

Raskaasta CAD-mallista kevyeksi peliasetiksi



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeen ammattikorkeakoulu, Tietojenkäsittelyn koulutusohjelma

Hämeenlinna, kevät 2019

Raisa Pernu

Tietojenkäsittelyn koulutusohjelma
Visamäki, Hämeenlinna

Tekijä	Raisa Pernu	Vuosi 2019
Työn nimi	Raskaasta CAD-mallista kevyeksi peliassetiksi	
Työn ohjaaja/t	Lauri Salminen	

TIIVISTELMÄ

Tämän opinnäytetyön aiheena on tutkia ja löytää sujuvin tapa tehdä kevyt low-poly peliasetti raskaasta high-poly 3D-mallista, joka on alun perin CAD-malli. Työn tilaajana on HAMK Smart tutkimusyksikkö. Opinnäytetyön tarkoitus ja syy on teollisuusalojen yritysten kiinnostus tehdä omista CAD-malleista peliasetteja opetus- ja markkinointitarkoitusta varten.

Opinnäytetyön tekijän täytyy yrittää löytää joitakin apuohjelmia koulutus-tarkoitukseen ja tämä tarkoittaa, että ohjelman lisenssin on oltava joko ilmainen tai ilmainen ainakin opiskelijoille. Yksi näistä on 3D-mallinnusohjelma Blender. Se toimii päätyökaluna tämän opinnäytetyön tekemisessä. Kaikki 3D-mallinnus tapahtuu Blenderissä sisältäen low-poly- ja cage meshien valmistuksen. Toiset ohjelmat keskittyivät pääosin Normal-mappien tekemiseen high-poly meshistä low-poly meshiin. Opinnäytetyön tekijän täytyy vertailla ja tutkia, mikä ohjelmista antaa parhaan tuloksen Normal-mappia tehdessä. Opinnäytetyössä käydään sivuavasti läpi muutamia apuohjelmia, joilla CAD-mallin pystyy viemään Blenderiin.

Tuloksissa käy ilmi, että kaikkein paras jälki ja sujuvin työtapa Normal-mappien tekemisessä tulee ainoastaan opiskelijoille ilmaisella Autodeskin 3DS Max- ohjelmalla. Täysin ilmaisista ohjelmista Blender ja xNormal yltyvät melkein yhtä hyvin tuloksiin. Opinnäytetyön tekijä ei ole aikaisemmin tehnyt mitään aiheeseen liittyvää 3D-mallinnusharrastuksestaan huolimatta. Tästä johtuen testitulokset voivat olla täysin erilaiset kokeneemman mallintajan suorittamana.

Avainsanat 3D-mallinnus, Normal-mapit, Unreal Engine 4, teollisuus

Sivut 28 sivua, joista liitteitä 3 sivua

Degree Programme in Business Information Technology
Visamäki, Hämeenlinna

Author	Raisa Pernu	Year 2019
Subject	From a heavy CAD-model to a light game asset	
Supervisors	Lauri Salminen	

ABSTRACT

The topic of this thesis is to research and find the most fluent way to create a light low-poly game asset from heavy high-poly 3D-model that was originally a CAD-model. The commissioner for this thesis was HAMK Smart research unit. The purpose for this thesis is the interest in industrial companies to convert their CAD-models to game assets for educational or marketing use.

The author of this thesis sought to find a software for educational use and this means free of charge for students. These include a 3D-modeling software which is called Blender. It is also the main tool in the process of making this thesis because all the 3D-modeling including the creation of low-poly mesh and cage are carried out in Blender. The additional software are mainly for baking Normal-maps from the high-poly mesh to the low-poly mesh and the author has to compare which software gives the best results. There is also some information about free software for converting CAD-models into Blender.

The end result is that the best Normal-map baking and the fastest workflow is reached in 3DS Max that is free only for students. Of the totally free software only Blender and xNormal reach almost the same quality. The author of this thesis has never done anything related to this thesis despite her hobby in 3D-modeling. That is why results may vary depending on whether a more experienced hobbyist or a professional has been involved in the process.

Keywords 3D-modeling, Normal-maps, Unreal Engine 4, industry

Pages 28 pages including appendices 3 pages

SISÄLLYS

1	JOHDANTO.....	1
2	3D- GRAFIIKKA	2
2.1	3D- tietokonegrafiikan historiaa	3
2.2	CAD-mallinnus	4
2.3	Ero high-poly- ja low-poly assettien välillä (ja miksi tarve keventää).....	5
2.4	UV-map ja Normal-map	5
2.5	Häkki low poly-ja high-poly meshien mukana Normal-mapia tehdessä.....	6
3	PELIMOOTTORI	7
4	OHJELMISTOT	8
4.1	Blender 2.79	9
4.2	Unreal Engine 4	10
4.3	FreeCAD.....	11
4.4	xNormal	12
4.5	Handplane 3D.....	12
4.6	3DS Max.....	12
4.7	CAD-tiedoston vienti Blenderiin teoriassa	13
4.7.1	.prt	14
4.7.2	.x_t.....	14
4.7.3	.stp	14
4.7.4	.jt.....	14
4.7.5	.stl	14
4.7.6	.obj.....	15
5	CAD-MALLI BLENDERIIN JA PELIASSETIKSI	16
5.1	Alkuvaikeudet.....	16
5.2	Toimiva ja toistaiseksi ilmainen apuohjelma	17
5.3	High-poly mesh ja low-poly mesh	17
5.4	Normal-mapit eri ohjelmissa.....	18
6	LOPPUTULOS	23
7	YHTEENVETO	25
8	LÄHTEET.....	26

Liitteet

Liite 1 Kuvia Normal-mapeista

Sanasto

(peli)Assetti	3D-mesh, joka on viety pelimoottoriin ja siellä siihen lisätään muitakin ominaisuuksia kuten läpinäkyvä törmäysmesh, fysiikat ja mahdolliset muut toiminnot
Bake	Ottaa säteiden avulla talteen erilaista informaatiota 3D-mallin pinnanmuodoista 2D-kuvaan
CAD	Computer Aided Design eli tietokoneavusteinen suunnittelu erityisesti teollisuuden eri aloilla
Cage	Häkki eli low-poly- ja high-poly meshejä ympäröivä mesh jota käytetään kun tehdään high-poly meshistä pinnanmuodot Normal-mapiksi
Export	Viedä tiedosto ohjelmasta pois halutussa tiedostomuodossa
FPS	Tilanteesta riippuen Frame Per Second eli kuinka sulavasti video tai ohjelma toimii näytöllä tai TV:ssä. Tarkoittaa myös First Person Shooteria eli peliä, jossa pelaaja ei välttämättä näe pelihahmostaan kuin käden tai aseensa ja pyrkii ampumaan pelissä erilaisia kohteita
High-poly mesh	Mahdollisimman tarkaksi ja yksityiskohtaiseksi tehty 3D-mesh
Import	Tuoda ohjelmaan tiedosto halutussa tiedostomuodossa
LOD	Level of Distance eli kuinka tarkasti peliasetti näkyy ja paljonko siinä on polygoneja tietyllä etäisyydellä (polygoneja ja assetin tarkkuutta vähennetään tai lisätään sisään- ja ulospäin suurennettaessa)
Low-poly mesh	Pelkistetty ja mahdollisimman vähän polygoneja eli tahkoja sisältävä 3D-mesh
Mesh	3D-mallin pinta, joka koostuu vertekseistä, särmistä ja tahkoista

Normal-map	2D-kuva yksityiskohtaisen ja tarkan 3D-meshin pinnanmuotojen informaatiosta
Pelimoottori	Ohjelma, jossa voi rakentaa pelin tai sovelluksen eri aseteista, äänistä ja muista materiaaleista (Unity, Unreal Engine, Source Engine, Cryengine...)
Polygon	Tahko, joka koostuu vertekseistä ja särmistä ja useimmiten se on joko kolmion- tai neliönmuotoinen
Quad polygon	2D-neliö, jossa on neljä särmää ja neljä verteksiä
Render(öinti)	Tietokone työstää joko prosessorilla tai näytönohjaimella valmiin 3D-mallin tai scenen lopulliseksi teräväpiirteiseksi kuvaksi/videoksi kaikkine valoineen, materiaaliasetuksineen ja varjoineen
Shader	Valotuksen ja varjostuksen asetus
Tekstuuri	3D-meshin pinnan kuvitus tai väritys 2D-kuvana
Tri(anguloitu) polygon	2D-kolmio, jossa on kolme särmää ja kolme verteksiä
UV-map	2D-kaavakuva 3D-mallin polygonien informaatiosta, monesti leikattu saumoja pitkin auki

1 JOHDANTO

Viime aikoina eri teollisuuden aloilla on herännyt kiinnostus käyttää heidän omia CAD-malleja koulutus- ja markkinointitarkoituksessa. Tätä varten CAD-malli on muutettava kevyeksi 3D-peliassetiksi. Tämä on kuitenkin ollut hyvin kallista yrityksille, koska se vaatii paljon manuaalista työtä 3D-mallien parissa. Tässä opinnäytetyössä tutkitaan, miten tätä työskentelytapaa voi nopeuttaa mahdollisimman paljon vaikka manuaalista työtä joutuukin tekemään edelleen.

Tarkoituksena on tarkastella ja tutkia teollisuutta sekä yleishyödyllistä opetuskäyttöä varten, kuinka raskaista CAD-malleista saadaan kaikista nopeiten ja parhaiten muutettua kevyitä peliassetteja Unreal Engine 4:ään. Tarkoituksena on hyödyntää lähinnä ilmaisohjelmia tai vähintäänkin opetuskäyttöön opiskelijalisenssillä ilmaisia ohjelmia.

Opinnäytetyössä kokeillaan ja testataan eri tekniikoita ja ohjelmia, joilla ensinnäkin saa CAD-malleja muutettua niin, että objekti pysyy samannäköisenä ja tarkkana kuten CAD-ohjelmassakin, mutta polygonien määrä on kuitenkin karsittu reilusti. Tarkoituksena on luoda sujuvasti toimivia, mutta silti näyttäviä peliassetteja.

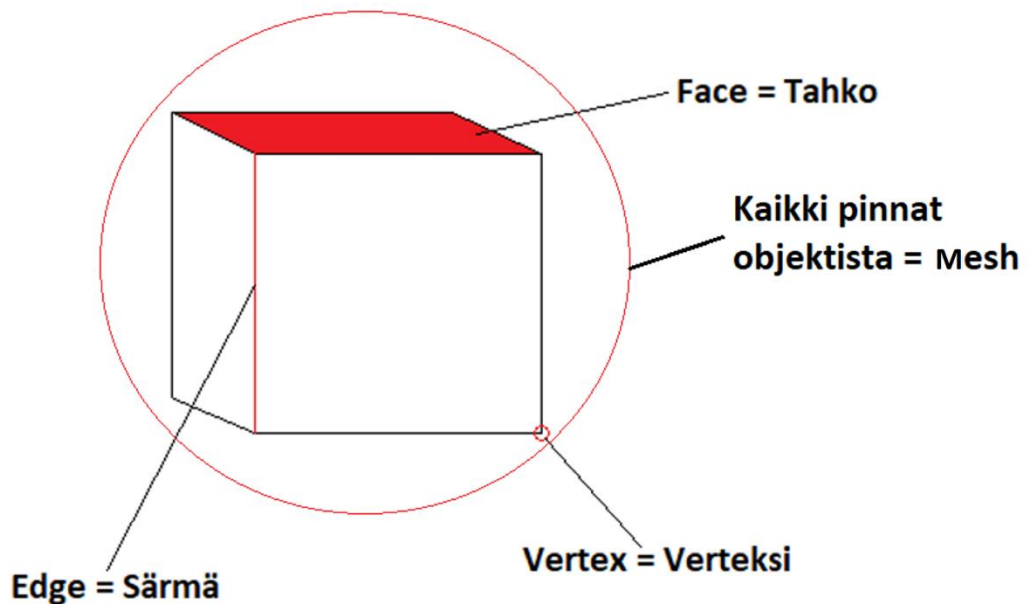
Teoriaosuudessa perehdytään pääosin 3D-mallintamisen maailmaan aloittaen sen peruskäsitteistä ja kerrotaan vähän historiaa. Sen lisäksi selvennetään eroja CAD-mallin ja peliassetin välillä. Esitellään myös tiedostomuodot, joita kokeillaan opinnäytetyön käytännönsuudessa. Kerrotaan itse käytännön työn prosessista, pelimootoreista yleisesti ja käytettävistä (apu)ohjelmista. Lopuksi käytännönsuudessa esitellään, miten itse työ raskaasta CAD-mallista kevyeksi low-poly peliassetiksi tapahtuu ja vertailaan saatuja tutkimustuloksia.

