisen tukena. Konseptikuvien tekeminen ei ole pakollinen vaihe onnistuneeseen 3D-mallintamiseen, mutta jos tavoitteena

on realistisen näköinen elävä olento, saattaa konseptikuvan olemassaolo helpottaa esimerkiksi mittasuhteiden ja luurangon rakenteen kanssa työskentelyä. Jonkinasteinen kohdeolennon anatomian tuntemus voidaan katsoa eduksi. Kuvan luonnokseen on hahmoteltu tässä työssä myös luuranko, niin kuin se voisi valmiissa mallissa olla.



Kuva 3. Luonnos kissasta

Konseptikuvasta otettiin valokuva, joka siirrettiin tietokoneelle muokattavaksi. Konseptikuvaa muokattiin ja kehitettiin paremmaksi Paint.NET-kuvankäsittelyohjelmalla. Näin tulevan 3D-mallin muotoja ja mittasuhteita oli parempi hahmottaa. Mahdolliset uudet versiot konseptikuvasta kannattaa toteuttaa työn alkuvaiheessa, mikäli kuvaa aiotaan käyttää suorana referenssinä 3D-mallin geometrian luomiseen.



Kuva 4. Väritetty konseptikuva

Konseptikuva voidaan värittää suuntaa antavasti, jotta kuvanmukainen lopputulos vastaa näkemystä mallinnettavasta eläimestä (Kuva 4). Konseptikuvan käytettävyyttä referenssikuvana voi parantaa käyttämällä esimerkiksi Paint.NETin Outline -efektiä, jolla voidaan selkeyttää referenssikuvan ääri viivoja (Kuva 5).



Kuva 5. Valmis referenssikuva

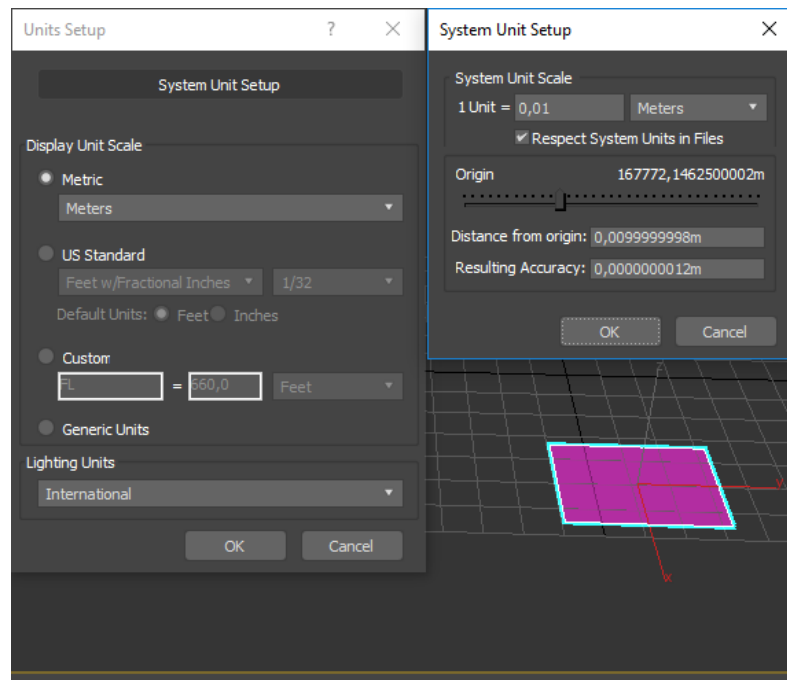
Valmiin referenssikuvan tallentamisen mahdollistamiseksi png -muodossa tulee kuvan kaikki tasot yhdistää yhdeksi tasoksi, mikäli tasoja on useampi kuin yksi. Useiden tasojen yhdistäminen yhteen tasoon tapahtuu litistämällä.

Paint.NETissä litistämisen voi toteuttaa esimerkiksi Ctrl + Shift + F -pikanäppäinyhdistelmällä. Litistämisen jälkeen kuva tallennetaan mallinnusohjelmaan viemistä varten.

7.3 Esivalmistelut mallinnusohjelmassa

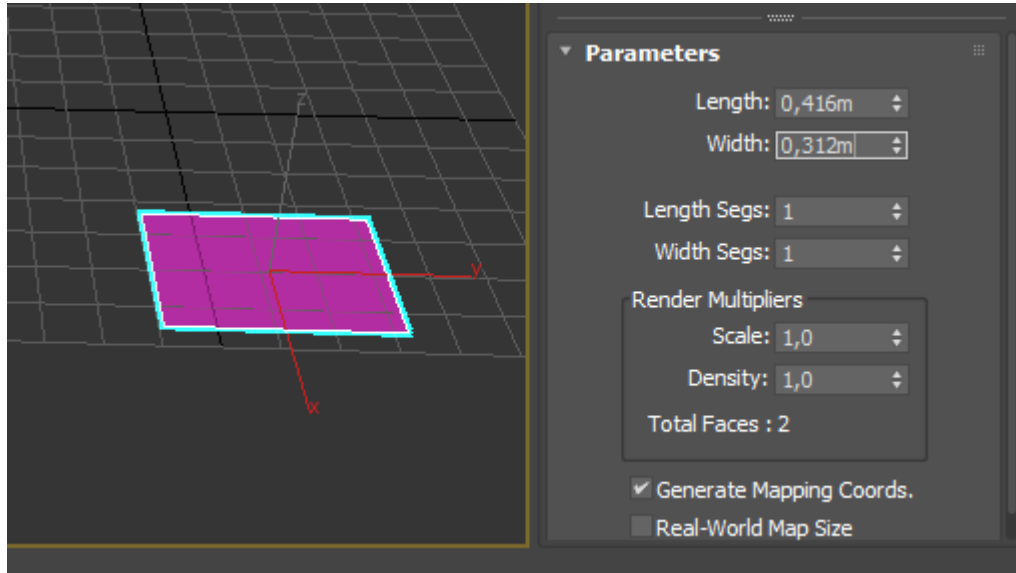
Pelihakmon mallintamista varten mallinnusohjelmassa tehtäviä esivalmisteluja on referenssikuvan tuomisen lisäksi pituuden mittayksikön vaihtaminen tuumista metreiksi. Pituuden mittayksikön vaihtaminen ei ole pakollista, mutta se helpottaa objektien mallintamista reaali-koossa. Väärästä mittayksiköstä saattaa koitua skaalautuvuusongelmia esimerkiksi mallinnettujen monikulmioverkkojen ja luuanimaatioiden pelimoottoriin viemisessä sekä fysiikkamoottorin törmäystarkastelussa.

Pituuden mittayksikön vaihtamiseksi avataan Autodesk:n 3ds Max -mallinnusohjelman Customize-välilehdestä Units Setup -ikkuna. Units Setup -ikkunasta valitaan Display Unit Scale -valikosta Metric ja pudotusvalikosta Meters. Seuraavaksi avataan System Unit Setup -ikkuna, jossa asetetaan yksi mittayksikkö vastaamaan yhtä metrin sadasosaa (Kuva 6).



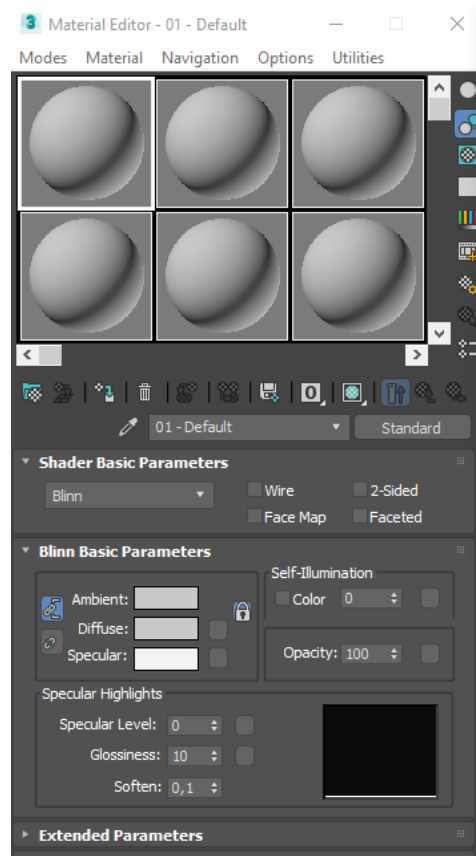
Kuva 6. Unit Setup- ja Display Unit Setup -ikkunat avoinna

Referenssikuvan sijoittamiseksi viewport-ikkunaan mallintamista varten luodaan taso-objekti (engl. plane). Tason leveydeksi ja pituudeksi määritetään referenssikuvan resoluutio metrin tuhannesosina eli millimetreinä (Kuva 7). Taso-objektin ja referenssikuvan leveyden ja pituuden ollessa suhteessa samat varmistetaan referenssikuvan sopivuus taso-objektin pinnalle.



Kuva 7. Referenssikuvan kokoinen taso-objekti

Referenssikuvan sijoittamiseksi viewport-ikkunassa näkyvän taso-objektin pinnalle tehdään referenssikuvasta materiaali. Materiaali tehdään Compact Material Editor -ikkunassa, joka avataan 3ds Maxin Rendering-välilehdestä. Ikkunassa valitaan materiaalipaikka, johon referenssikuvan materiaali tehdään (Kuva 8).

