

Tomi Karhu

3D-MALLIEN OPTIMOINTI PELIMOOTTORILLE

Opinnäytetyö
Tradenomi

2020



**Kaakkois-Suomen
ammattikorkeakoulu**

Tekijä/Tekijät	Tutkinto	Aika
Tomi Karhu	Tradenomi (AMK)	Toukokuu 2020
Opinnäytetyön nimi		37 sivua
3D-mallien optimointi pelimoottorille		
Toimeksiantaja		
Kaakkois-Suomen ammattikorkeakoulu / Virrake-hanke		
Ohjaaja		
Jukka Selin		
Tiivistelmä		
<p>Opinnäytetyön tarkoituksena oli tutkia erilaisia keinoja 3D-mallien optimointiin. Työvälineinä käytettiin Unity, Unreal Engine, 3ds Max, SketchUp ja SketchUp Pro -ohjelmia. 3D-mallit mallinnettiin 3ds Max ja SketchUp/Pro -ohjelmilla. Mallit tuotiin pelimaailmaan Unity- ja Unreal Engine-pelimoottoriohjelmiin. Malleista tehtiin eri tiedostoja, joilla pystyi katsomaan niiden toimivuutta ja ulkonäöllistä vaikutusta. Samalla käytiin läpi mallien virheitä ja keinoja, joilla korjata niitä.</p> <p>Mallina toimi korealainen temppeli, joka mallinnettiin työtä varten. Mallintamisessa käytettiin kuvaa temppelistä ja mallista tehtiin kuvaa vastaava malli. Temppelit olivat samantyyppisiä, mutta 3ds Max -ohjelman malli oli low polygon -tyylinen ja SketchUp-ohjelman mallit olivat high polygon tyylisiä.</p> <p>Unityn ja Unreal Enginen pelimaailmat olivat peruspohjalla tehtyjä, ilman ylimääräisiä elementtejä. Pelillistämisen yhteydessä pelimoottoreissa malleihin tuotiin omat värit sekä materiaalit, ja mallit asetettiin paikoilleen. Kaikki mallin eri versiot tuotiin samaan pelimaailmaan, jotta voitiin nähdä niiden erot.</p> <p>Työssä esitellään myös ohjelmia, jotka tukevat Unreal Datasmith -työkalua, joilla voi mallintaa malleja ja tuoda Unreal Engine-ohjelmaan. Samalla esitellään mallinnusohjelmia, joita monet yritykset käyttävät mallinnuksessa.</p>		
Asiasanat		
3D-malli, 3D-mallinnus, optimoiminen, pelillistäminen		

Author (authors)	Degree	Time
Tomi Karhu	Bachelor of Business Administration	May 2020
Thesis title		37 pages
Optimizing 3D-models to a game engine		
Commissioned by		
South-Eastern Finland University of Applied Sciences / Virrake project		
Supervisor		
Jukka Selin		
Abstract		
<p>The purpose of the thesis was to study the best way to optimize 3D-models. The tools were Unity, Unreal, 3ds Max, SketchUp and SketchUp Pro. 3D-models were modeled with 3ds Max and SketchUp/Pro programs. The models were brought to the game world Unity as well as to Unreal, Different versions of the models were made to see how different techniques and how they affective the outlook. The thesis also studied the most typical mistakes of the models and how to fix them.</p> <p>A Korean temple served as a model, which was modeled for the work. A picture of the temple was used in modeling and the model was made to match the picture. The temples had both the same style, but 3ds Max program model was a low polygon and SketchUp models were high polygons.</p> <p>Unity and Unreal game worlds were made on a basic basis, without any extra elements. In the gaming programs, the models received their own colors and they were positioned. All the model files were imported into the same game world to see their differences.</p> <p>The work presented programs which supported Unreal Datasmith tool to model the models and to import them into the Unreal program. At the same time modeling programs used by many companies in modeling were introduced.</p>		
Keywords		
3D-model, 3D-modeling, optimizing, gamification		