Opinnäytetyön tutkimuskysymykset ovat:

- Kuinka CAD-mallista voi tehdä manuaalisella työllä mahdollisimman sujuvasti kevyt 3D-peliassetti ilman kalliita maksullisia ohjelmia?
- Mitkä ilmaisohjelmat tai opiskelijalisenssillä ilmaiset ohjelmat auttavat parhaiten tässä ja varsinkin manuaalisen työn kanssa?

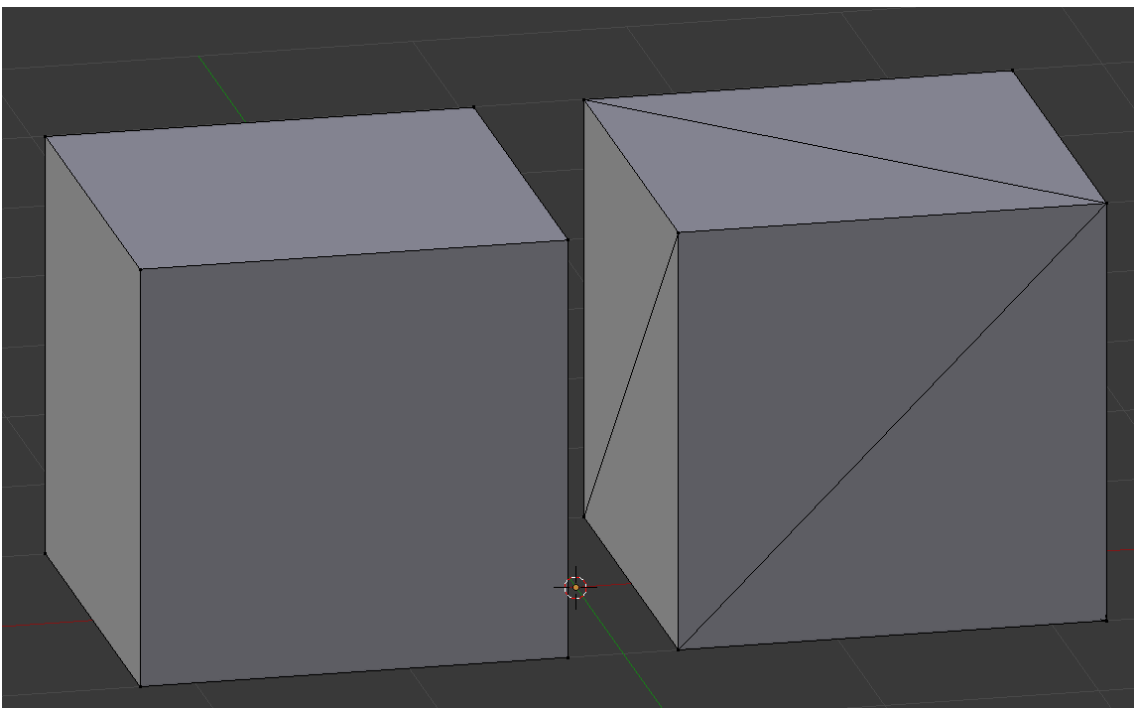
2 3D- GRAFIKKA

3D- objektit koostuvat monitahoisista malleista, joita kutsutaan polygoneiksi. Kaikki 3D-mallinnusohjelmat sisältävät joitakin perusmuotoja, joista mallintaja voi lähteä liikkeelle muovaamalla sitä. Esimerkiksi perusmuoto kuutiossa (kuva 1) on kuusi tahkoa (face), kahdeksan verteksiä (vertex) ja kaksitoista särmää (edge). Tahkoista käytetään myös nimitystä polygon. Yksi tahko koostuu vähintään kolmesta särmästä muodostaen kolmion. Yleisenä nyrkkisääntönä 3D-objekteja valmistaessa voidaan pitää, että yksikään tahko ei saisi olla neljää särmää eli quadia (kuva 2) suurempi. Yleisemmin tahkot ovat valmiissa työssä kolmen särmän kokoisia eli trianguloituja. 3D-mallin kuori tai pinta on nimeltään mesh. (Chopine 2011, s. 21-23)



Kuva 1. Yksinkertaisen 3D- mallin osat.

3D grafiikkaa tuottaessa oleellista on, että työt renderöidään lopuksi. Siinä tietokone kerää tiedot kaikista työlle asetetuista arvoista ja ominaisuuksista kuten tekstuureista, valaistuksesta, materiaaleista ja monista muistakin huomioon otettavista asioista. Renderöintiprosessissa voidaan käyttää erilaisia algoritmeja, jotka vaikuttavat lopputulokseen huomattavasti. Algoritmit muokkaavat värimaailmaa, valaistusta ja varjoja ja täten lopputulos voi olla hyvinkin monenlainen riippuen siitä, millä algoritmeilla haluaa työnsä renderöidä. Renderöinti voi viedä hyvinkin paljon aikaa riippuen tietokoneen prosessointitehoista, näytönohjaimesta ja lopputulokseen halutuista ominaisuuksista. Jotkin ohjelmat tarjoavat reaaliaikaista renderöintiä, mutta kuten edellä mainittu, se vaatii tietokoneelta enemmän tehoja koko renderöinnin ajaksi. (Mullen 2012, s. 133)



Kuva 2. Quadeista koostuva mesh (vas.) ja trianguloitu mesh (oik.).

2.1 3D- tietokonegrafiikan historiaa

Alussa 60- ja 70-luvulla tietokonegrafiikka oli vielä hyvin alkeellista. Graafinen näkymä oli mustavalkoista ja ensimmäiset tietokonepelit olivat erittäin yksinkertaisen näköisiä 2D-pelejä. Virallista ensimmäistä 2D-grafiikkaista peliä ei varmuudella tunneta, mutta 70-luvulla tunnetuksi tuli ainakin Steve Russellin ja Martin Graetzin ohjelmoima Spacewar!. Tuohon aikaan 3D-grafiikkakin oli mahdollista, mutta ainoastaan wireframe- muodossa eli rautalankamallina ja 3D-objektit olivat aina läpinäkyviä. (Chopine 2011, s. 4)

Sittemmin Yhdysvalloissa Utahin yliopistossa alettiin kehittää 3D-tietokonegrafiikkaa lento-opetuksen simulaatioiden vuoksi, sillä simulaatiolennot ovat halvempi ja turvallisempi metodi opettaa uusia pilotteja. Samaan aikaan tietokonegrafiikkaa alettiin kehittämään muihinkin tarpeisiin kuten CAD-mallintamiseen eli tietokoneavusteiseen suunnitteluun. (Chopine 2011, s. 5)

Tästä alkoi 3D-grafiikan kehityksen nousukausi. Useampi Utahin yliopiston opiskelija jatkokehitti toistensa saavutuksia tietokonegrafiikan ja sen algoritmien parissa. Monet tuona aikana kehitetyistä shadereista ja renderointitekniikoista ovat käytössä yhä tämän päivän 3D-mallinnusohjelmissa. (Chopine 2011, s. 6)

3D-mallintaminen oli pitkään vain yritysten ja varakkaiden yksityishenkilöiden etuoikeus, koska riittävillä tehoilla varustetut tietokoneet olivat niin kalliita pitkälle 90-luvun loppuun asti. Vasta 2000-luvulle tultaessa yksityishenkilöilläkin alkoi olla varaa riittävän tehokkaisiin tietokoneisiin. Sen lisäksi ilmainen 3D-mallinnusohjelma Blender julkaistiin vuonna 1998. (Chopine 2011, s. 11; Blender n.d.a)

Suora 3D-mallintaminen on yleisesti käytössä elokuva- ja peliteollisuudessa. Suorassa 3D-mallintamisessa ei välttämättä huomioida tarkkoja mittoja ja speksejä. Siinä muotoillaan ja muovataan ilman parametrisiä arvoja halutun näköinen objekti melko suurpiirteisesti tai taustalle asetettua 2D-mallipiirrosta vastaavaksi siirtelemällä verteksien sijainteja. (Chopine 2011, s. 22)

2.2 CAD-mallinnus

CAD-mallintaminen on yleisesti ottaen parametrinen mallintaminen. Lähinnä se soveltuu eri teollisuuden alojen suunnittelukäyttöön kun suunnitellaan tuotteita valmistettavaksi erilaisista materiaaleista. Siinä 3D-objekti eli tuote koostuu monesti useista elementeistä. Jokaisella osalla ja elementin palasella on tarkat mittansa ja yleensä myös muita speksejä (= parametrejä). Spekseihin kuuluvat materiaalitiedot ja määrät kuin myös mahdollinen hinnoittelu ja menekki materiaaleille. Nämä muodostavat monitasoisen ja erityyppistä informaatiota sisältävän hierarkisen kokonaisuuden eli kokonaisen lopputuotteen. Kun tuotteesta suunnitellaan vaikka uutta versiota, niin suunnittelijan ei tarvitse kuin muuttaa esimerkiksi vain yhtä ulkoista osaa ja kaikki elementit materiaali- ja hintatietoineen tämän yhden osan ympärillä muuttuvat automaattisesti vastaamaan tämän muunnellun osan reunoja ja seinämiä vastaaviksi. (Alba 2018; Brunelli 2017; PCMAG 2018)

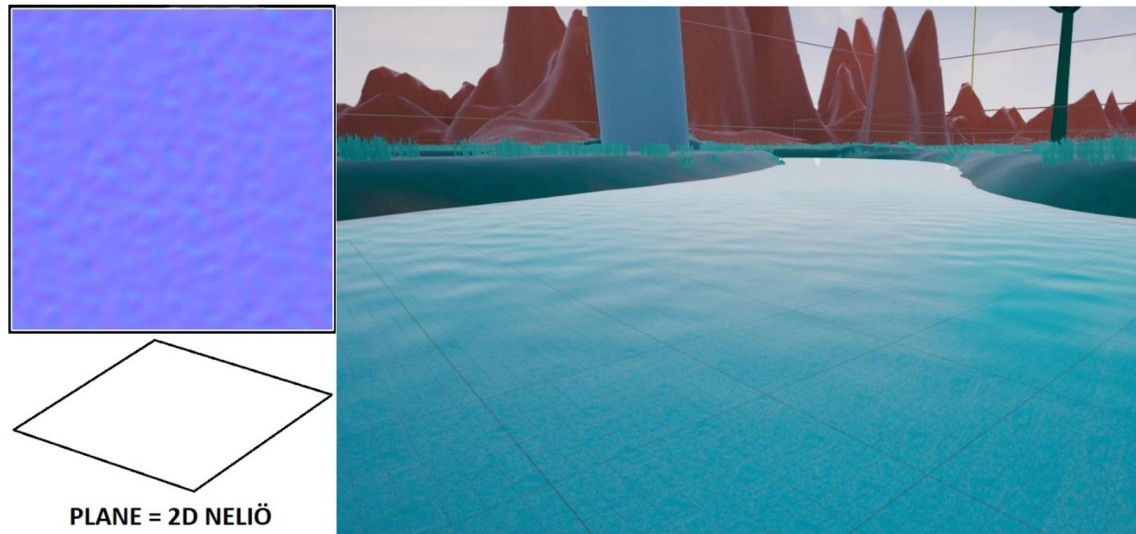
2.3 Ero high-poly- ja low-poly asettien välillä (ja miksi tarve keventää)