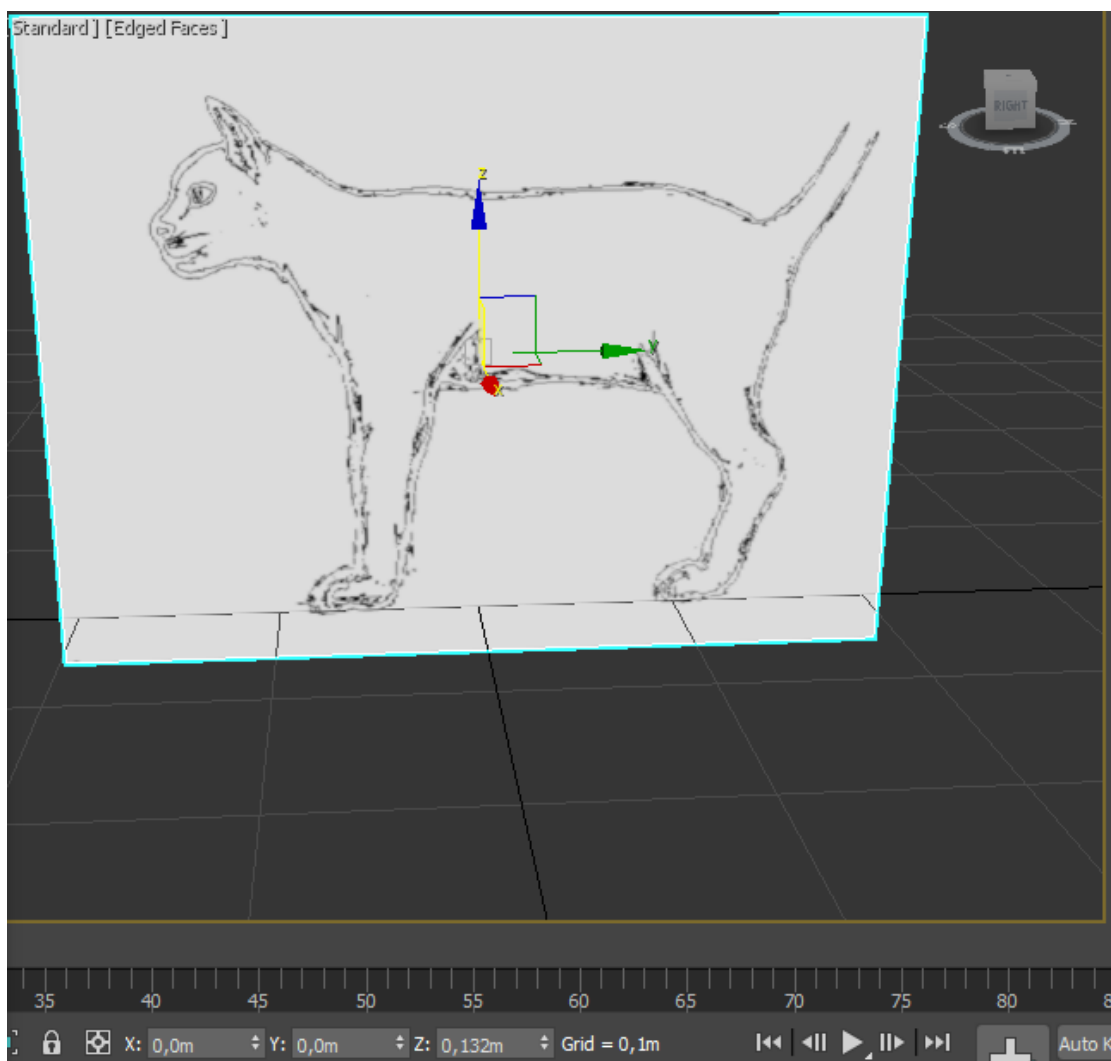


Kuva 8. Compact Material Editor -ikkuna

Materiaalipaikan valitsemisen jälkeen painetaan Blinn Basic Parameters -valikon Diffuse-kohdan oikealla puolella olevaa neliönmuotoista painiketta. Painikkeesta avautuu Material Browser -ikkuna, josta valitaan Bitmap-valinta. Valinnan jälkeen avautuu Select Bitmap Image File -ikkuna, jossa valitaan referenssikuva ja painetaan Open-painiketta.

Referenssikuvasta tehdyn materiaalin ollessa valittuna painetaan Material Editor -ikkunan Show Shaded Material In Viewport -painiketta.

Referenssikuva -materiaali sijoitetaan taso -objektin pinnalle Material Editor -ikkunan Assign Material to Selection -painikkeesta tai raahaamalla materiaali taso-objektin päälle (Kuva 9).



Kuva 9. Referenssikuvalla varustettu taso -objekti oikealla paikallaan

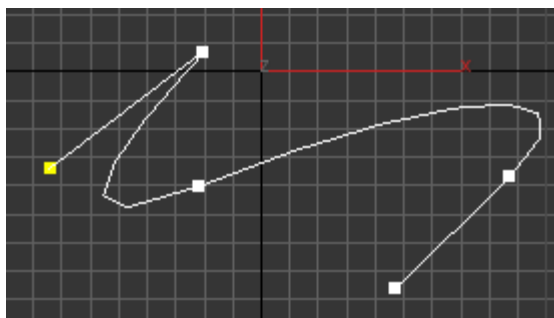
Materiaali raahataan Material Editor -ikkunan material-paikasta viewport-ikkunassa näkyvään taso-objektiin. Kuva tulee näkyviin taso-objektin pinnalle. Kuva sopii taso-objektiin täydellisesti, koska sen mittasuhteet ovat identtiset referenssikuvan resoluution kanssa. Referenssikuva voidaan nyt asetella viewport-ikkunassa haluttuun sijaintiin.

Referenssikuva sijoitetaan mallinnusalueen keskelle nollaamalla X- ja Y-akselien arvot translate-toiminnon ollessa päällä. Translate-toiminto mahdollistaa objektien liikuttamisen XYZ-akseleilla. Toiminnon voi aktivoida esimerkiksi W-pikanäppäimestä tai valikosta, johon pääsee painamalla hiiren oikeanpuoleista painiketta. Z-akselin arvo asetettiin niin, että referenssikuvassa kissan tassujen pohjat ovat samalla tasolla Z-akselin arvon 0 kanssa.

7.4 Mallintamisen aloittaminen

Mallintamisen aloittamiseen on useita vaihtoehtoja. Yleisesti mallintamisessa on suositeltavaa aloittaa yksinkertaisella muodolla, joka on lähimpänä haluttua lopputulosta. Tässä työssä on käytetty spliniä työn mallinnuksen aloittamiseen. Tällä tavoin on päästy esittelemään vaihtoehtoinen ratkaisu yleisimpien 3D-mallinnuksen aloittamiseen käytettyjen lähestymistapojen rinnalle.

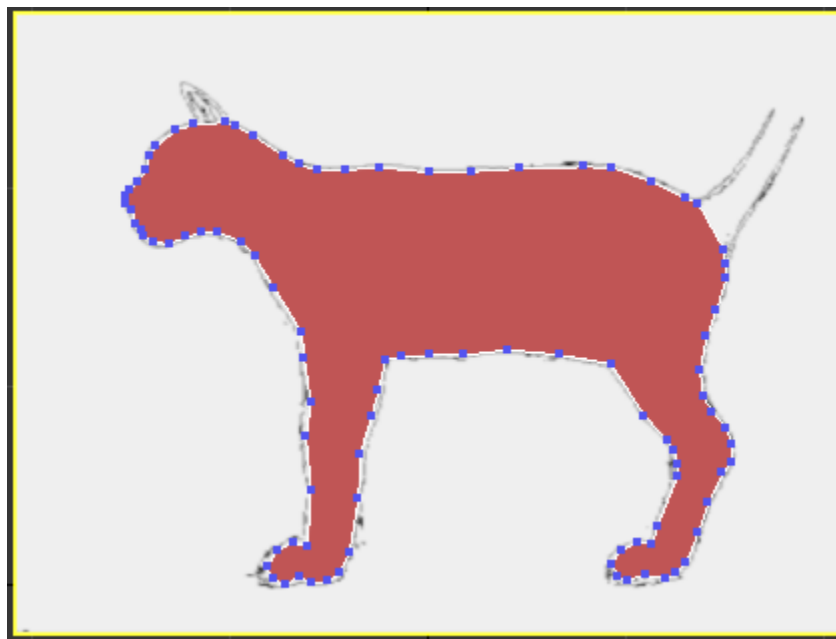
Splini on käyrä, jolla on vähintään kaksi kontrollipistettä, joiden mukaan se kulkee (Kuva 10). Uuden splinin saa luotua 3ds Maxin Command Panel -ikkunan Create-välilehden Shapes-valinnan alta löytyvistä muodoista valitsemalla haluamansa objektityypin. Splinille luotiin kontrollipisteitä seuraten referenssikuvan piirrettyjä kissan ääri viivoja. Splinin ensimmäinen ja viimeinen kontrollipiste yhdistettiin toisiinsa referenssikuvan ääri viivojen kiertämisen jälkeen.



Kuva 10. Esimerkki splini -käyrästä

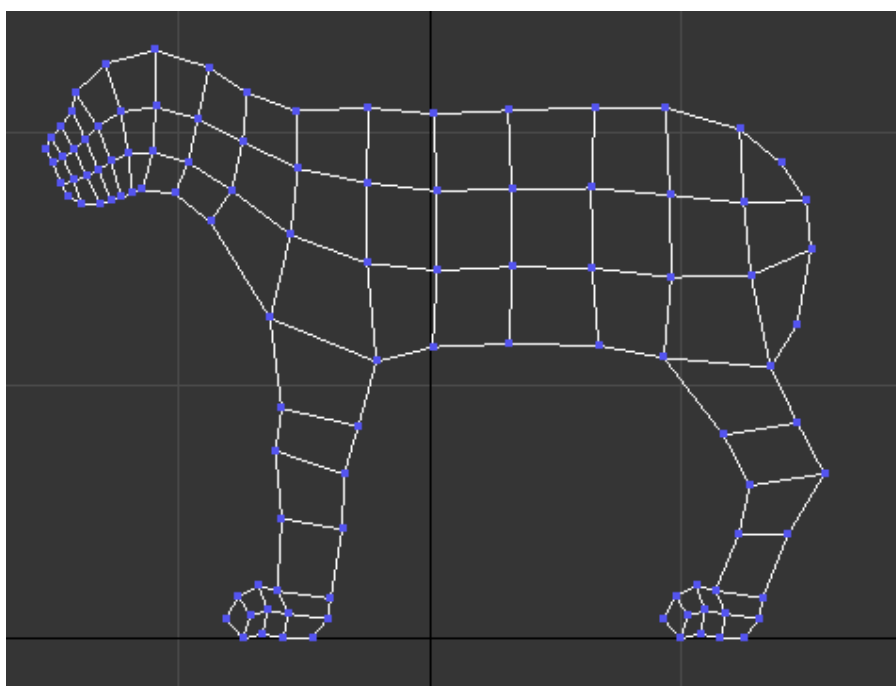
Tämän jälkeen splini muokattiin Editable Poly -objektiksi, painamalla hiiren oikealla painikkeella ja painamalla Convert To -valinnan alta Convert to Editable Poly -painiketta.

Muokattava objekti on nyt taso, jonka ääri viivat muodostavat vertekspisteet seuraavat referenssikuvan ääri viivoja (Kuva 11). Osa mallin vertekspisteistä on ylimääräisiä ja hankaloittamassa mallille hyvän topologian saavuttamiseksi edellyttäviä toimenpiteitä.