3D-peliasetteja mallinnettaessa on huomioitava polygonien lukumäärä. Mitä enemmän polygoneja, sen yksityiskohtaisempi ja kauniimpi lopputulos, mutta samalla asetista tulee raskaampi ja tietokoneella menee kauemmin aikaa laskea kaikki yksityiskohdat mallista. Varsinkin peliasetteja tehtäessä tämä on tärkeä asia huomioida, koska pelikentissä on paljon asetteja ja niiden olisi kuitenkin oltava esteettisen näköisiä. Peleissä yleensä asettien laatua kevennetään kun kamera zoomataan kauemmas näyttämään pelikenttää laajemmin, koska siinä tilanteessa asetit eivät tarvitse tarkkoja yksityiskohtia ja se säästää tietokoneeltakin tehoja. Peli jaksaa pyöriä sujuvammin. Kun zoomataan takaisin jotain pelikentän yksityiskohtaa, eli asetta kohti niin polygonien määrää lisätään, jotta asetti näyttäisi paremmalta ja yksityiskohtaisemmalta. Näitä kutsutaan nimellä LOD eli Level Of Detailiksi. Siltikin nämä asetit on kevennettävä erikseen, jotta lähtäisyydelläkin peli pyörisi sulavasti. (Chopine 2011, s. 33-34)

Jotta saadaan hienoja ja yksityiskohtaisia asetteja mahdollisimman vähäisellä prosessointiteholla, on asettien polygoneja vähennettävä menettämättä asettien yksityiskohtaisia pinnanmuotoja. Tämä onnistuu siten, että on tehtävä kaksi versiota. Ensimmäisenä on tehtävä raskas ja korkealaatuinen versio, jossa polygoneja saa tarpeen mukaan olla paljonkin. Tätä kutsutaan high-poly asetiksi. Tämä on se, jolta peliasetti tulee näyttämään lopuksi. Sen jälkeen tehdään vähemmän yksityiskohtainen ja pelkistetympi versio asetista eli niin sanottu low-poly -versio. Tämän kohdalla koetetaan pitää polygonit minimissä. (Mullen 2012, s. 126)

2.4 UV-map ja Normal-map

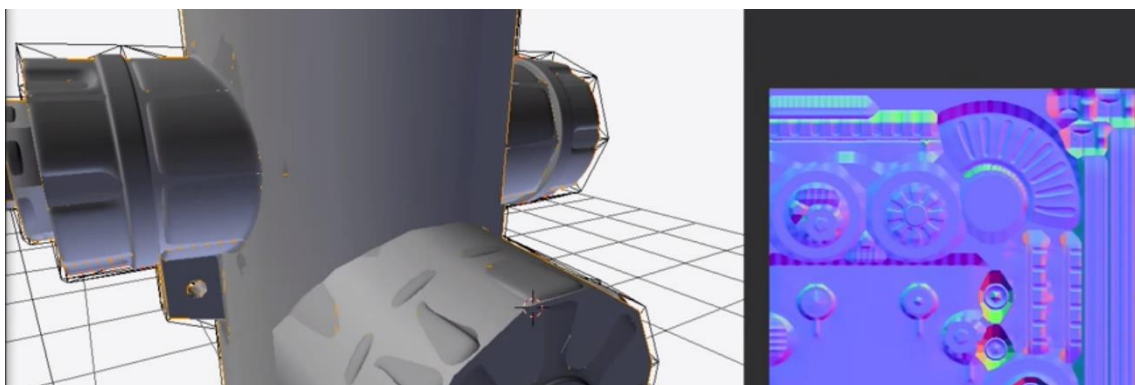
UV-map on 2D-versio, eli litteä kaava 3D-mallin polygoneista ja verteksi-pisteiden sijainneista. Peliasettien low-poly versiosta tehdään Blenderissä tai jossakin muussa ohjelmassa UV-mappaus. Sen jälkeen tähän UV-mappiin otetaan (bake) yksityiskohtaiset pinnanmuodot high-poly -versiosta. Tätä kutsutaan Normal-mapiksi. Normal-map on yleensä sinertävä kuva, jossa on talletettuna kaikki informaatiot pinnan muodoista ja yksityiskohdista. Tämän Normal-mapin avulla pelkistetympi ja kevyempi low-poly -versio peliasetista saadaan näyttämään korkealaatuisemmalta (kuva 3) kuin se oikeasti on. (Chopine 2011, s. 33-34; Mullen 2012, s. 126)



Kuva 3. Esimerkki siitä, miten Normal-map muuttaa tasaisen 2D-pinnan näyttämään yksityiskohtaisemmalta kuin se oikeasti on tekemättä assetista yhtään raskaampaa. Kuva opinnäytetyön tekijän omasta vapaa-ajan projektista Unreal Engine 4:ssä.

2.5 Häkki low poly-ja high-poly meshien mukana Normal-mapia tehdessä

Häkki (kuva 4) on suurennettu versio low-poly meshistä ja se ampuu säteitä sen sisällä olevan high-poly meshin pintaan. Täten saadaan high-poly meshin pinnanmuodot tehtyä 2D-kuvaksi eli Normal-mapiksi, vaikka il-mankin häkkiä voidaan tehdä Normal-mappeja. Häkin käyttöä perustellaan sillä, että säteiden suuntaa pystytään tällöin kontrolloimaan paremmin. (Blender Manual n.d.)



Kuva 4. Kuva wireframe -muodossa eli rautalankamallina olevasta häkistä, jonka sisällä näkyy limittäin pelkistetty low-poly malli ja yksityiskohtaisempi high-poly malli. Vieressä oikealla näkyy Normal-map. (Felinto 2014)

3 PELIMOOTTORI

Pelimoottori on ohjelma, jolla rakennetaan pelejä peliteollisuudessa. Se sisältää kaiken oleellisen mitä tarvitaan pelimaailman rakentamiseen kuten fysiikat, käyttäjän antama syöte peliin, renderöinti, graafinen käyttöliittymä, peliasettien suhteet toisiinsa sekä ympäröivään pelikenttään eli törmäysininformaatio, tekoäly esimerkiksi vihollisten käyttäytymiselle ja peliasettien animaatioiden säätäminen käyttäjän syötteille sekä tekoälylle sopivaksi. Pelimoottoreita on useita, mutta yleisimmät 3D-grafiikkaa varten olevat pelimoottorit ovat Unity, Unreal Engine, Source Engine ja Cryengine. (GameCareerGuide, 2008)

Aikaisemmin 90- luvun alussa käsite ”pelimoottori” tarkoitti käytännössä yksittäistä peliä, jonka asetteja ja ulkoisia ominaisuuksia pystyi muuttamaan omin päin sekoittamatta pelin ydintoimintoja ja mekaniikkoja. Tästä hyvänä esimerkkinä on peli nimeltä Doom. Sille kehittyi oma fanikuntansa, joita kutsuttiin ”modaajiksi”. Myöhemmin 90- luvun loppua kohti tultaessa julkaistiin pelit Quake III Arena ja Unreal. Molempien julkaisijat kehittivät pelinsä modausta silmällä pitäen. Epic Games Ltd kehitti ja julkaisi myöhemmin Unreal- pelinsä pohjalta tehdyn pelimoottorin nimeltä Unreal Engine. (Gregory, 2018, s. 11)

4 OHJELMISTOT

Koska tarkoituksena on saada Unreal Engine 4:ään peliaseteiksi CAD-malleja, joiden tiedostotyypeistä yksikään ei suoraan ole Blenderin kanssa yhteensopiva, niin tässä tutkitaan sivuavasti, minkä ohjelmien kautta nämä tiedostot saadaan muunnettua Blenderille sopivaksi. Opinnäytetyössä käytetään sekä 3D-mallinnusohjelma Blenderiä ja sen lisäksi myös verrataan muita ohjelmia, joilla saadaan tehtyä high-poly mallin pinnanmuodoista Normal-map low-poly mallille.

Taulukko 1. Tietoja ohjelmista, joita käytetään tässä opinnäytetyössä Normal-mappien tekemiseen (Blender n.d.c; Software.Informer, n.d.; Handplane3D, n.d.; Autodesk n.d.)

Nimi	Blender	xNormal	Handplane baker	3DS Max
Hinta/Lisenssi	Ilmainen myös kaupalliseen käyttöön	Ilmainen myös kaupalliseen käyttöön	Maksullinen, mutta asiakas saa itse päättää summan (vaikka 1\$), myös kaupalliseen käyttöön	Maksullinen ~1500-3300€/a, mutta opiskelijoille täysin ilmainen. Ei kuitenkaan tällöin kaupalliseen käyttöön.
Tuetut tiedostomuodot import	.dae, .abc, .3ds, .fbx, .bvh, .ply, .obj, .x3d/.wrl, .stl, .svg	.sia, .sib, .off, .ovb, .dxf, .3ds, .sbm, .obj, .ase, .ply, .lwo, .fbx, .lxo, stooage mesh, ogre mesh, .dae, .ms3d, directX mesh	.obj, .fbx	.fbx, .abc, .ai, .catpart, .catproduct, .cgr, .dae, .dem, .xml, .ddf, .dwg, .dxf, .flt, .htr, .ige, .igs, .iges, .ipt, .iam, .jt, .model, .mdl, .session, .exp, .dlv, .dlv3, .dlv4, .obj, .prt, .neu, .g, .asm, .rvt, .sat, .shp, .skp, .sldprt, .sldasm, .stl, .stp, .step,

				.trc, .wire, .wrl, .wrz
Tuetut tiedosto- muodot export	.dae, .abc, .3ds, .fbx, .bvh, .ply, .obj, .x30, .stl	(vain normal map vaihtoeht- dot, 2D kuvatiedostoja) .bmp, .jpg, .png, .hdr, .tga, .tiff, .raw, .dds, OpenEXR, .WebP	.tiff, .png, .psd, .tga	.fbx, .3ds, .abc, .ai, .ase, .ass, .dae, .dwf, .dwg, .dxf, .flt, .htr, .igs, .obj, .pxproj, .sat, .stl, .svf, .wrl

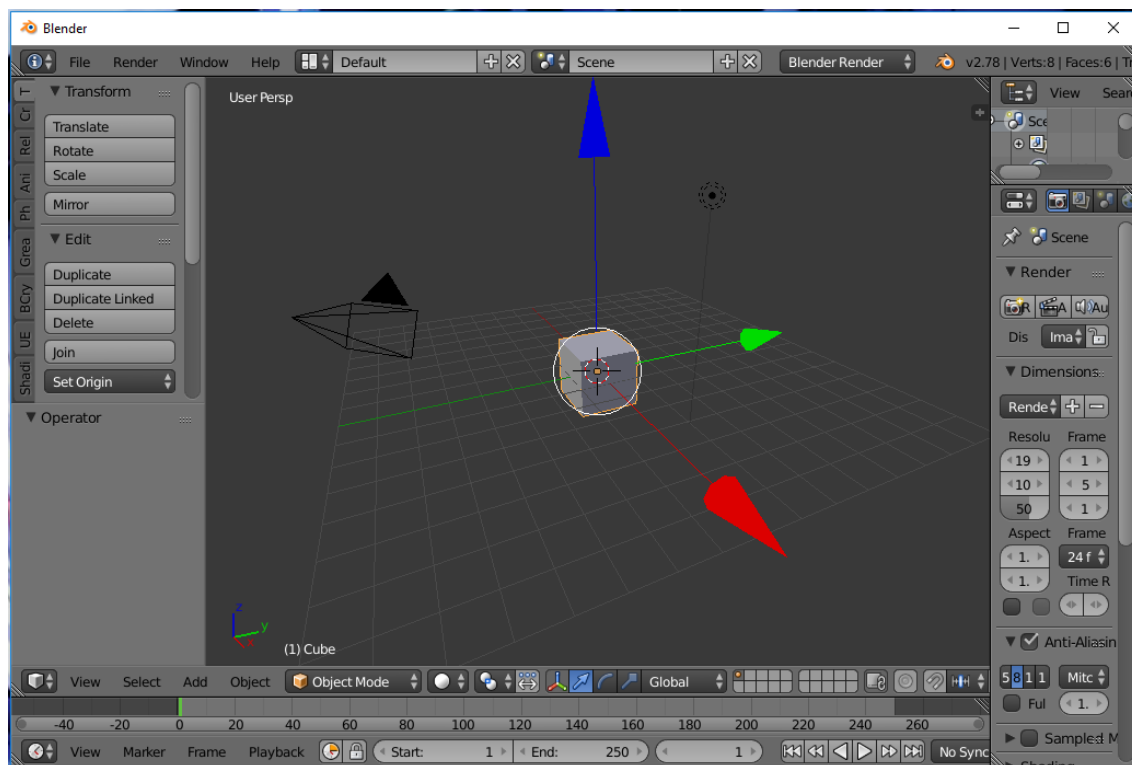
4.1 Blender 2.79

Blender on alankomaalainen vapaaehtoisten yhdessä tekemä 3D-mallin-
nusohjelma, jonka on tarkoitus helpottaa yksittäisiä tekijöitä ja pieniä aloit-
tavia yrityksiä tuottamaan 3D-grafiikkaa. Blender-säätiön on perustanut
Ton Roosendaal vuonna 1998 ja säätiö ottaa yhä edelleen vastaan lahjoit-
uksia ohjelman jatkokehitystä varten. (Blender n.d.c,a)

Blender on avoimen lähdekoodin ohjelma ja sen pystyy asentamaan kai-
kille kolmelle yleisimmälle käyttöjärjestelmälle (Windows, Mac OS X ja Li-
nux). Blenderiin saa paljon erilaisia lisäosia niin yksityistenkin kuin yritys-
tenkin valmistamina sekä ilmaiseksi että maksullisena. Blender on yleisesti
käyttäjien eteenpäin kehittämä ohjelma. Jotkut käyttäjien tekemistä lisä-
osista otetaan joskus virallisesti mukaan Blenderin päivityksiin. Lisäosia
pystyy tekemään python-ohjelmointikielellä. (Blender n.d.)

Blenderissä pystyy tekemään perus mallintamisen lisäksi riggausta (= ob-
jektin valmistelu animointia varten), animaatioita, simulaatioita, liiketun-
nistamista (= motion capture), ja onpa Blenderissä myös oma pelimootto-
rikin sisällä. (Blender n.d.)

Blender 2.79:ssä (kuva 5) on kaksi renderöintiengineä nimeltä Blender
Render ja Cycles, mutta tulevassa 2.80 päivityksessä nämä korvataan ko-
konaan uudella ja nopeammalla renderöintienginellä nimeltä Eevee. Sa-
malla koko käyttöliittymä muuttuu täysin uudenlaisemmaksi. (Blender
n.d.d)



Kuva 5. Blender- ohjelman version 2.79 yleisnäkymä.

4.2 Unreal Engine 4

Unreal Engine on Epic Gamesin rakentama pelimoottori, joka rakentui yrityksen pelin Unreal pohjalta. Alkuaikoina siinä oli jo sisäänrakennettu tekoäly, näkyvyys, törmäyksen tunnistaminen, verkkopeliominaisuus, renderöinti, skriptaus ja tiedostojen hallintajärjestelmä. Nykyään Unreal engine 4 (kuva 6) on yksi maailman edistyneimmistä ja käytetyimmistä pelimoottoreista. Se tuli kaikille ilmaiseksi käytettäväksi vuonna 2015. (MaximumPC 2009)

Sillä pystyy tekemään pelejä ja sovelluksia eri alustoille niin tietokoneille, mobiililaitteille kuin myös pelikonsoleille. Myös VR pelien ja sovellusten tekeminen onnistuu nykyään. Ohjelmointikielenä Unreal Engineissä käytetään C++:aa, mutta sen rinnalla on myös Blueprint- nodeista koostuva editori, jolla voidaan visuaalisella tavalla rakentaa pelimaailmaa tai sovellusta. (Epic Games, Inc. n.d.)



Kuva 6. Unreal Engine 4- pelimoottorin yleisnäkymä, jossa on jo rakennettu pelikenttää. Kuva opinnäytetyön tekijän omasta vapaa-ajan projektista.

Unreal Studio on Epic Gamesin tekemä ja ilmaisena betaversiona oleva lisäys ilmaiseen Unreal Engine 4:ään. Siihen pystyy tuomaan CAD-malleja kuten .3dm, .sldasm, .cgr, .catpart, .catproduct, .asm.1, .prt, .neu, .prt.1, .iam, .ipt, .gbf, .unv, .x_t, .rf, .iges, .step, .jt, .sat, .3dxml, .dwg ja .wire. kun normaalisti CAD-malleja ei pysty tuomaan pelimoottoriin. Näitä pystyy käsittelemään sitten peliasetteina. Tämä mahdollistaa suunniteltujen tuotteiden ja rakennuksien tarkastelun interaktiivisesti kuten pelissä. Tämä on erityisesti suunnattu eri alojen suunnittelijoille kuten autoteollisuuteen ja arkkitehdeille. Unreal Studio tulee olemaan kuukausimaksullinen kun betatestaus päättyy. (Epic Games, Inc. n.d.a)

4.3 FreeCAD

FreeCAD on täysin ilmainen avoimen lähdekoodin CAD-mallinnusohjelma. Sitä alettiin kehittää vuonna 2001 ja se on monen yksityisen käyttäjän yhdessä ylläpitämä tänä päivänä. Ohjelma on suunnattu mekaaniseen insinööri- ja tuotesuunnittelutyöhön, mutta se soveltuu silti myös arkkitehtikäyttöön. Se on ominaisuuspohjainen parametrinen mallintaja, jossa on modulaarinen arkkitehtuuri eli siihen saa lisätoimintoja muuttamatta ohjelman ydinsysteemiä kuitenkaan. FreeCAD toimii sujuvasti monella eri alustalla kuten Windowsin, Mac OS X:n ja Linuxin kanssa. Se tukee

seuraavia tiedostomuotoja kuten .iges, .stp, .obj, .dxf, .svg, .stl ja .dae. (FreeCAD n.d.; Fileinfo n.d.f)

4.4 xNormal

xNormal on myös ilmainen ohjelma, jolla pystyy tekemään Normal-mappeja sekä muitakin mappeja eli 3D-mallien pinnanmuotojen informaation tallentamista 2D-kuvaksi. Näitä ovat mm. Height-map ja Cavity-map. Ohjelma sisältää myös reaaliaikaisen ja interaktiivisen 3D-mallin tarkastelun sekä se tukee monta eri tiedostomuotoa (taulukko 1). xNormalilla pystyy sekä häkin kanssa että ilman häkkiä ottamaan (bake) informaation korkeatasoisen high-poly meshin tekstuureista ja siirtämään sen kevyempään low-poly meshiin Normal-mapin muodossa. Normaalisti molempien meshien topologiat (verteksien, särmien ja tahkojen määrä) niin low-polyssä kuin häkissäkin on oltava samat, mutta tässä ohjelmassa se ei ole välttämättömyys kun tekstuurin informaatiosta on kyse. (Software.Informer, n.d.)

4.5 Handplane 3D

Handplane on lähes ilmainen ohjelma, koska sen latauslinkin saa jos maksaa sen kehitystyöstä. Summan asiakas saa kuitenkin valita itse, koska sillä ei ole mitään määritettyä hintaa. Ohjelmaa kehitetään jatkuvasti ja kehittäjät ottavat mielellään käyttäjien palautetta vastaan tehdäkseen ohjelmastaan vieläkin paremman. Ohjelmalla pystyy tekemään Normal-mappien lisäksi muitakin 3D-mallin pinnanmuotojen informaation tallentamisia kuten Curvature-mappeja, Cavity-mappeja ja Height-mappeja. (Handplane3D, n.d.)

Ohjelma on 3D-artisteille suunnattu ja sen tarkoitus on helpottaa ja nopeuttaa asettien tekemisen workflowta. Ohjelma pystyy käsittelemään myös hyvin raskaita tiedostoja kuten heidän testeissään on ollut 10-20 miljoonaa kolmekulmaista polyonia. (Handplane3D, n.d.)

4.6 3DS Max

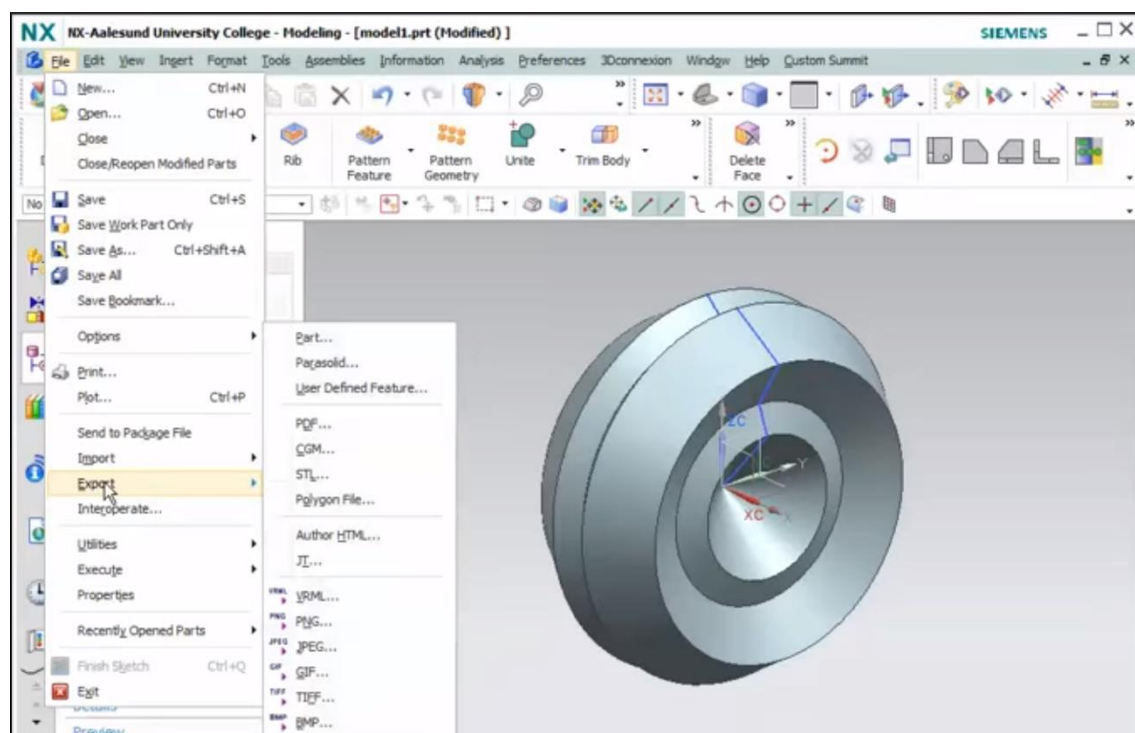
3DS Max on Autodeskin kehittämä 3D-mallinnusohjelma, joka on laajalti käytössä TV-, elokuva-, mainos- ja peliteollisuudessa. Se on opiskelijoille ilmainen ei-kaupallista käyttöä varten, mutta kaupalliseen käyttöön tarkoitettun lisenssin hinta vaihtelee 248€/kk summasta yli kolmen tuhannen euron vuosihintaan riippuen käyttäjän tarpeista. (Autodesk n.d.)