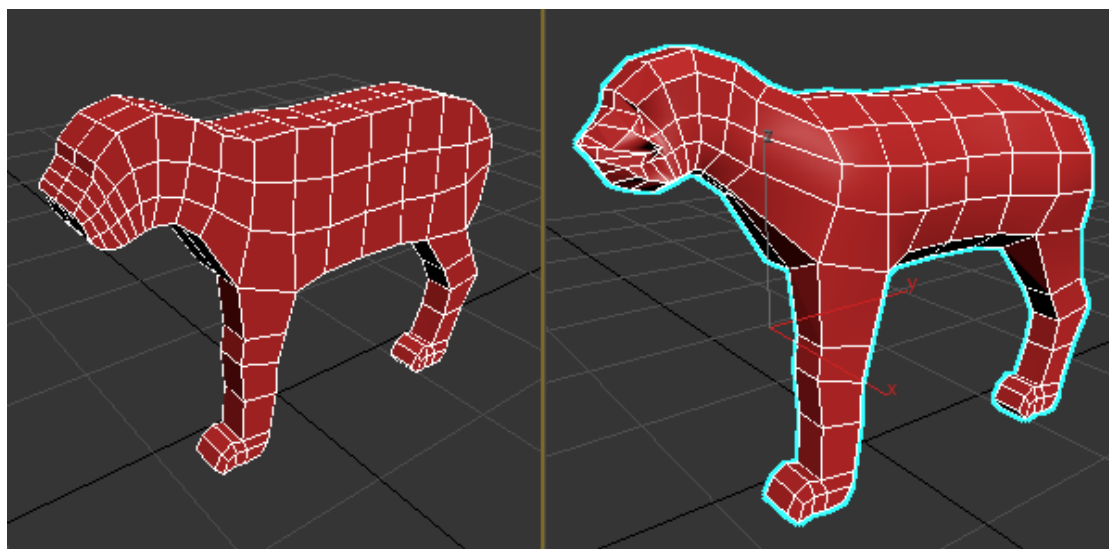
Kuva 11. Lukuisista vertekspisteistä muodostettu taso

Mallista voi poistaa vertekspisteitä valitsemalla vertekspisteen ja painamalla askelpalautinta niin, että jäljelle jäävät vertekspisteet ovat mahdollisimman tasaisin välimatkoin toisiinsa nähden. Tämän jälkeen yhdistetään vertekspisteet valitsemalla kaksi vastakkaista vertekspistettä ja painamalla connect-painiketta 3ds Maxin Command Panel -valikosta. Lopuksi valitaan vielä kaikki keskivertoa suuremmat vertekspisteiden välit muodostavat särmät Edge-valinnalla ja painetaan connect-painiketta kuvassa (Kuva 12) näkyvän lopputuloksen saavuttamiseksi.



Kuva 12. Kaksiulotteisen mallin verkkorakenne

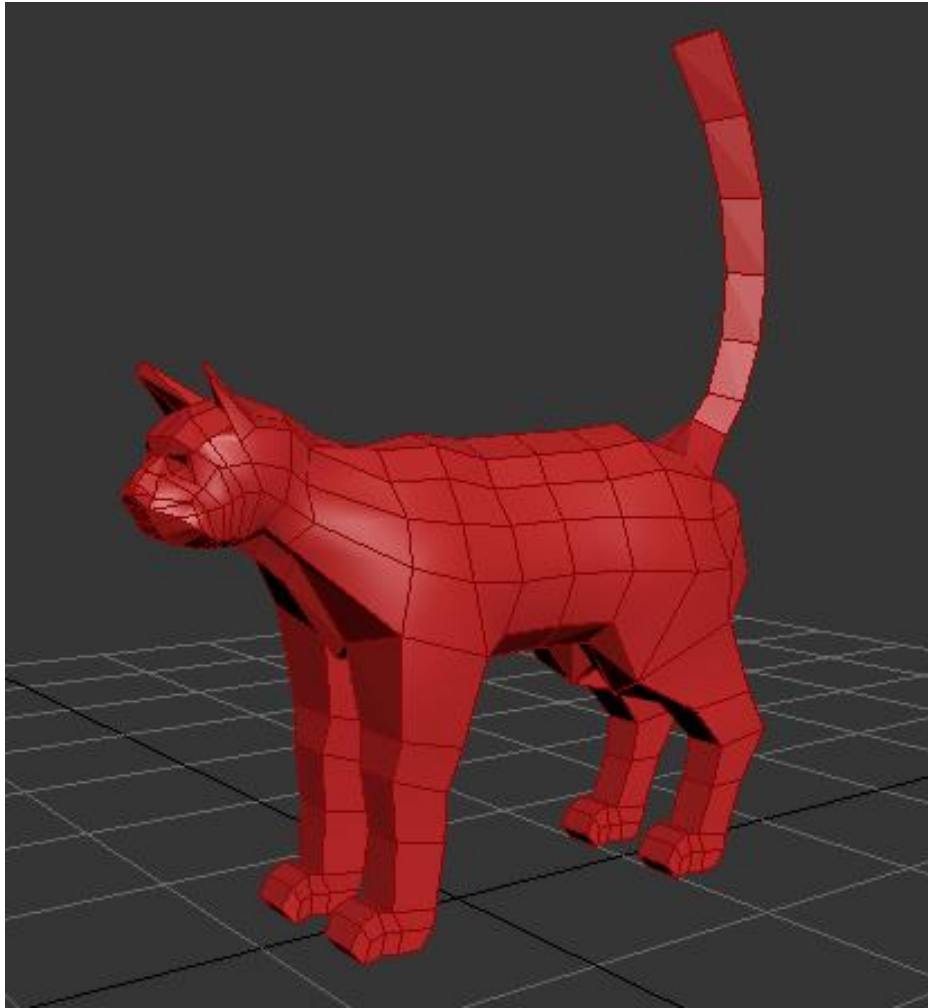
Tasomaisen objektin muuttamisen kolmiulotteiseksi monikulmioverkoksi voi toteuttaa Extrude Modifier -työkalua käyttäen. Valitaan Command Panel -valikosta tahko -valintatila, jonka jälkeen Ctrl + A -näppäinyhdistelmällä valitaan kaikki tahkot mallista ja painetaan Extrude-painiketta, jotta mallista saadaan kolmiulotteinen lisäämällä mallille tahkoja X-akselin suunnassa (Kuva 13).



Kuva 13. Extrude Modifier -työkalun käyttämisen vaikutukset

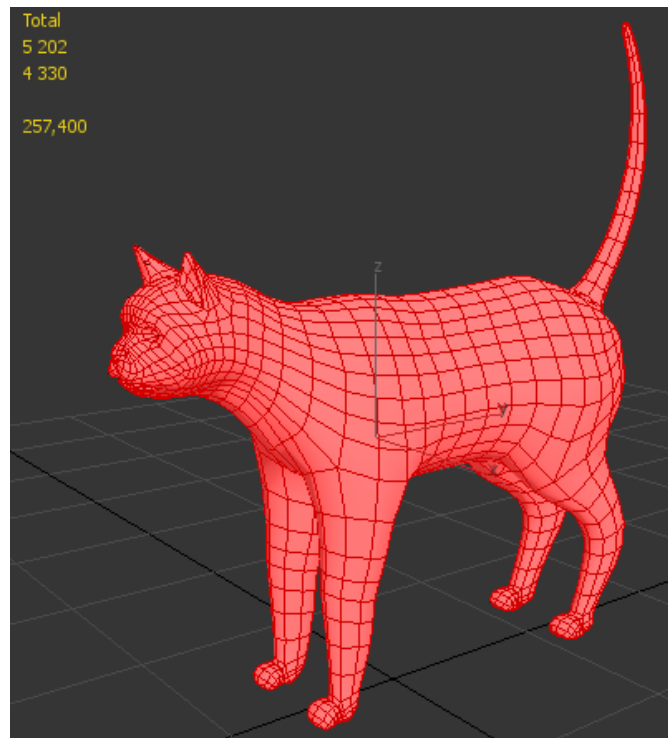
3ds Max -mallinnusohjelmassa on useita monikulmioverkon peilaamiseen ja monistamiseen tarkoitettuja modifier-työkaluja. Jos halutaan, että mallista tulee täysin symmetrinen, voidaan käyttää esimerkiksi Symmetry Modifier -työkalua. Peilaamalla 3D-mallille valmiiksi mallinnetusta puolikkaasta mallista toisenkin puolen saadaan kokonainen, symmetrinen 3D-malli. Peilaamalla voi säästää huomattavia määriä aikaa mallinnettaessa symmetrisiä monikulmioverkkoja.

Monikulmioverkkoon lisätään peilaamisen jälkeen korvat ja häntä Extrude Modifier -työkalulla (Kuva 14).



Kuva 14. Monikulmioverkko peilattuna sekä häntä ja korvat lisättyinä

Monikulmioverkon kulmikkuutta vähentämällä saadaan 3D-mallista orgaanisemman näköinen. Kulmikkuuden vähentämiseen käytetään TurboSmooth Modifier -työkalua (Kuva 15).



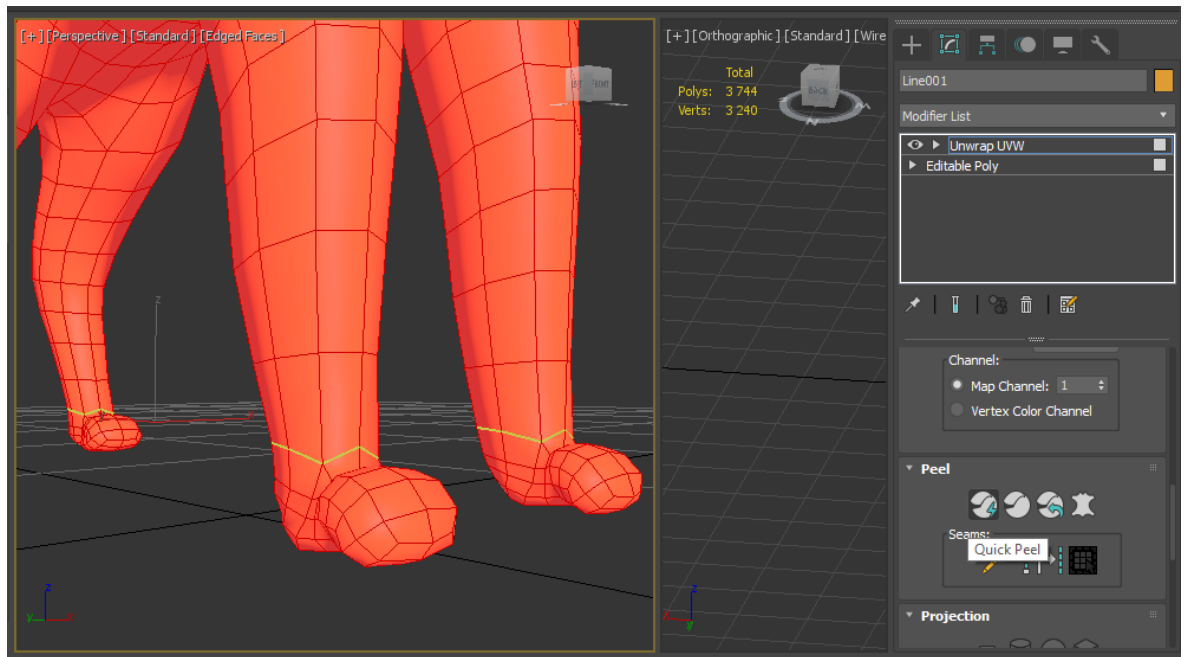
Kuva 15. TurboSmooth Modifier -työkalun siloittama monikulmioverkko

7.5 Teksturointi

Kolmiulotteisen mallin teksturoimiseen käytetään yleisimmin kaksiulotteista tekstuuria, joka kiedotaan mallin ympärille. Tämä toteutetaan UV-kartalla. UV-karttaan sijoitetaan UV-kartoitettavan monikulmioverkon verteksipisteet. 3D-avaruudessa sijaitseviin verteksipisteisiin tallennetaan niiden sijainti kaksiulotteisella UV-kartalla. Tämä mahdollistaa kaksiulotteisen diffuusiokartan näyttämisen kolmiulotteisen monikulmioverkon pinnalla.

UV-kartta saadaan luotua Unwrap UVW Modifier -työkalulla. Työkalusta avataan UV Editor -ikkuna painamalla Unwrap UVW Modifier -työkalun Edit UVs -valikosta Open UV Editor -painiketta. Monikulmioverkon levittämiseen UV-kartalle on useita tapoja. Tässä työssä monikulmioverkon osat toisistaan erottavat saumat UV-kartassa luodaan siihen tarkoitetun saumoitus -työkalun sijaan

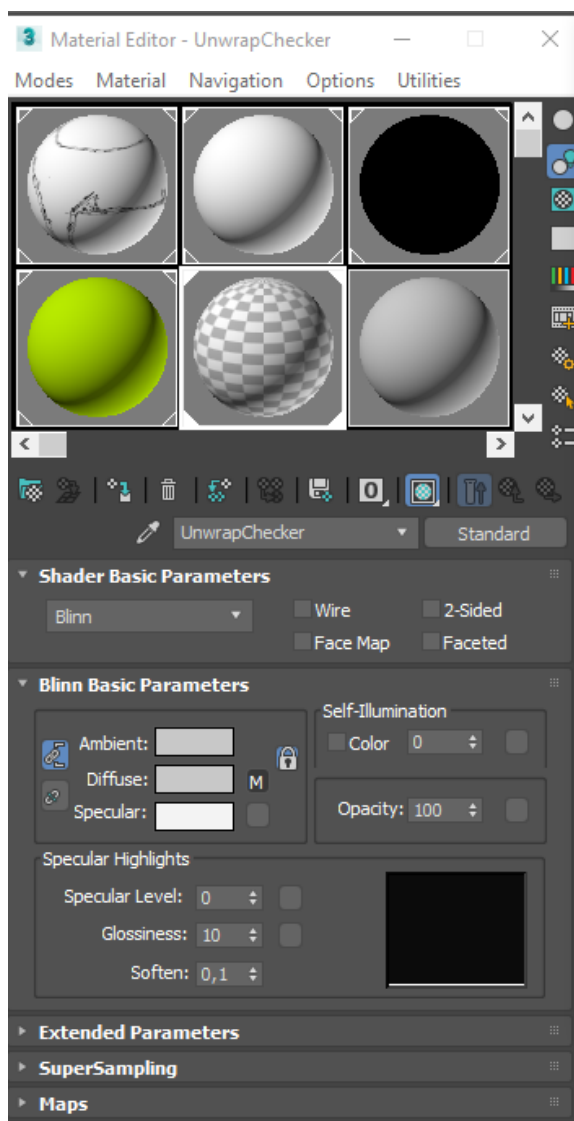
Unwrap UVW Modifier -työkalun särmä -valinnalla ja Quick Peel -toiminnolla (Kuva 16).



Kuva 16. UV-karttaan saumojen luomista Quick Peel -toiminnolla

Saumojen luomisen jälkeen valitaan saumojen erottamat monikulmioverkon alueet yksi kerrallaan Unwrap UVW Modifier -työkalun tahkovalintatilassa ja painetaan Flatten by Polygon Angle -painiketta UV Editor -ikkunan Explode-valikosta. Tämän jälkeen Saumojen erottamat monikulmioverkon alueet asetellaan UV Editor -ikkunaan niin, että niiden päälle tekstuurien piirtäminen on mahdollisimman helppoa.

UV-kartoituksen sopivuutta monikulmioverkon pinnalla voidaan tarkastella CheckerPattern -materiaalin avulla. Avataan Compact Material Editor -ikkuna 3ds Maxin Rendering-välilehdestä (Kuva 17).

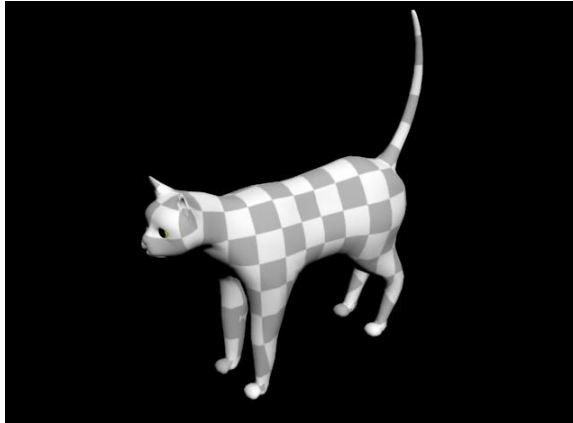


Kuva 17. CheckerPattern -materiaali Compact Material Editor -ikkunassa

Ikkunassa valitaan materiaalipaikka, johon CheckerPattern -materiaali sijoitetaan. Materiaali -paikan valitsemisen jälkeen painetaan Blinn Basic Parameters -valikon Diffuse-kohdan oikealla puolella olevaa neliönmuotoista painiketta. Painikkeesta avautuu Material Browser -ikkuna, josta valitaan CheckerPattern kyseiseen materiaalipaikkaan.

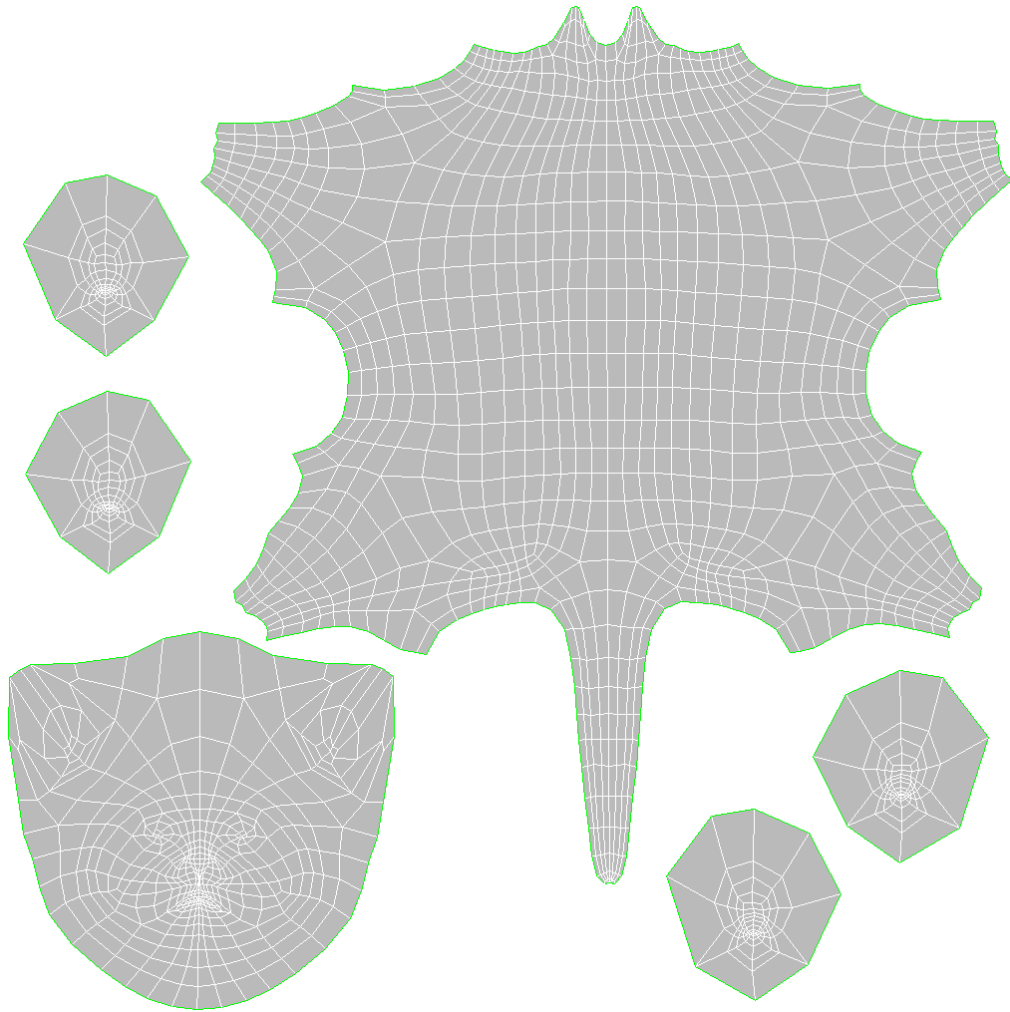
Luodun CheckerPattern -materiaalin ollessa valittuna painetaan Material Editor -ikkunan Show Shaded Material In Viewport -painiketta.

CheckerPattern -materiaali sijoitetaan monikulmioverkon pinnalle Material Editor -ikkunan Assign Material to Selection -painikkeesta tai raahaamalla materiaali monikulmioverkon päälle (Kuva 18).



Kuva 18. CheckerPattern -materiaali monikulmioverkon pinnalla

Luotu UV-kartoitus renderöidään rasterikuvaksi, jotta se on avattavissa kuvankäsittelyohjelmassa. UV-kartoituksen renderöintivalikon avaaminen tapahtuu UV Editor -ikkunan Tools -välilehden Render UVW Template -painikkeesta. UV-kartan renderöimiseksi painetaan Render UVs -ikkunan Render UV Template -painiketta. Renderöity kuva UV-kartasta tallennetaan tai kopioidaan leikepöydälle ikkunassa olevista painikkeista (Kuva 19).



Kuva 19. Renderöity kuva UV-kartasta

Renderöity kuva UV-kartasta avataan kuvankäsittelyohjelmassa, jossa sitä käytetään pohjana monikulmioverkon pinnalla näytettävän diffuusiokartan piirtämiseen.



Kuva 20. Valmis diffuusiokartta

Diffuusiokartan luomiseen käytettiin valokuvaa kissasta ja Paint.NET -kuvankäsittelyohjelman Clone Stamp -työkalua. Kuvassa ylhäällä (Kuva 20) on valmis diffuusiokartta, joka tallennetaan png -muodossa.

Diffuusiokartasta luodaan materiaali 3ds Maxin Material Editor -ikkunassa, josta se sijoitetaan monikulmioverkon pinnalle (Kuva 21).