Ohjelmalla saadaan aikaan hyvin realistisia 3D-malleja ja niitä pystyy työstämään samoin kuin Blenderissäkin eli riggausta, animointia, simulaatioita ja liiketunnistamista. Tämän lisäksi ohjelmassa on monta muutakin ominaisuutta, joita ei löydy Blenderistä (esim. yhteensopivuus CAD-mallien kanssa suoraan). 3DS Max sopii yhteen muiden Autodeskin ohjelmien kanssa. (FinancesOnline n.d)

4.7 CAD-tiedoston vienti Blenderiin teoriassa

Ensimmäisenä on saatava CAD-tiedosto vietyä 3D-mallinnusohjelmaan, joten tarkoituksena on ensin tutkia, että miten joku annetuista tiedostomuodoista saadaan muunnettua Blenderille sopivaksi. Tähän tarvitaan jokin apuohjelma välille ja tässä opinnäytetyössä kokeiltavat ohjelmat ovat FreeCAD ja Unreal Studio. FreeCAD:n lisäksi mukaan ei oteta muita ilmaisia avoimen lähdekoodin CAD-ohjelmia, koska FreeCAD on niistä kaikista edistynein. Unreal Studio on Unreal Engine 4:ään kuuluva betatestauksessa oleva lisäosa, joka on suunnattu erityisesti pelikäyttöön tuleville CAD-malleille. (Fileinfo n.d.f; Epic Games, Inc n.d.)

Toinen tapa on hyödyntää Siemens NX 10 opiskelijalisenssiä, jolla pystyy viemään CAD-malleja .stl -tiedostoiksi (kuva 7). Blender kykenee tuomaan tätä tiedostotyyppiä. (TRED Print 2014)



Kuva 7. Siemens NX 9.0 version yleisnäkymä ja vientimahdollisuudet. (TRED Print 2014)

4.7.1 .prt

Creo Parametric Part File on PTC:n kehittämä tiedostomuoto. Tämä tiedostomuoto kuuluu CAD-mallinnuksen pariin ja sitä käytetään kun suunnitellaan tuotettavia ja kasattavia osia. Siinä on osan tiedot sisältäen 3D-mallin ja rakenteen. PRT -tiedostoja voidaan yhdistellä muiden osien kanssa assemblytiedostoksi, joka on .ASM. (Fileinfo n.d.a)

4.7.2 .x_t

Siemensin Parasolid CAD- tiedostomuoto, joka sisältää 3D-informaation, topologian ja värityksen. Tämä tiedostomuoto on suunnattu standardiksi export muodoksi jaettaessa Parasolid CAD-malleja. Tästä vanhempia versioita ovat XMT_TXT ja XMP_TXT -tiedostomuodot. X_T -tiedostot ovat tallennettu tekstimuodossa ja niitä voidaan tuoda ja viedä lukuisiin eri CAD-ohjelmiin. X_T -tiedostoja voidaan myös tallentaa binäärimuodossa .X_B -tiedostoiksi. Nämä ovat kyllä harvinaisempia kuin X_T -tiedostot. (Fileinfo n.d.b)

4.7.3 .stp

STEP- niminen tiedostomuoto, joka toimii laajalti eri CAD-suunnitteluohjelmissa. Se sisältää kolmiulotteisen mallin tiedot ja joskus teollisuudessa STP -tiedostoja käytetään suunnitellun mallin tai muotin hinnoittelussa. STP -tiedosto on kokoonpano useammasta osasta. Tätä käytetään pääosin kun siirretään 3D-grafiikkaa eri CAD-ohjelmien välillä. (Fileinfo n.d.c)

4.7.4 .jt

Siemens PLM ohjelmistoon kehitetty hyvin kevyt tiedostomuoto 3D-mallille. Auttaa tarkastelemaan ja pyörittelemään suunniteltua CAD-mallia tehokkaasti. Näyttää melko tarkkaa informaatiota mallin pinnanmuodoista sisältää muutakin informaatiota objektin tuotannosta ja metadataa. Tätä tiedostomuotoa käytetään pääosin auto- ja lentokone-teollisuudessa, mutta sopii myös muillekin teollisuudenaloille. (Fileinfo n.d.d)

4.7.5 .stl

Steliotopografia tiedostotyyppi. Suunnitteluformaatti, joka on nopea prototyyppiteknologia. Sillä tuotetaan kolmiulotteisia osia, jotka sisältävät 3D suunnitelman tai mallin. Tätä voidaan käyttää prototyyppien tuottamiseen. Toimii erittäin monessa eri ohjelmassa monella eri alustalla. (Fileinfo n.d.e)

4.7.6 .obj

Wavefront 3D tiedostomuoto, joka on varsin yleisessä käytössä hyvin monessa eri 3D-mallinnusohjelmassa. Sitä käytetään 3D-mallien siirtämiseen ohjelmasta toiseen ja se sisältää 3D-mallin muodon lisäksi koordinaatit, tekstuurimapit, polygonit ja muut pintamateriaalitiedot. Tähän tiedostomuotoon vieminen luo samalla .mtl -tiedoston, joka sisältää kaikki materiaalitiedot. (Fileinfo n.d.)

5 CAD-MALLI BLENDERIIN JA PELIASSETIKSI

Tässä luvussa vertaillaan eri ohjelmia, joilla saadaan ensinnäkin CAD-malli vietyä Blenderiin, jossa sitä on tarkoitus työstää edelleen päämääränä saada siitä peliasetti Unreal Engine 4:ään. Sen lisäksi tutkitaan apuohjelmia, joilla saadaan tehtyä high-poly meshistä Normal-map eli pinnanmuodot low-poly meshiin. Verrataan eri ohjelmilla tehtyjä Normal-mappeja ja sitä, millä ohjelmalla saadaan paras lopputulos.

CAD-mallista saatiin toimeksiantajalta neljä eri tiedostomuotoa (.x_t, .jt, .stp ja .prt) ja ensimmäinen tehtävä oli löytää tiedostomuoto, joka olisi yhteensopiva Blenderin kanssa, koska Blender on ensisijainen 3D-mallinnustyökalu tässä opinnäytetyössä ilmaislisenssinsä vuoksi. Tämä oli melko ongelmallista, koska näistä tiedostomuodoista yksikään ei löytynyt Blenderin tuontivaihtoehtoista. Täytyi alkaa etsiä sopivaa apuohjelmaa välikappaleeksi.

5.1 Alkuvaikeudet

Ensimmäisenä otettiin kokeiluun FreeCAD ohjelma, koska siihen pystyy tuomaan .stp -tiedoston ja koska se on täysin ilmainen ohjelma. 3D-malli näkyi FreeCAD:ssa kuten pitikin. Myös materiaalitiedot olivat näkyvissä. FreeCADin vientivaihtoehtoista valittiin ensin .stl -tiedostomuoto. 3D-malli näkyi Blenderiin tuotaessa ilman materiaalitietoja ja monet osat kyseisestä mallista olivat vaihtaneet sijaintia ja asentoa. Joko FreeCAD ei osaa pakata CAD-mallin osien sijaintitietoja oikein tai sitten Blender ei osaa tulkita niitä. Seuraavaksi kokeiltiin viedä malli FreeCAD:sta Blenderiin .obj -tiedostomuotodolla ja lopputulos oli sama. FreeCAD:sta ei ollut apuohjelmaksi tässä tapauksessa. Seuraavana vaihtoehtona olisi ollut kokeilla Siemens NX 10 ohjelman opiskelijalisenssiä. Kävi ilmi, että kyseisellä lisenssillä ei pysty tuomaan ohjelmaan omia CAD-malleja, vaan ohjelmaa voi testata ainoastaan Siemensin valmiilla esimerkkimalleilla. Tämäkään ohjelma ei siis soveltunut haluttuun käyttötarkoitukseen.

Opinnäytetyöajan alkupuolesta hyvin suuri osa meni sopivan apuohjelman etsimiseen annettujen rajojen mukaisesti. FreeCAD oli ainoa, jota kokeiltiin kaikista avoimen lähdekoodin ilmaisohjelmista, koska se on kaikista pisimmälle kehitetty kaikista niistä ja koska tämän opinnäytetyön pääpiste on enemmän Normal-mapeissa sekä itse 3D-mallintamisessa high-poly -asestista low-poly peliassetiksi. Tarkoitus ei siis ollut käyttää liikaa aikaa sen tutkimiseen, että mikä ilmainen ohjelma saa CAD-mallin käännettyä Blenderille sopivaksi.

5.2 Toimiva ja toistaiseksi ilmainen apuohjelma

Lopuksi päädyttiin kokeilemaan Unreal Engine 4 uutta lisäominaisuutta eli Unreal Studiota, koska eräs toinen opiskelija kertoi sen mahdollisuuksista CAD-mallien viemisessä suoriksi 3D-malleiksi esimerkiksi Blenderiin. Kyseisen ohjelman tuontivaihtoehdoista .x_t oli ainoa, joka toimi saadun CAD-mallin kanssa. Mallin pystyi jakamaan haluttuihin osiin ja viemään nämä ryhmitellyt osat .fbx muodossa Blenderiin. Materiaalitiedot eivät kuitenkaan tallentuneet mukaan, mutta tällä kertaa osien sijainnit ja asennot olivat juuri kuten pitikin.

CAD-mallin vienti Blenderiin onnistuu myös Autodeskin ohjelmien opiskelijalisenssiä hyödyntäen. Tällöin CAD-malli on vietävä ensin 3DS Max-ohjelmaan, josta sen voi viedä .fbx muotoisena Blenderiin. Tästä ei kuitenkaan ole opinnäytetyön kannalta mitään erityistä hyötyä.

5.3 High-poly mesh ja low-poly mesh

Kun CAD-malli oli saatu Blenderiin vietyä onnistuneesti niin tehtävänä oli tehdä siitä peliasetti. CAD-mallista alettiin poistaa kaikkea turhaa sisältöä, jotka eivät kuuluneet peliasettiin. Tämä tarkoittaa kaikkia niitä alueita meshistä, joiden ei ole tarkoitus olla visuaalisesti näkyvissä peliasetissa.

Tässä vaiheessa 3D-malli näytti siltä, että se on valmis seuraavaa vaihetta varten, eli siitä piti tehdä low-poly mesh. Opinnäytetyön tekijä ei ollut koskaan aiemmin tehnyt peliasettia, joten hän ei oikein tiennyt kuinka tarkasti low-poly meshin olisi jäljiteltävä high-poly meshin ulkomuotoja. Koska toimeksiantajalta saatu CAD-malli on rengas vanteineen ja pultteineen, mallinnettiin renkaan kumiosan nappulat ja vanteissa olevat pultit jokseenkin yksityiskohtaisesti. Loput low-polyistä jätettiin hyvin suurpiirteiseksi. Opinnäytetyön tekijän kokemattomuuden vuoksi hän ei tiennyt, mikä olisi hyvä määrä polygoneja low-poly meshissä, jotta se ei olisi liian raskas. Low-poly mesh pyrittiin tekemään silmämääräisesti mahdollisimman simppeleksi säilyttäen jokseenkin muotonsa. Normaalisti peliasettien low-poly versioita valmistettaessa kaikille aseteille asetetaan oma maksimi polygonmäärä (kuva 8), jota ei saa ylittää. Tämä auttaa siinä, että pelistä tulee sulavasti toimiva eikä yksittäiset assetit tee pelistä raskasta ja hidasta. Tässä opinnäytetyössä valmistetaan vain yksi assetti eikä tässä ole sovittu erikseen polygonien maksimimäärää asetilte.