Kuva 21. Valmis tekstuurikartta mallinnetun monikulmioverkon pinnalla

7.6 Animoidut mallit

Animaatioilla tuodaan liikkumattomiin 3D-malleihin eloa. Animoiduissa malleissa on animoimattomien 3D-mallien tavoin geometria ja tekstuuri. Animaatio toteutetaan muuttamalla 3D-mallin geometriaa liikuttamalla monikulmioverkon verteksipisteitä. Koska yksittäisten verteksipisteiden animoiminen on erittäin työlästä, käytetään usein apuna luuanimaatioita.

Luananimaatioilla tarkoitetaan monikulmioverkon verteksien liikuttamista jonkin objektin mukana ennalta määritettyjen parametrien mukaisesti. Näitä objekteja kutsutaan yleisesti 3D-mallinnuksesta puhuttaessa luuobjekteiksi.

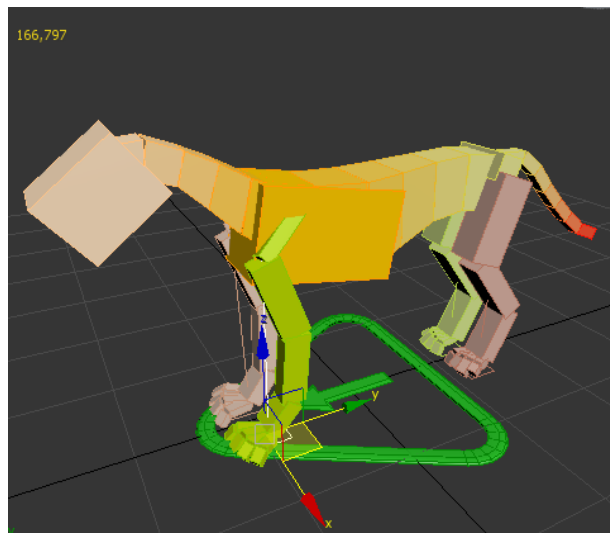
7.7 Luuobjektit

Luuobjekteilla on tukipiste, jolla on sijainti, asento ja skaala sekä mahdollisesti yksi isäntäobjekti, joka vaikuttaa näihin parametreihin hierarkkisesti. Animaatioiden kannalta luut ovat ainoastaan tukipisteitä 3D-avaruudessa, joiden mukaan verteksipisteitä liikutetaan, vaikka luut usein esitetään selkärankaisten

luiden kaltaisina luina. Luuobjektiksi kelpaa 3ds Maxissa mikä tahansa objekti, jolla on tukipiste. Luuobjektit ovat toimintansa puolesta lähempänä niveliä kuin luita.

Useamman luuobjektin hierarkiaa kutsutaan usein luurangoksi (engl. skeleton), vaikka kyseessä ei ole 3D-mallin tukiranka vaan kokoelma tukipisteitä, jotka liikkuvat isäntätukipisteen mukana.

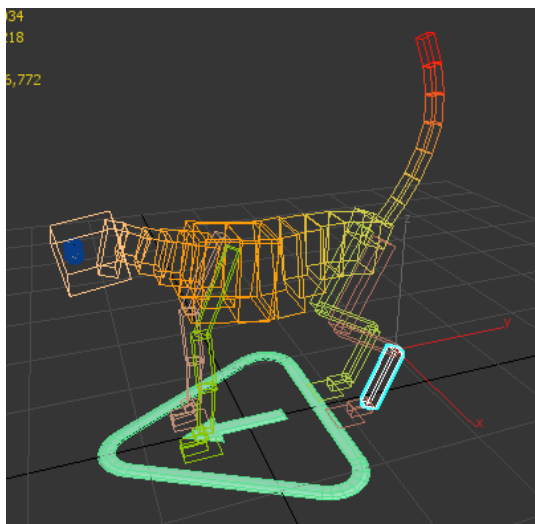
Tässä opinnäytetyössä käytetään luurangon tekemiseen ja animoimiseen 3ds Maxin CAT-liitännäistä (Character Animation Toolkit). CAT -liitännäisessä on paljon valmiita luurankoja, joista valitaan eniten mallinnetun monikulmioverkon rakennetta muistuttava esiasetus (Kuva 22).



Kuva 22. CAT -liitännäisen valmis luuranko

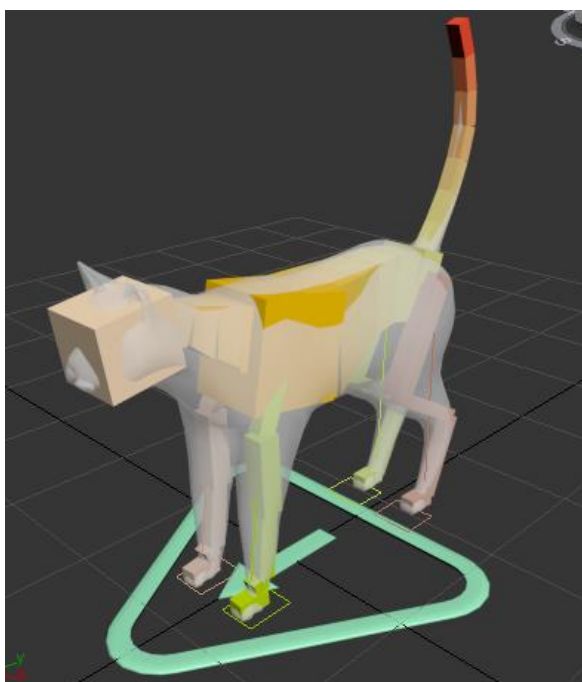
CAT -luuranko luodaan valitsemalla Command Panel -valikon Create -välilehdeltä Helpers -valinnan alta pudotusvalikosta CAT Objects -valinta, jonka jälkeen valitaan CATParent. CATParent luodaan Viewport -ikkunaan, jonka jälkeen Command Panel -valikon CATRig Load Save -valikosta ladataan Panther-esiasetus.

Luotu CAT -luuranko muokataan mallinnettuun monikulmioverkkoon sopivaksi muuttamalla ensin CATUnits Ratio -lukuarvoa Command Panel -valikon Modify-välilehdeltä niin, että luuranko on samankokoinen mallinnetun monikulmioverkon kanssa (Kuva 23). CAT -luurangon jokainen tukipiste käydään yksi kerrallaan läpi muuttaen samalla niiden sijainteja tarvittaessa.



Kuva 23. CAT -luurangosta muokattu luuranko

CAT -luurangon ollessa samassa asennossa mallinnetun monikulmioverkon kanssa (Kuva 24), siirrytään monikulmioverkon liikuttamisen CAT -luurangon mukaan mahdollistavaan Skinning-prosessiin.

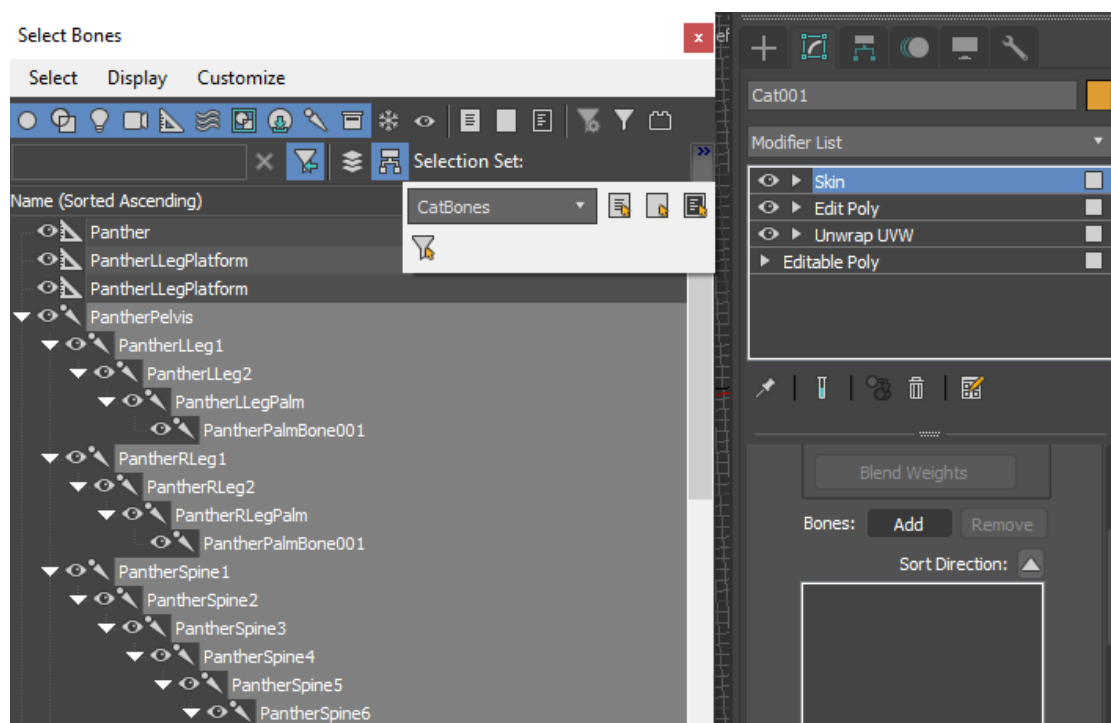


Kuva 24. CAT -luuranko sovitettuna monikulmioverkkoon

7.8 Skinning-prosessi

Skinning-prosessin tarkoitus on sitoa monikulmioverkon vertekspisteet seuraamaan animoitavaa luurankoa, splinejä tai muita objekteja. Prosessi mahdollistaa luuanimaatioiden käyttämisen monikulmioverkon geometrian muokkaamiseen animaatioissa. Prosessin toteuttamiseen 3ds Maxissa voidaan käyttää Skin Modifier -työkalua. Skin Modifier -työkalu mahdollistaa luuobjektien asennon vaikutuksen määrän määrittämisen monikulmioverkon geometriaan verteksikohtaisesti. Tätä kutsutaan vertekspisteiden painottamiseksi.

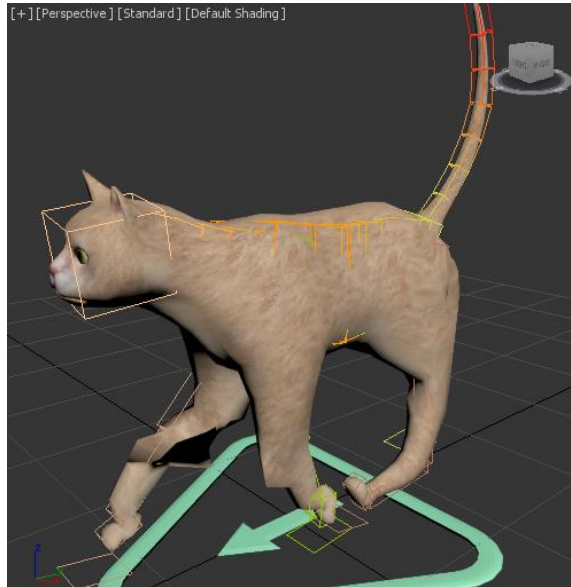
Prosessissa lisätään Skin Modifier -työkalu monikulmioverkon modifier-piinoon. Skin Modifier -työkaluun lisätään luuobjektit, joiden halutaan vaikuttavan monikulmioverkon geometriaan. Luiden lisääminen tapahtuu Skin Modifier -työkalun luuobjektien lisäyspainikkeesta. Lisäyspainikkeesta avautuu luuobjektien valintaikkuna, josta valitaan kaikki luuobjektit (Kuva 25).



Kuva 25. Luiden lisääminen Skin Modifier -työkaluun

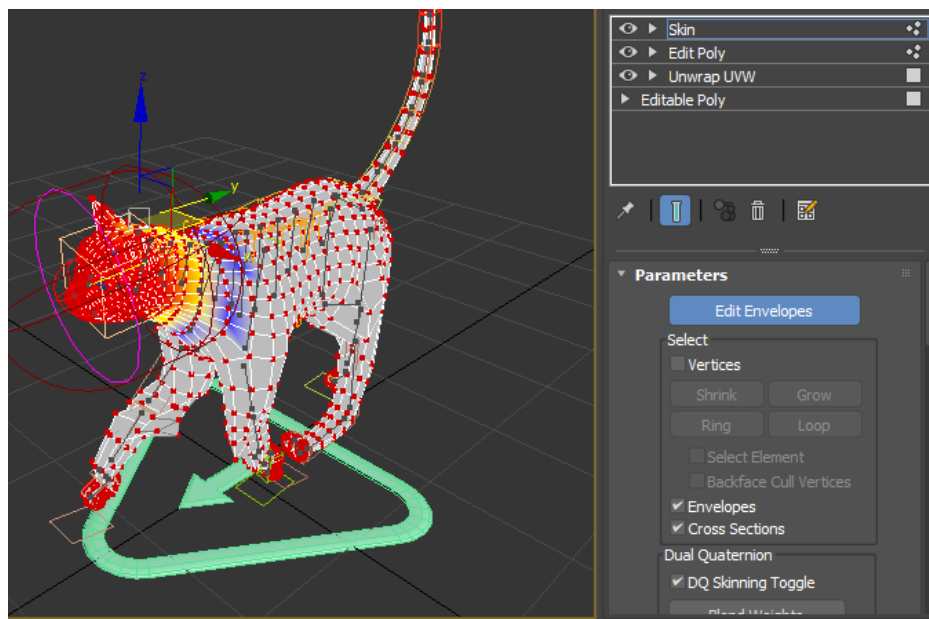
Luiden lisäämisen jälkeen vertekspisteillä on automaattiset oletuspainoarvot, joita säätämällä pyritään toteuttamaan monikulmioverkon mukautuminen luurankon animaatioihin halutulla tavalla.

Verteksipisteiden painoarvojen vaikutuksen näkemiseksi Viewport-ikkunassa luodaan avainkehys, jossa luubjektien asentoja muutetaan. Tässä olisi hyvä käyttää luubjekteilla asentoa, joka vastaisi kissan raajoille luonnollista ääri-asentoa. Luubjektien asentojen muutoksen vaikutukset näkyvät verteksipisteissä (Kuva 26).



Kuva 26. Oudosti käyttäytyvät verteksipisteet

Verteksipisteiden painoarvoja säädetään Skin Modifier -työkalusta aktivoimalla Edit Envelopes -tila (Kuva 27). Tilan aktivoimisen jälkeen valitaan Skin Modifier -työkalun luubjektien listasta luubjekti, jonka vaikutusta verteksipisteisiin halutaan muokata. Luubjektikohtaisia verteksipisteiden painoarvoja muokataan Skin Modifier -työkalusta avattavasta Weight Tool -ikkunasta.



Kuva 27. Skin Modifier -työkalun Edit Envelopes -tila aktiivisena

Weight Tool -ikkunaa käytetään vertekspisteiden painoarvojen asettamiseen. Vertekspisteiden painoarvojen muokkaamiseksi valitaan Viewport-ikkunasta muokattavat vertekspisteet. Valituille vertekspisteille asetetaan haluttu painoarvo Weight Tool -ikkunan painikkeista. Asetettu painoarvo voi olla mitä vain nollan ja yhden välillä kahden desimaalin tarkkuudella. Arvolla 0 luuobjektilla ei ole vaikutusta kyseiseen vertekspisteeseen. Arvolla 1 vertekspiste seuraa luuobjektin vaikutusta täysin. Luuobjektien yhteenlaskettu painoarvo vertekspistettä kohtaan on aina 1, joten useammalla luuobjektilla ei voi olla sataprosenttista vaikutusta yhtä vertekspistettä kohtaan (Autodesk 2016).

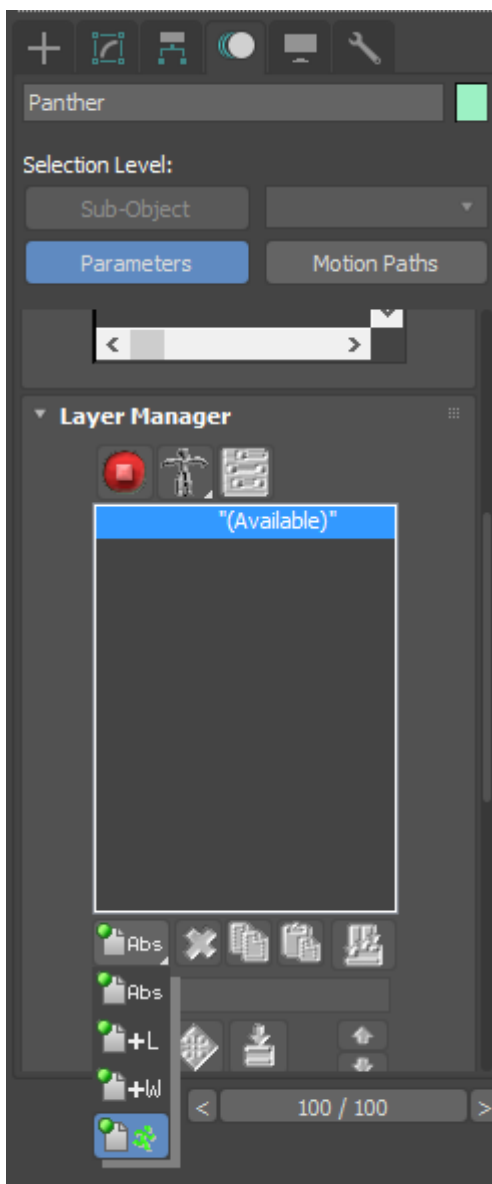
Vertekspisteiden luuobjektikohtaisia painoarvoja muutetaan niin, että 3D-mallin raajat taipuisivat mahdollisimman luonnollisen näköisesti animaatioissa.

7.9 Animointi

Videopeleissä pelihahmojen liikkeiden animoiminen toteutetaan yleisimmin luuanimaatiotekniikalla. Luuanimaatiotekniikassa tallennetaan luiden asennot animaatioiden aikajanalle avainkehysinä (engl. keyframe). Avainkehysinä ollessa aikajanalla useita, luiden asennon muutokset interpoloidaan avainkehysten välisellä alueella olemassa olevista avainkehysistä. Automaattinen avainkehysten välinen interpolaatio mahdollistaa sulavien animaatioiden toteuttamisen ilman, että jokainen avainkehysten välinen luiden asento täytyy lisätä aikajanalle käsin.

Tässä opinnäytetyössä tehdään mallinnetulle monikulmioverkolle kävelysyklin animaatio. Animaatio toteutetaan CAT -liitännäisen CATMotion -järjestelmän avulla. CATMotion -järjestelmässä on luotujen animaatioiden hallitsemiseen käytettävä Layer Manager -valintataulu. Layer Manager -valintataulu on CAT-Layer -järjestelmän käyttöliittymä, jossa käsitellään animaatioita animaatiotasoina.

Layer Manager -valintataulu (Kuva 28) avataan CAT -liitännäisen Motion-välilehdeltä parameters-painikkeen alta.



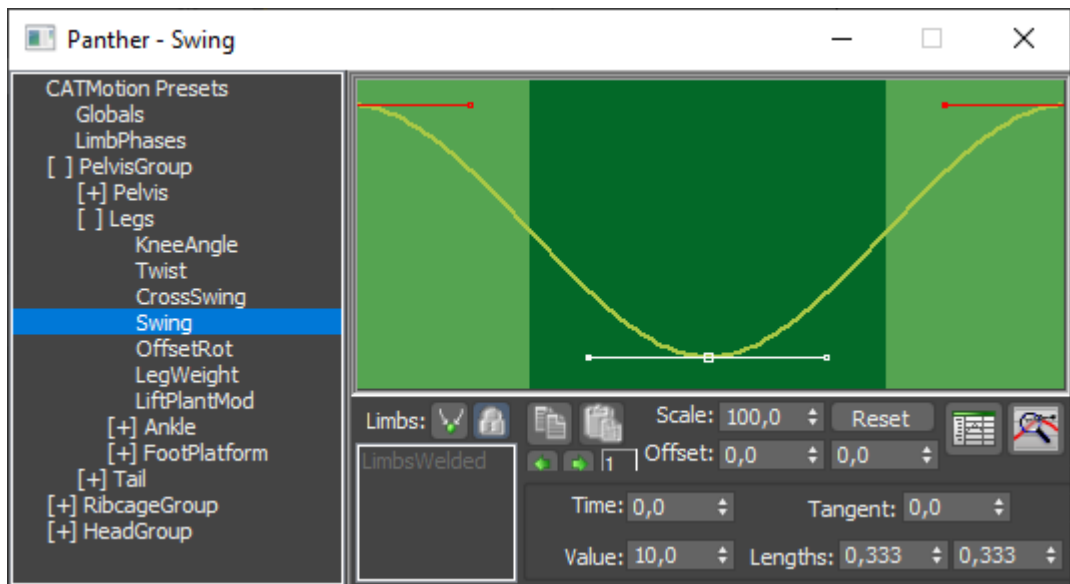
Kuva 28. Layer Manager -valintataulu

Layer Manager -valintauluun lisättävät tasot pinotaan järjestyksessä ylhäältä alas. Lisättäviä animaatiotasoja ovat Absolute, Relative Local, Relative World ja CATMotion -tasot. Absolute-tasoon aikajanelle asetetut avainkehykset peittävät aina edellisen pinossa olevan animaatiotason. Relative Local -taso lisää avainkehykset aiempaan pinossa olevaan animaatiotasoon. Relative World -taso lisää avainkehykset niin, että aiemmilla animaatiotasolla ei ole vaikutusta Relative World -tason vaikutukseen. CATMotion -tasoa käytetään kävelysykli-animaatioiden tekemiseen.