Asset Type	Max Polycount (Pre-LOD)					Unique features/considerations
	Hero	Foreground	Midground	Background	Mobile	
AAA - 8th Gen Console (PS4/Xbone/PC)	Hero	Foreground	Midground	Background	Mobile	
AAA - 7th Gen Console (PS3/360/Nintendo Switch)	Hero	Foreground	Midground	Background	Background	
VR - High end		Hero	Foreground	Background	Background	
Mobile - High End				Hero	Other	
Mobile - Low End/VR					Budget Cap	
Character - Humanoid	100k	50k	20k	10k	10k	Hair/fur, Subsurface scattering, rigging, cloth physics
Creature - 1m	50k	20k	10k	2k	5k	AAA
Creature - Human +	75k	50k	20k	5k	5k	AAA
Creature - Giant	150k	75k	50k	10k	5k	AAA

Kuva 8. Esimerkkinä pieni osa Polygon budget- esimerkkitaulusta, jossa on sovitut polygonien maksimimäärät eri asseleille. (Bickle 2017)

Low-poly meshistä tehtiin kaksi versiota. Toisen polygonit olivat quad-muodossa ja toisessa ne olivat trianguloituna. Vaikka opinnäytetyössä on tarkoitus tutkia mahdollisimman sujuvaa ja nopeaa tapaa työstää CAD-mallista peliassetti, päätettiin opinnäytetyöhön lisätä tämä vaihe mukaan. Näin saadaan lopulta Normal-mappien laatuihin hieman enemmän eroja ja vertailua. Trianguloitua versiota tarvittiin Handplane ohjelman testaamiseen, sillä se ei hyväksynyt quad polygoneja.

Kun low-poly mesh oli valmis, oli aika merkata siihen terävät reunat ja saumat, joiden myötä kyseinen 3D-malli avataan UV-mapiksi. Alun perin oli tarkoitus tutkia, kannattaako tätä vaihetta tehdä jossain apuohjelmassa Blenderin lisäksi. Päätettiin olla keskittymättä siihen ja tehdä UV-mappaus vain ja ainoastaan Blenderissä, koska opinnäytetyölle varattu ajanjakso oli jo tässä vaiheessa liian pitkällä. Aika ei olisi yksinkertaisesti riittänyt monen eri vaiheen apuohjelmia varten.

5.4 Normal-mapit eri ohjelmissa

Peliassettia eli low-poly meshiä varten tarvitaan Normal-map high-poly meshistä. Normal-mappeja tehtiin useammalla eri asetuksella. Alkuperäisestä CAD-mallista on tehty low-poly muodossa sekä quad- että trianguloitu mesh. Molemmille versioille piti tehdä myös häkit kokeilun vuoksi. Normal-mapin voi tehdä joko suoraan high-polysta low-polyyn, mutta myös erillisen häkin avulla. Häkin käyttöä perustellaan pinnanmuotoa tallentavien säteiden paremmalla ohjattavuudella.

Testattaviksi apuohjelmiksi Normal-mappeja varten valittiin apuohjelmat xNormal ja Handplane 3D, koska xNormal on yleisesti alan harrastajien ja ammattilaisten keskuudessa kehuttu sekä ilmainen ohjelma ja Handplane on pienen maksumuurin takana. Ostaja saa kuitenkin itse valita summan, jonka maksaa ohjelmasta. Tämä tekee siitä lähes ilmaisen. Näiden lisäksi kokeiltiin sekä Blenderin että 3DS Maxin sisäisiä Normal-map baketoimintoja. Kaikilla kokeilukerroilla säteen pituus asetettiin maksimiin eli 1.000 mittaiseksi, koska tätä pienemmillä arvoilla joitakin pintoja jäi

kartoittamatta. Sen lisäksi xNormalissa oli mahdollisuus pehmentää särmiä, joten opinnäytetyön tekijä päätti tehdä erän Normal-mappeja tälläkin asetuksella.

Jotta lukijalle Normal-mappien vertailu olisi selkeämpää niin pisteytetään laatuvaatimuskriteerit. Katvealueista puhuttaessa Liite 1 eli kuvat muutamasta Normal-mapista selkiyttävät käsitettä.

- 1 = Erinomainen lopputulos, hyvin sileä jälki (lähinnä renkaan sivuissa), katvealueita vähän tai ei ollenkaan.
- 2 = Normal-map asettuu hyvin ja näyttää oikealta, mutta jälki ja pinnanmuodot on karkeahkoa (huomaa tässä lähinnä renkaan sivut). Katvealueita voi olla, mutta lopputulos assetissa näyttää ok: lta.
- 3 = Normal-mapissa karkeutta ja selkeitä virheitä. Katvealueita ja virheitä on jonkun verran tai paljon.

Taulukko 2. 3D-mallinnusohjelmilla tehtyjä Normal-map testejä

Tekotapa (bake normals, säde 1.0)	Blender	3DS Max
Quad, ei häkkiä, laatu	Selkeä ja siisti kuva, kovin syviä pinnanmuotoja ei tallentunut, mutta säde ylsi laajalti kaikkialle, vähän jäi renkaan nappuloiden sivuja katveeseen. Arvosana 1.	Selkeä ja siisti kuva, renkaan nappuloissa oli joitakin kohtia mihin säde ei osunut. 3DS Maxin sisäinen häkki eli projection modifier. Arvosana 1 Normal-mapia tarkastellessa, mutta UE4:ssa sileä jälki.
Quad, ei häkkiä, nopeus	Ei nopeustietoja saatavilla	Ei nopeustietoja saatavilla
Quad, häkki, laatu	Paljon informaatiota puuttuu ja säde ei yltänyt kaikkialle. Arvosana 2.	Erillistä käsin tehtyä mesh häkkiä ei pystynyt hyödyntämään.
Quad, häkki, nopeus	Ei nopeustietoja saatavilla	Ei nopeustietoja saatavilla
Tri, ei häkkiä, laatu	Selkeä ja siisti kuva, kovin syviä pinnanmuotoja ei tallentunut, mutta säde ylsi laajalti kaikkialle.	Selkeä ja siisti kuva, renkaan nappuloissa oli joitakin kohtia, mihin säde ei osunut. 3DS

	Vähän jäi renkaan nappuloiden sivuja katveeseen. Paljon parempilaatuinen kuin quad- meshillä tehty, yksi parhaimmista ellei paras kun tarkastelee meshin päällä Unreal Enginessä. Arvosana 1.	Maxin sisäinen häkki eli projection modifier. Arvosana 1 Normal-mapia tarkastellessa, mutta UE4: ssa sileä jälki.
Tri, ei häkkiä, nopeus	Ei nopeustietoja saatavilla	Ei nopeustietoja saatavilla
Tri, häkki, laatu	Paljon informaatiota puuttuu ja säde ei yltänyt kaikkialle. Parempilaatuinen kuin quad meshillä tehty. Arvosana 1.	Erillistä käsin tehtyä mesh häkkiä ei pystynyt hyödyntämään.
Tri, häkki, nopeus	Ei nopeustietoja saatavilla	Ei nopeustietoja saatavilla

Taulukko 3. Erillisissä baking ohjelmissa tehtyjä Normal-map testejä

Tekotapa (bake normals, säde 1.0)	xNormal	Handplane
Quad, fbx, ei häkkiä, laatu	Hyvin vähän katvealueita, yksi parhaimmista Normal-mapeista. Arvosana 1.	Ei ota vastaan quad- muodossa olevia 3D-malleja
Quad, fbx, ei häkkiä, nopeus	35,6s	Ei ota vastaan quad- muodossa olevia 3D-malleja
Quad, fbx, häkki, laatu	Paljon puutteita renkaan nappulapinnoissa, säde ei yltänyt kaikkialle. Näyttää UE4: ssa kuitenkin ihan siedettävältä. Arvosana 2.	Ei ota vastaan quad- muodossa olevia 3D-malleja

Quad, fbx, häkki, nopeus	33,14s	Ei ota vastaan quad- muodossa olevia 3D-malleja
Quad, obj, ei häkkiä, laatu	Laadukas Normal-map, hyvin vähän säteen ulottumattomiin jääneitä kohtia. Arvosana 2, melkein 1.	Ei ota vastaan quad- muodossa olevia 3D-malleja
Quad, obj, ei häkkiä, nopeus	32,95s	Ei ota vastaan quad- muodossa olevia 3D-malleja
Quad, obj, häkki, laatu	Paljon puutteita renkaan nappulapinnoissa, säde ei yltänyt kaikkialle. Arvosana 2.	Ei ota vastaan quad- muodossa olevia 3D-malleja
Quad, obj, häkki, nopeus	33s	Ei ota vastaan quad- muodossa olevia 3D-malleja
Tris, fbx, ei häkkiä, laatu	Kaikista paras Normal-map yhdessä obj muodossa olevan version kanssa. Arvosana 1.	Paras lopputulos kaikista Handplannella tehdyistä Normal-mapeista, vain vähän katvealueita renkaan nappulapinnoissa. Arvosana 2. Näyttää UE4: ssa melkein 1 arvosanan ansaitsevalta.
Tris, fbx, ei häkkiä, nopeus	32,64s	4,34s
Tris, fbx, häkki, laatu	Jonkun verran puutteita renkaan nappulapinnoissa, säde ei yltänyt kaikkialle. Arvosana 2.	Paljon puutteita renkaan nappulapinnoissa, säde ei yltänyt kaikkialle. Arvosana 2.

Tris, fbx, häkki, nopeus	33,45s	4,35s
Tris, obj, ei häkkiä, laatu	Toiseksi paras Normal-map yhdessä fbx muodossa olevan version kanssa. Arvosana 1.	Huonoin lopputulos kaikista. Paljon alueita, joihin säde ei yltänyt. Arvosana 3.
Tris, obj, ei häkkiä, nopeus	32,94s	3,52s
Tris, obj, häkki, laatu	Paljon puutteita renkaan nappulapinnoissa, säde ei yltänyt kaikkialle. Arvosana 2.	Paljon puutteita renkaan nappulapinnoissa, säde ei yltänyt kaikkialle. Arvosana 2.
Tris, obj, häkki, nopeus	33,26s	3,09s