Layer Manager -valintatauluun lisätään CATMotion -animaatiotaso painamalla valintataulun alaosassa sijaitsevasta painikkeesta, josta avautuvasta pudotusvalikosta valitaan juoksevaa ihmishahmoa esittävä valinta.

Stop -painikkeen näköisestä painikkeesta käynnistyy animaatiotila, jonka jälkeen luut asettuvat valitun animaatiotilan määrittämiin asentoihin. Luodussa animaatiotilassa on oletuksena kävelysyklarimaation esiasetus.

Valitun animaatiotilan animaatiota hallitaan Layer Manager -valintataulussa sijaitsevien animaatiotasojen painoarvoja hallitsemalla ja CATMotion Editor -ikkunassa parametreja muuttamalla (Kuva 29), joka hallitsee aikajanelle sijoitettuja avainkehysksiä sekä tarvittaessa lisää niitä.



Kuva 29. CATMotion Editor -ikkuna, jossa näkyy säädettäviä parametreja

Kävelysyklarimaatioiden lisäksi CATMotion Editor -ikkunassa pystyy luomaan muitakin animaatiocyklejä, kuten esimerkiksi juoksu- ja paikallaanoloanimaatioita (Kuva 30).



Kuva 30. Kuvakaappaus valmiin 3D-mallin kävelyanimaatiosta

CATMotion -järjestelmässä luotuja animaatioita pystyy käyttämään fbx -tiedostoformaattia tukevissa ohjelmissa.

7.10 Pelimoottoriin vieminen

Monikulmioverkko ja luuranko viedään yhdessä luurankomallina fbx -tiedostoformaattiin. Vieminen tapahtuu valitsemalla Viewport -ikkunassa monikulmioverkko ja kaikki luuobjektit. Tämän jälkeen painetaan 3ds Maxin File -painikkeesta, viedään osoitin Export -valinnan kohdalle ja valitaan Export Selected -valinta.

Select File to Export -ikkunassa annetaan tallennettavalle fbx-tiedostolle nimi ja haluttu tiedostopolku ja painetaan Save-painiketta. Save-painikkeen painamisen jälkeen avautuu FBX Export -ikkuna, jossa määritetään tiedostoon vietävät kohteet ja ominaisuudet. FBX Export -ikkunasta tarkistetaan, että lisäasetuksien alla Units-asetusten Scene units converted to: -kohdan pudotusvalikko on asetettu senttimetreiksi. Tämän jälkeen painetaan OK -painiketta.

CAT-luurangolle tehdyt animaatiot viedään fbx-tiedostoformaattiin erikseen. Vieminen tapahtuu valitsemalla Viewport-ikkunassa kaikki luuobjektit. Tämän

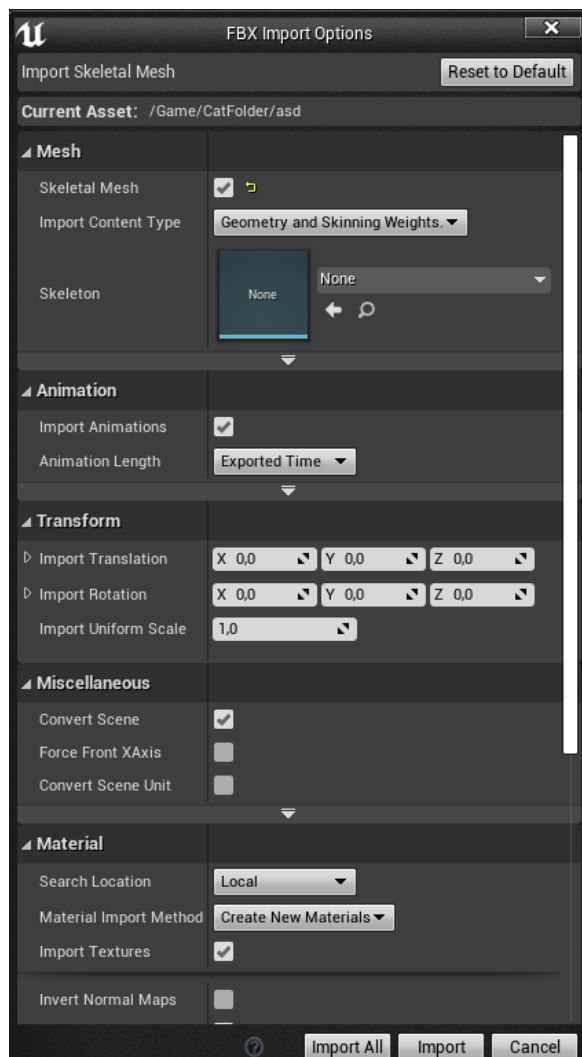
jälkeen painetaan 3ds Maxin File-painikkeesta, viedään osoitin Export-valinnan kohdalle ja valitaan Export Selected -valinta.

Select File to Export -ikkunassa annetaan tallennettavalle fbx-tiedostolle nimi ja haluttu tiedostopolku, jonka jälkeen painetaan Save-painiketta. Save-painikkeen painamisen jälkeen avautuu FBX Export -ikkuna, jossa täytetään Animation-asetusten alla oleva Animation-valintaruutu sekä Bake Animation -asetusten alta Bake Animation -valintaruutu. Bake Animation -valintaruudun oikealla puolella oleviin numerokenttiin asetetaan vietävän animaation ensimmäisen ja viimeisen avainkehysten sijainti aikajanalla. Tämän jälkeen painetaan OK-painiketta.

7.11 Mallin tuominen pelimoottoriin

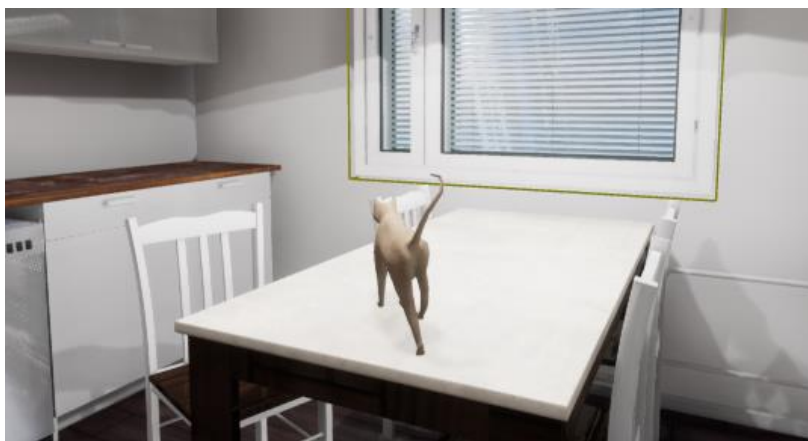
3D-mallien tuominen pelimoottoreihin on suosituimmista pelimoottoreissa kuten Unityssä ja Unrealissä samankaltaisia: luurankomalli ja tarvittavat tekstuurit kopioidaan pelin projektikansion tiedostopolkuun tiedostoselaimella tai raahaamalla tiedostot pelimoottorin ikkunaan. Pelimoottori tunnistaa tuomisen automaattisesti ja tiedostot lisätään avoimena olevan peliprojektin tietokantaan. Tässä opinnäytetyössä kuvataan tarkemmin prosessi, jolla malli tuodaan Unreal Engine -pelimoottoriin.

Pelimoottoriin tuodaan ensin luurankomalli. Luurankomallin fbx -tiedosto raahataan Unreal Enginen Content Browser -tilan sisälle tiedostopolusta, johon se on tallennettu. Tämän jälkeen avautuu FBX Import Options -ikkuna, josta täytetään Skeletal Mesh -valintaruutu. FBX Import Options -ikkunasta asetetaan Transform -asetusten Import Uniform Scale arvoon yksi. Tämän jälkeen painetaan Import-painiketta (Kuva 31).



Kuva 31. Unreal Engine -pelimoottorin FBX Import Options -valikko

Animaatioiden tuomisen yhteydessä valitaan FBX Import Options-valikon mesh-osion skeleton-kohtaan aiemmin tuotu luurankomalli. Luurankomallin valitsemisen jälkeen täytetään Import Animations-valintaruutu ja painetaan Import -painiketta. Tämän jälkeen luurankomalli ja animaatiot on viety 3ds Max -mallinnusohjelmasta pelimoottoriin onnistuneesti (Kuva 32).



Kuva 32. Animoitu 3D-malli pelimoottorissa

8 POHDINTA

Autodeskin 3ds Max oli työn toteuttamisen kannalta hyvä valinta. 3ds Maxin kattava dokumentaatio mahdollisti työssä käytettyjen, hahmomallinnuksessa hieman kokeellisten lähestymistapojen soveltamisen ilman aiempaa kokemusta kyseisistä työkaluista.

Työn aikana käytettiin joitakin tunteja työn vaiheiden automatisointiin käytettävien ohjelmavaihtoehtojen tutkimiseen. Täysin automatisoituja ja ilmaisia ratkaisuja ei kuitenkaan 3ds Maxiin löydetty. Etsintöjen tuloksena päädyttiin kokeilemaan 3ds Maxin CAT-liitännäistä. CAT-liitännäisen CATParent-objekti osoittautui tehokkaaksi apuvälineeksi pelihahmon luurakenteen luomisprosessissa. CATMotion-animaatioeditorilla realistisen näköisen kävelysklianimaation toteuttaminen ja muokkaaminen onnistui lähes vaivatta.

Mallin animoimisen kannalta olisi ollut järkevintä mallintaa häntä suoraan taaksepäin ojennettuna. Suoran hännän saa helposti taivutettua kaarelle, mutta kaarevan hännän oikaiseminen suoraksi niin, että mallin pinta pysyy täysin virheettömänä, on lähes mahdotonta toteuttaa pelkkiä luuobjekteja manipuloidalla.

Teksturoimisen suunnitteluun olisi kannattanut käyttää enemmän aikaa laadun parantamiseksi tai ylimääräisen työn ennalta ehkäisemiseksi. Työssä tekstuurikartat aseteltiin ja leikattiin niin, että yksi UV-kartan leikkauksista sijaitsi mal-

linnetun kotikissan niskassa näkyvällä paikalla. Sauman sijainti niskassa aiheutti tekstuuriin virheellisen näköisen kohdan, joka olisi voitu välttää tekemällä leikkaus huomaamattomampaan kohtaan.

Pelihahmon tekstuurikarttojen tekemiseen käytetty Paint.NET -kuvankäsittelyohjelma ajoi asiansa ilman ongelmia. Tekstuurikartat olisi voinut maalata myös 3ds Maxin sisällä käyttäen mallinnusohjelman Viewport Canvas -työkaluja. Viewport Canvas -työkaluvalikoima on kuitenkin suppeampi kuin Paint.NET -kuvankäsittelyohjelmassa. Viewport Canvasin etuna Paint.NET -kuvankäsittelyohjelmaan verrattuna on mahdollisuus maalata tekstuurit suoraan mallinnetun monikulmioverkon pinnalle, josta maalatut tekstuurit tallentuvat valittuun kuvatiedostoon.