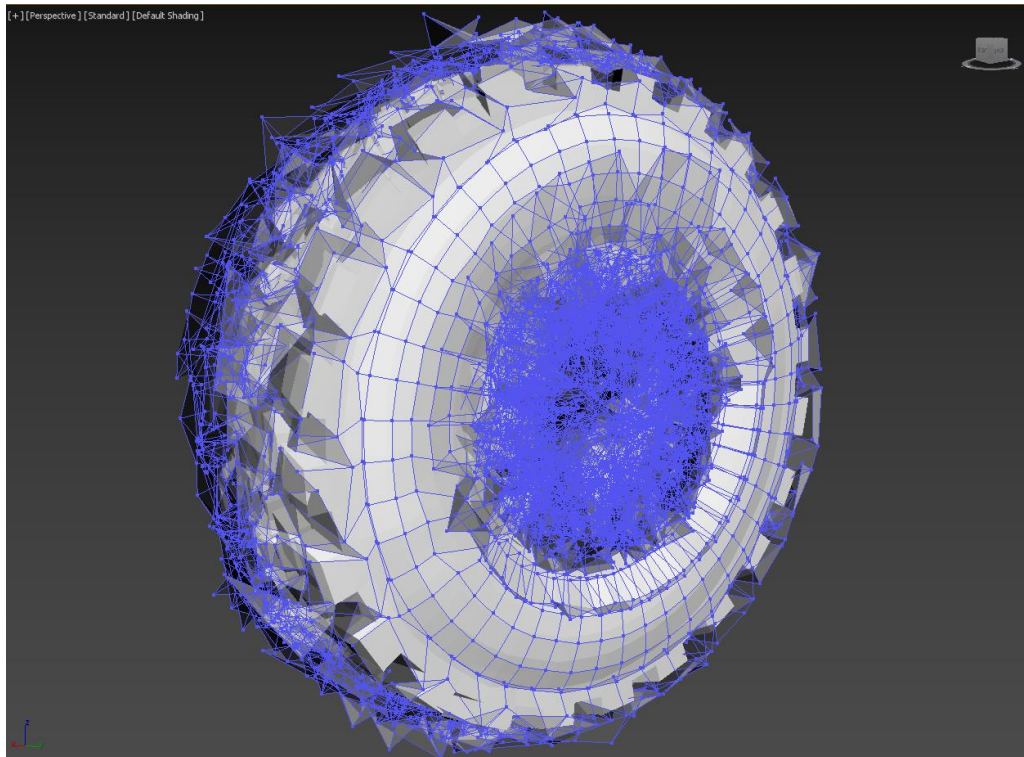
6 LOPPUTULOS

Kaikista paras tulos saatiin opiskelijoille ilmaisella 3DS Max -ohjelmalla ja sen projection modifierilla, joka luo automaattisesti häkin meshin päälle. Ilmaisista ohjelmista parhaimmat tulokset tekivät Blenderin sisäinen bake toiminto sekä xNormal ohjelma trianguloidulla meshillä ilman häkkiä. Häkittömyys tekee prosessista myös nopeamman. Luultavasti oikeanlaisen häkin kanssa lopputuloskin näyttäisi paremmalta. Tekijän vähäinen kokemus peliasettien tekemisessä on luultavasti syynä siihen, että häkin kanssa tehdyt kokeilut tässä opinnäytetyössä eivät ehkä tuoneet parasta mahdollista lopputulosta. Häkillä kun pystyy ohjaamaan tarkemmin ammuttavien säteiden suuntaa ja täten Normal-mapistakin tulee kattavampi ja tarkempi.

Handplane ei hyväksynyt quad- muodossa olevia meshejä, vaan niiden oli oltava trianguloituja. Huonommat tulokset johtuvat todennäköisesti siitä, että kyseinen ohjelma ei ole niin pitkälle kehitetty kuin xNormal. Sen lisäksi Handplanessa ei säädetty kaikkia asetuksia, jotka olisivat mahdollisesti vaikuttaneet lopputulokseen suotuisammin. Koitettiin pitää asetukset mahdollisimman yhdenmukaisina kaikissa ohjelmissa.

3DS Maxissa ei pystynyt käyttämään erillistä käsin tehtyä häkkimeshiä vaan siinä oli projection modifier. Kyseinen asetus luo automaattisesti low-poly meshistä häkin, mutta tässä tapauksessa se automaattisesti luotu häkki näyttää kamalalta. (kuva 9). Normal-mappeja verrattiin rinnakkain Unreal Engine 4:ssä. (kuva 10)

High-poly mallissa oli 60024 tri polygonia. Käsin mallinnetussa low-poly mallissa oli 11501 tri polygonia. Vähennys oli siis suunnilleen ~81%. Peliasetiksi renkaassa on ihan sopiva määrä polygoneja. Alkuperäinen CAD-malli olisi vaatinut huomattavasti enemmän tehoja tietokoneelta ilman vähennystä.



Kuva 10. 3DS Maxin Projection modifier, joka luo automaattisesti häkin ja tässä tapauksessa renkaan muoto oli ongelmallinen sille. Lopputulokset Unreal Engine 4:ssä oli kuitenkin paras kaikista. Kuva omasta kokeilusta.



Kuva 9. Viisi erilaista Normal-mappia vertailussa Unreal Engine 4 pelimoottorissa. Renkaalla ei ole tekstuureja tai sen omia värejä, koska niitä ei pystynyt tuomaan Blenderiin. xNormal, ei häkkiä eikä pehmennysasetuksia (ylä vas.), Handplane, häkin kanssa (ylä kesk.), 3DS Max, ei häkkiä (ylä oik.), xNormal, ei häkkiä ja pehmennysasetus päällä (ala vas.), high-poly mesh (ala kesk.) ja Blender, ei häkkiä (ala oik.).

7 YHTEENVETO

Opinnäytetyön tutkimuskysymyksiin saatiin vastattua vaihtelevalla menestyksellä. Ensimmäisenä tutkimuskysymyksenä oli löytää opetuskäyttöön sopivia ja mieluiten ilmaisia ohjelmia, joilla työ CAD-mallista peliasetiksi Unreal Engine 4:ään onnistuu. Ilmaisia ohjelmia löytyi siitä vaiheesta alkaen, kun CAD-malli on saatu jo Blenderiin. Tämä oli myöskin opinnäytetyön pääpainoalue, vaikkakin opinnäytetyön tekijä halusi ottaa mukaan myös tutkittavaksi alkuvaiheen eli miten CAD-malli saadaan ilmaisohjelma Blenderiin. Tähän liittyvät ongelmat tulevat ilmi opinnäytetyön käytännönsuudessa. Toisena tutkimuskysymyksenä täytyi löytää ratkaisu mahdollisimman sujuvaan työtapaan tehdä CAD-mallista peliasetti pääpainona nimenomaan high-polyistä low-poly peliasetti Normal-mappeineen. Opinnäytetyön tekijän kokemattomuus peliasettien tekemisestä vaikutti oleellisesti lopputulokseen ja tämäkin tulee ilmi käytännönsuudessa. Nopeampi työtapa tuo siedettävän ja onnistuneen lopputuloksen, mutta yhden välivaiheen lisäys olisi tehnyt asetista paljon hienomman ja tähän opinnäytetyön tekijä ei lähtenyt, koska se olisi vaatinut hyvin paljon manuaalista työtä ja aikaa. Tällöin toiseen tutkimuskysymykseen voisi siis sanoa, että nopeammalla tavalla voi tehdä keskinkertaisen taseisia peliasetteja ilman tätä välivaihetta.

Tätä opinnäytetyötä tehdessä tekijä oppi CAD-mallinnuksen ja suoran 3D-mallintamisen väliset erot. Aikaisempi kokemus peliasettien tekemisestä oli lähinnä sitä, että pyritään tekemään peliasetista jo 3D-mallinnusohjelmassa mahdollisimman vähäpolygoninen ja tämän jälkeen Unreal engine 4:ssä määritetään vielä Level of Distance asetukset sille. Uutena asiana tuli siis Normal-mappien tekeminen high-poly mallista low-poly mallille ja siihen liittyvä mahdollinen häkin käyttäminen apuvälineenä. Tekijä on aiemmin käyttänyt pääosin Blenderiä harrastusprojektinsa 3D-mallinnusohjelmassa ja kokeillut joskus monta vuotta aiemmin 3DS Maxia, joten tämänkin ohjelman käyttö tuli tekijälle laajemmin tutuksi.

Tämä opinnäytetyö on hyödyllinen lähinnä siinä mielessä, että tämä koostuu monta 3D-mallintamisen harrastajan YouTubevideoa ja foorumikeskustelua tutkimukseksi oikeita lähteitä käyttäen. 3D-mallinnuksen kirjallisuus nykyaikana on hyvin suppeaa ja jopa kenties hieman vanhanaikaista, koska ohjelmat kuten Blender päivittyvät jatkuvasti ja suuri osa uudesta informaatiosta on ohjelmien julkaisijoiden virallisten nettisivujen lisäksi lähinnä epävirallisissa yksityishenkilöiden tekemissä blogipostauksissa, videoissa ja keskustelupalstoilla.

8 LÄHTEET

Autodesk (n.d.) 3DS MAX. 3D modeling, animation, and rendering software. Haettu 18.3.2019 osoitteesta

<https://www.autodesk.com/products/3ds-max/overview>

Ben Bickle (2017). Polycounts for Real-Time Content – Looking for Feedback. Haettu 23.4.2019 osoitteesta

<https://blog.turbosquid.com/2017/06/01/polycounts-real-time/>

Blender Foundation (n.d.). About. Haettu 26.1.2019 osoitteesta

<https://www.blender.org/about/>

Blender Foundation (n.d.a). Blender's 25th birthday! Haettu 26.2.2019 osoitteesta

<https://www.blender.org/press/blenders-25th-birthday/>

Blender Foundation (n.d.b). Blender Funding. Haettu 26.1.2019 osoitteesta

https://www.blender.org/wp-content/uploads/2019/01/blender_funding_2019_01_28.png?x96075

Blender Foundation (n.d.c). Foundation. Haettu 26.1.2019 osoitteesta

<https://www.blender.org/foundation/>

Blender Foundation (n.d.d). What's New in Blender 2.80. Haettu 26.2.2019 osoitteesta

<https://www.blender.org/2-8/>

Blender Manual (n.d.). Render Baking. Haettu 3.3.2019 osoitteesta

<https://docs.blender.org/manual/en/latest/render/cycles/baking.html>

Chopine, A. (2011). 3D Art Essentials. The Fundamentals of 3D Modeling, Texturing, and Animation. Amsterdam: Elsevier.

Dalai Felinto. (2014). Blender 2.71 - Cycles Baking using Cages. Haettu 14.3.2019 osoitteesta

<https://www.youtube.com/watch?v=jOMnNb82DRI&t=291s>

Epic Games, Inc. (n.d.). Frequently Asked Questions (FAQ). Haettu 26.1.2019 osoitteesta

<https://www.unrealengine.com/en-US/faq>

Epic Games, Inc. (n.d.a). Unreal Studio. Haettu 18.2.2019 osoitteesta

<https://www.unrealengine.com/en-US/studio?sessionInvalidated=true#datasmithOverview>

Fileinfo (n.d.). .OBJ File Extension. Haettu 18.2.2019

<https://fileinfo.com/extension/obj>

Fileinfo (n.d.a). .PRT File Extension. Haettu 6.2.2019 osoitteesta

<https://fileinfo.com/extension/prt>

Fileinfo (n.d.b). .X_T File Extension. Haettu 6.2.2019 osoitteesta

https://fileinfo.com/extension/x_t

Fileinfo (n.d.c). .STP File Extension. Haettu 6.2.2019 osoitteesta <https://fileinfo.com/extension/stp>

<https://fileinfo.com/extension/stp>

Fileinfo (n.d.d). .JT File Extension. Haettu 6.2.2019 osoitteesta

<https://fileinfo.com/extension/jt>

Fileinfo (n.d.e). .STL File Extension. Haettu 6.2.2019 osoitteesta

<https://fileinfo.com/extension/stl>

Fileinfo (n.d.f) FreeCAD. Haettu 10.2.2019 osoitteesta

<https://fileinfo.com/software/freecad/freecad>

FreeCAD (n.d.). About FreeCAD. Haettu 18.2.2019 osoitteesta

https://www.freecadweb.org/wiki/About_FreeCAD

FinancesOnline (n.d) Autodesk 3ds Max REVIEW. Haettu osoitteesta 18.3.2019

<https://reviews.financesonline.com/p/autodesk-3ds-max/>

GameCareerGuide (2008) What is a Game Engine? Haettu 6.3.2019 osoitteesta

http://www.gamecareerguide.com/features/529/what_is_a_game_.php

Gregory, J. (2018). Game Engine Architecture, Third Edition. Chapman and Hall/CRC.