Suurin ongelma työn aikana oli CAT -luurangon peilattujen luuobjektien nimien identtisyys alkuperäisten luuobjektien kanssa. Ongelmatilanteelta olisi voitu välttyä asettelemalla mallin luuobjektit molemmille puolille käsin. Ongelma kuitenkin ratkaistiin nimeämällä peilatut luuobjektit uudelleen luurangon symmetrisyyden takaamiseksi. Luuobjektien nimien kaksoiskappaleista olisi koitunut ongelmia todennäköisesti viimeistään luurangon pelimoottoriin viemisen yhteydessä.

Työtä voitaisiin jatkokehittää luomalla mallinnetulle kissalle turkki käyttämällä Nvidian HairWorks -työkaluja tai Unreal Enginen Hair Rendering and Simulation -ominaisuuksia. Jatkokehitysidea olisi myös verhonnut alleen UV-kartan suunnittelemattomuudesta mallille aiheutuneen tekstuurivirheen kissan niskassa.

LÄHTEET

All3DP. 2020a. 2020 Best 3D Modeling Software/ 3D Design Software.

WWW-dokumentti. Saatavissa:

<https://all3dp.com/1/best-free-3d-modeling-software-3d-cad-3d-design-software/> [Viitattu 29.4.2020].

All3DP. 2020b. 2020 Best 3D Animation Software (Some are Free). WWW-dokumentti. Saatavissa:

<https://all3dp.com/1/best-20-3d-animation-software/> [Viitattu 21.3.2020].

Android developers. 2018. Improve your game's performance. WWW-dokumentti. Saatavissa:

<https://developer.android.com/games/optimize> [Viitattu 18.3.2020].

Andys arcade. 2018. Battlezone (Atari 1980). WWW-dokumentti. Saatavissa:

<http://www.andysarcade.de/battlezone.html> [Viitattu 14.4.2020].

Autodesk. 2013. MAXScript Help: How to output geometry data to text file - part one. WWW-dokumentti. Saatavissa:

<http://docs.autodesk.com/3DSMAX/15/ENU/MAXScript-Help/index.html?url=files/GUID-38562330-1435-4112-9265-DE965319A2F2.htm,topicNumber=d30e108846> [Viitattu 7.2.2020].

Autodesk. 2016. Weight Tool Dialog (Skin Modifier). WWW-dokumentti. Saatavissa:

<https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-C5EB44E2-5A3E-44DF-8B64-A0E00B8EDC71-htm.html> [Viitattu 7.2.2020].

Bethke, E. 2003. Game Development and Production. Plano: Wordware Publishing, Inc.

Blackman, S. 2013. Beginning 3D Game Development with Unity 4: All-in-One, Multi-Platform Game Development, Second Edition. New York: Apress.

Christensen, S. 2015. Defining your target audience: Who are you serving? WWW-dokumentti. Saatavissa: <https://99designs.com/blog/business/define-your-target-audience/> [Viitattu 2.5.2020].

Computer Hope. 2017. Z-buffering. WWW-dokumentti. Saatavissa: <https://www.computerhope.com/jargon/z/zbuffering.htm> [Viitattu 6.5.2020].

Eck, D. 2010. Fundamentals of Computer Graphics: With Java, OpenGL, and JOGL. Geneva: Hobart and William Smith Colleges.

Ellis, C. 2020. 16 essential tools for graphic designers in 2020. WWW-dokumentti. Saatavissa: <https://www.creativebloq.com/graphic-design/tools-every-graphic-designer-should-have-6133208> [Viitattu 3.4.2020].

Ford, I. 2019. The five purposes of digital graphics. WWW-dokumentti. Saatavissa: <http://www.bluesquarething.co.uk/imedia/r082/82lo1files/dgpurpose.pdf> [Viitattu 26.2.2020].

Game Designing. 2020. The Top 10 Video Game Engines. WWW-dokumentti. Saatavissa: <https://www.gamedesigning.org/career/video-game-engines/> [Viitattu 4.6.2020].

Gamespot. 2001. QOTW: How Important Are Graphics to Games? WWW-dokumentti. Saatavissa: <https://www.gamespot.com/articles/qotw-how-important-are-graphics-to-games/1100-2693475/> [Viitattu 20.1.2020].

Lansdown, H. 2019. Digital Modelmaking: Laser Cutting, 3D Printing and Reverse Engineering. Ramsbury: The Crowood Press Ltd.

Lincoln Laboratory. 2018. Sage: Semi-automatic ground environment air defense system. WWW-dokumentti. Saatavissa:

<https://www.ll.mit.edu/about/history/sage-semi-automatic-ground-environment-air-defense-system> [Viitattu 20.5.2020].

Mobsby, N. 2019. What Is Substance Painter? – Simply Explained. WWW-dokumentti. Saatavissa:

<https://all3dp.com/2/what-is-substance-painter-simply-explained/> [Viitattu 18.2.2020].

Murdock, K. 2004. 3ds Max 6 Bible. Hoboken: Wiley Publishing, Inc.

Okun, J & Zwerman, S. 2010. The VES handbook of visual effects: industry standard VFX practices and procedures. Burlington: Elsevier Inc.

Oracle. 2020. B.3 Summary of Image File Formats and Image Compression Formats. WWW-dokumentti. Saatavissa:

https://docs.oracle.com/cd/B28359_01/appdev.111/b28414/ap_imgfmts.htm#BABFJDDI [Viitattu 1.5.2020].

Overmars, M. 2012. A Brief History of Computer Games. WWW-dokumentti. Saatavissa:

https://www.stichtingspel.org/sites/default/files/history_of_games.pdf [Viitattu 8.5.2020].

Polycount. 2018. Normal map. WWW-dokumentti. Saatavissa:

http://wiki.polycount.com/wiki/Normal_map [Viitattu 19.5.2020].

Puhakka, A. 2008. 3D-grafiikka. Talentum. Helsinki: Alma Talent

Taimisto. s.a. Lyhyt taidehistorian oppimäärä. WWW-dokumentti. Saatavissa:

<https://peda.net/valkeakoski/opetuspalvelut/pk/tyry/oppiaineet/kuvataide/8-kuvataide-uusi3/teoriaa/lto2:file/download/8f5078142cd836a3362e4d6569e447ce44ad66c9/LY-HYT%20TAIDEHISTORIAN%20OPPIM%C3%84%C3%84R%C3%84.pdf> [Viitattu 27.5.2020].

Takala, T & Savioja, L. 2007. Tietokonegrafiikka. WWW-dokumentti. Saatavissa:

<http://www.cse.hut.fi/fi/opinnot/T-110.1100/2007/Luennot/11.Tietokonegrafiikka.pdf> [Viitattu 1.4.2020].

TechTerms. 2009. Graphics. Saatavissa: <http://techterms.com/definition/graphics/> [Viitattu 3.2.2020].

Thorn, A. 2017. Mastering Unity 2017 Game Development with C#: Second Edition. Birmingham: Packt Publishing Ltd.

Turbosquid. 2017. Quad-Based Topology. WWW-dokumentti. Saatavissa: <https://www.turbosquid.com/3d-modeling/training/modeling/quad-based-topology/> [Viitattu 29.5.2020].

UFO 3D. 2019. History of 3D modeling: from euclid to 3D printing. WWW-dokumentti. Saatavissa: <https://ufo3d.com/history-of-3d-modeling> [Viitattu 3.5.2020].

Vector conversion. s.a. Raster vs Vector. WWW-dokumentti. Saatavissa: https://vector-conversions.com/vectorizing/raster_vs_vector.html [Viitattu 3.5.2020].

Workerbee (nimimerkki). 2018. WWW-julkaisu. Saatavissa: <https://99designs.com/blog/freelancing/optimize-workflow-designers/> [Viitattu 25.3.2020].

Yrittäjät. 2019. Suomen peliala työllistää ensi kertaa yli 3000 – tekijöitä tarvitaan ulkomailta. WWW-dokumentti. Saatavissa: <https://www.yrittajat.fi/uutiset/606691-suomen-peliala-tyollistaa-ensi-kertaa-yli-3000-tekijoita-tarvitaan-ulkomailta> [Viitattu 19.5.2020].

KUVALUETTELO

Kuva 1. Vektorigrafiikkaa ja rasterigrafiikkaa vierekkäin niiden erojen havainnollistamiseksi.....	12
Kuva 2. Kuution kylki, jossa on monikulmioverkon vertekspisteet, särmät ja tahkot havainnollistettuna	14
Kuva 3. Luonnos kissasta.....	23
Kuva 4. Väritetty konseptikuva	24
Kuva 5. Valmis referenssikuva	24
Kuva 6. Unit Setup -ja Display Unit Setup -ikkunat avoinna	25
Kuva 7. Referenssikuvan kokoinen taso -objekti	26
Kuva 8. Compact Material Editor -ikkuna	27
Kuva 9. Referenssikuvalla varustettu taso -objekti oikealla paikallaan.....	28
Kuva 10. Esimerkki splini -käyrästä.....	29
Kuva 11. Lukuisista vertekspisteistä muodostettu taso	30
Kuva 12. Kaksiulotteisen mallin verkkorakenne	30
Kuva 13. Extrude Modifier -työkalun käyttämisen vaikutukset.....	31
Kuva 14. Monikulmioverkko peilattuna sekä häntä ja korvat lisättyinä	32
Kuva 15. TurboSmooth Modifier -työkalun siloittama monikulmioverkko.....	33
Kuva 16. UV-karttaan saumojen luomista Quick Peel -toiminnolla.....	34
Kuva 17. CheckerPattern -materiaali Compact Material Editor -ikkunassa	35
Kuva 18. CheckerPattern -materiaali monikulmioverkon pinnalla.....	36
Kuva 19. Renderöity kuva UV-kartasta	37
Kuva 20. Valmis diffuusiokartta	38
Kuva 21. Valmis tekstuurikartta mallinnetun monikulmioverkon pinnalla.....	39
Kuva 22. CAT -liitännäisen valmis luuranko	40
Kuva 23. CAT -luurangosta muokattu luuranko.....	41
Kuva 24. CAT -luuranko sovitettuna monikulmioverkkoon	41
Kuva 25. Luiden lisääminen Skin Modifier -työkaluun	42
Kuva 26. Oudosti käyttäytyvät vertekspisteet	43
Kuva 27. Skin Modifier -työkalun Edit Envelopes -tila aktiivisena.....	43
Kuva 28. Layer Manager -valintataulu.....	45
Kuva 29. CATMotion Editor -ikkuna, jossa näkyy säädettäviä parametreja ...	46
Kuva 30. Kuvakaappaus valmiin 3D-mallin kävelyanimaatiosta.....	47
Kuva 31. Unreal Engine -pelimoottorin FBX Import Options -valikko	49
Kuva 32. Animoitu 3D-malli pelimoottorissa	50