<https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/reader.action?docID=5150818>

Handplane3D (n.d.). Overview. Haettu 6.3.2019 osoitteesta

<http://www.handplane3d.com/>

Mark Brunelli (2017). Parametric vs. Direct Modeling Which Side Are You On?. Haettu 9.2.2019 osoitteesta

<https://www.ptc.com/en/cad-software-blog/parametric-vs-direct-modeling-which-side-are-you-on>

MaximumPC (2009). Doom to Dunia: A visual History of 3D Game Engines. Haettu 26.1.2019 osoitteesta

https://web.archive.org/web/20090724065520/http://www.maximumpc.com/article/features/3d_game_engines?page=0%2C3

Michael Alba (2018). What's the difference Between Parametric and Direct Modeling?. Haettu 9.2.2019 osoitteesta

<https://www.engineering.com/DesignSoftware/DesignSoftwareArticles/ArticleID/16587/Whats-the-Difference-Between-Parametric-and-Direct-Modeling.aspx>

Mullen, T. (2012). *Mastering Blender*. New York City: Wiley.

<https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/reader.action?docID=947869>

PCMAG (2018). Definition of: parametric modeling. Haettu 9.2.2019

<https://www.pcmag.com/encyclopedia/term/48839/parametric-modeling>

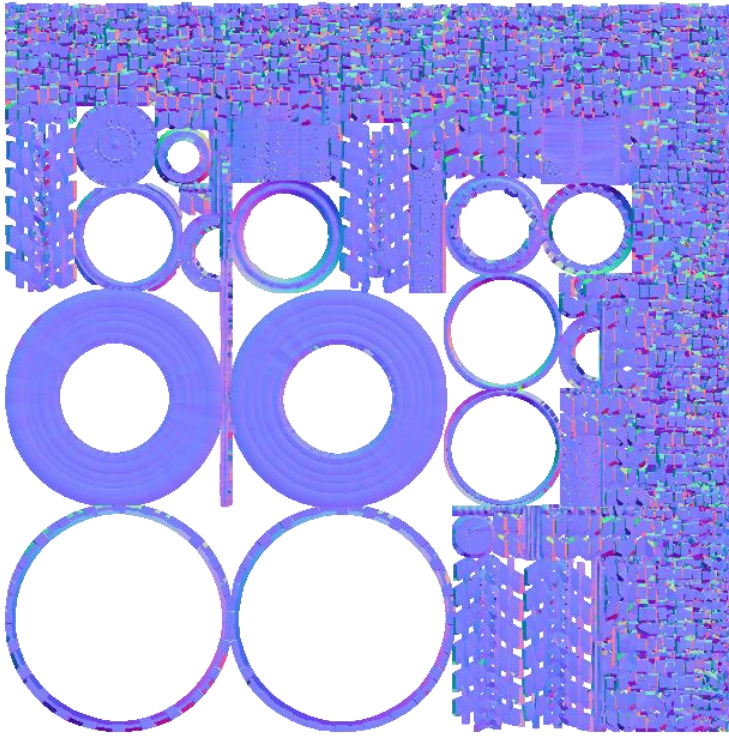
Software.Informer (2019). xNormal 3.17. Haettu 6.3.2019 osoitteesta

<https://xnormal.software.informer.com/>

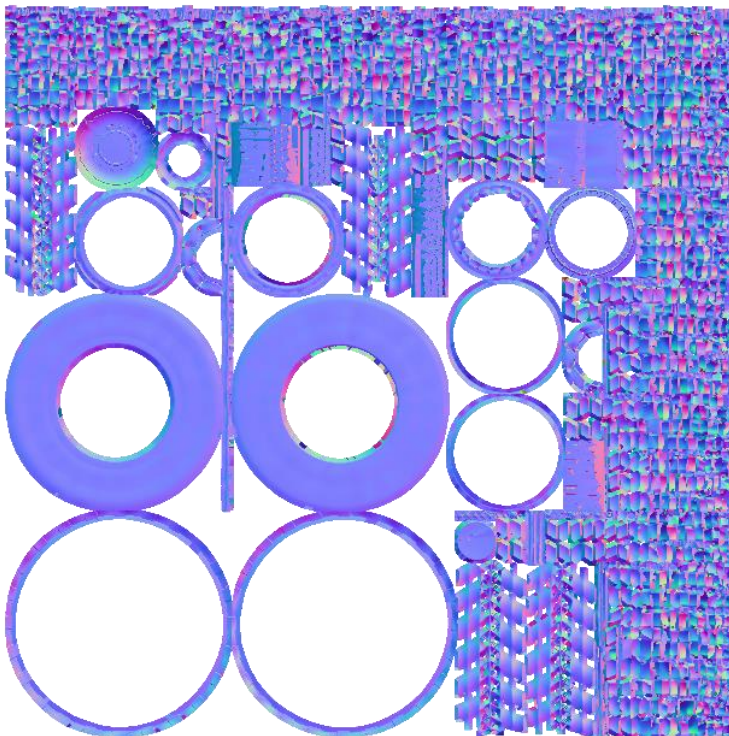
TRED Print. (2014). Siemens NX 9.0 – Export to STL. Haettu 9.2.2019 osoitteesta

<https://www.youtube.com/watch?v=wIWgKkYiZys>

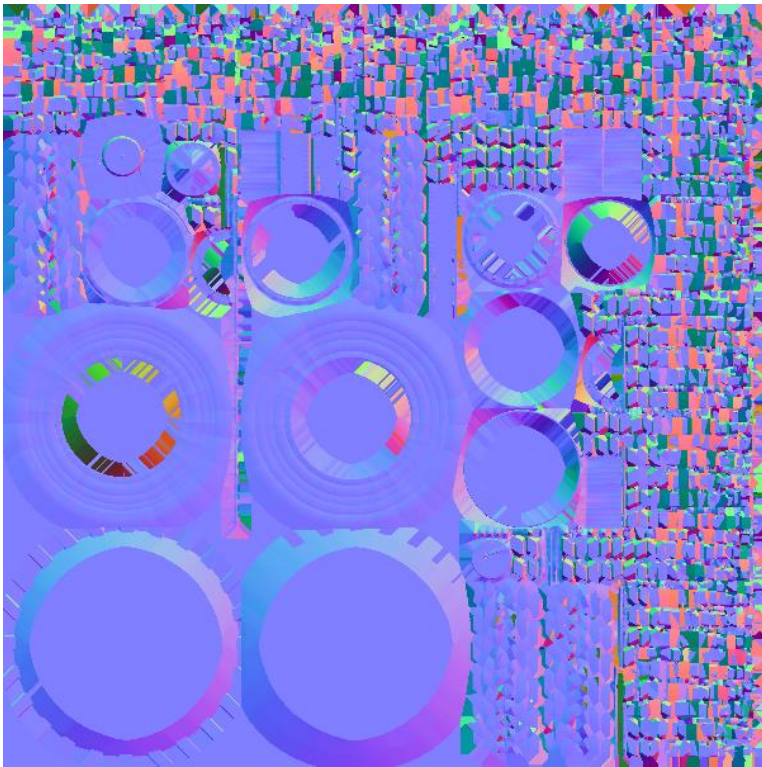
Kuvia Normal-mapeista



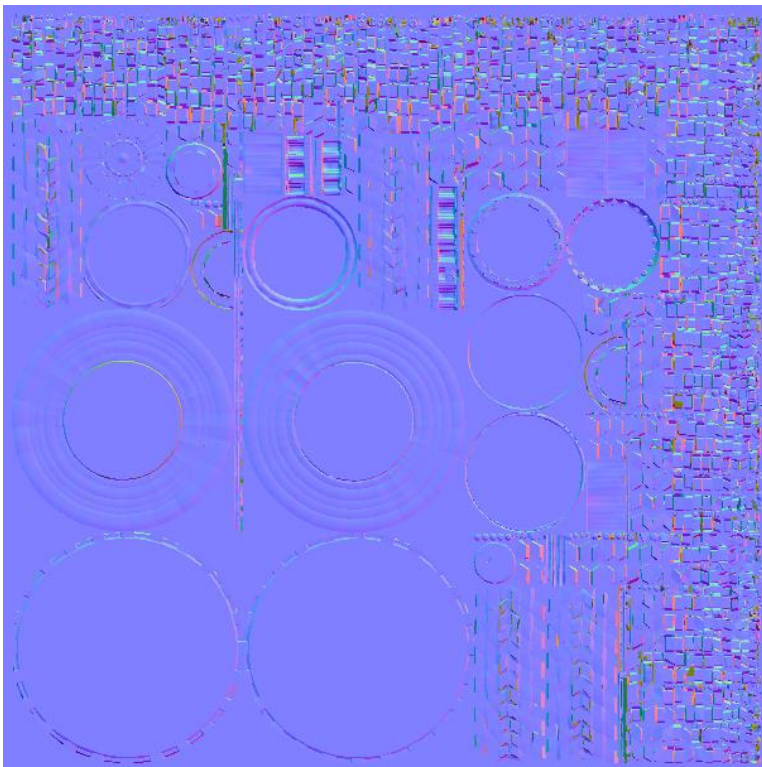
Kuva 12. Paras lopputulos xNormal ohjelmassa, .fbx -tiedostomuodossa oleva ja trianguloitu mesh ilman häkkiä.



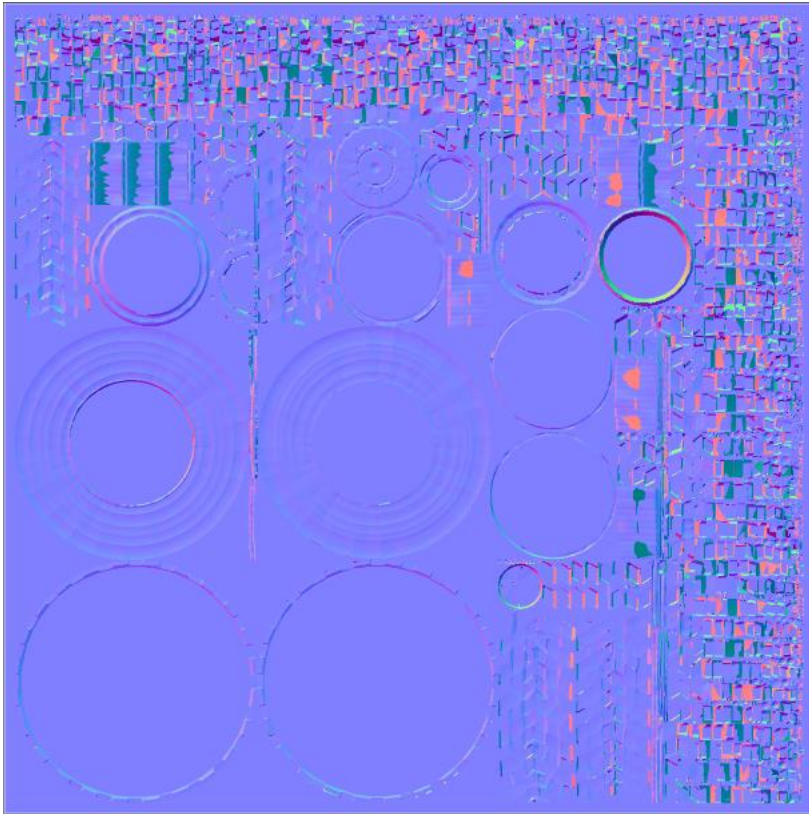
Kuva 11. xNormalin pehmenysasetus trianguloidussa meshissä, .fbx -tiedostomuodossa ilman häkkiä.



Kuva 13. Handplane 3D jälkeä, trianguloidun mesh häkin kanssa. Huomaa laajat tummanvihreät ja kirkkaan oranssit alueet. Häkin säteet tallensivat pinnanmuodot eri tavalla kuin häkitön versio



Kuva 14. Blenderillä trianguloidusta meshistä tehty Normal-map ilman häkkiä.



Kuva 15. Kuva 15 3DS Maxin Projection Modifierillä eli automaattisella häkillä tehty Normal-map