SISÄLLYS

1	JOHDANTO.....	6
2	KOLMIULOTTEISEN MALLIN MALLINTAMINEN	6
2.1	Kolmiulotteisen mallinnuksen historiaa	7
2.2	Kolmiulotteinen mallinnus	8
2.3	Kolmiulotteisen mallin rakenne	9
2.4	Mallin pelillistämisen apuvälineitä	10
3	3D- MALLINNUSOHJELMIEN ESITTELY	11
4	OPTIMOINTI.....	13
4.1	Keinoja mallien optimoimiseen	14
4.2	Mallien optimointi ulkopuolisilla ohjelmilla	16
4.3	Ohjelmat jotka tukevat Datasmith-työvälinettä	18
5	MALLIEN PELILLISTÄMINEN	19
5.1	Mallit pelimaailmassa.....	19
5.2	Mallien tuonti Unity-pelimoottoriin	20
5.3	Mallien tuonti Unreal Engine-pelimoottoriin	24
5.4	Kolmiulotteisten mallien tiedostoformaatit.....	31
5.5	Yleisesti 3D-mallien käytöstä pelimoottoreissa	32
5.6	Työn vaiheet ja ongelmat.....	32
6	YHTEENVETO	34
	LÄHTEET.....	35

KUVALUETTELO

Kuva 1 Mallien rakenteet	9
Kuva 2 Datasmith ohjelman sivusto (Unreal Engine 2019)	10
Kuva 3 Microsoft harjoitusmalli (Microsoft 2019)	15
Kuva 4 Datasmithillä tuodun mallin muovinen olemus	16
Kuva 5 Ryhmitetty malli, jonka värit eivät näy oikein	17
Kuva 6 Mallin tuonti Unity-pelimoottoriin	20
Kuva 7 Mallin mahdollisia ongelmia	21
Kuva 8 Unityn materiaalikorjaus	21
Kuva 9 Mallin värikartta	22
Kuva 10 Mallin materiaalien haku Unityssä	22
Kuva 11 Mallin materiaalit Unityssä	23
Kuva 12 Mallin piilotetut elementit Unityssä	23
Kuva 13 Mallin piilotetut elementit Unrealissa	24
Kuva 14 Mallin tuonti Unreal Engine -pelimoottoriin	25
Kuva 15 Mallin luuranko Unrealissa	26
Kuva 16 Mallin materiaalit Unrealissa	27
Kuva 17 Mallin tuominen Datasmith lisäosan avulla	28
Kuva 18 Mallin tuontiasetukset Datasmith-lisäosassa	29
Kuva 19 Mallit tuotuna eri asetuksilla	30
Kuva 20 Mallin rakenne on tullut virheellisesnä Unreal Engine-pelimoottoriin	31
Kuva 21 Mallin virheiden sijainnit SketchUp-ohjelmassa	33

TAULUKKOLUETTELO

Taulukko 1 Datasmithin tukemat tiedostoformaatit ja ohjelmat (Unreal Engine 2019)	18
-----------------------------------------------------------------------------------------	----

1 JOHDANTO

Kolmiulotteisia malleja on nykypäivänä kaikkialla ja varsinkin peleissä, jossa ne luovat oman maailman sekä antavat sisältöä peliin. Kolmiulotteisessa mallinnuksessa tärkeintä on mallin laatu, tekstuurien asettelu ja verteksien määrä. Kolmiulotteinen malli voidaan tehdä monella tavalla.

Tietokoneille on saatavana erilaisia mallinnusohjelmia ja mallintajan työvälineinä on näppäimistöt ja hiiri, sekä esimerkiksi joillakin pelifirmoilla on elektroniset kynät, joilla voidaan piirtää malleja. Kolmiulotteisia malleja pystytään esittämään myös kaksiulotteisesti, valokuvina ja tulostamaan 3D-tulostimella.

Tässä työssä tutkittiin keinoja, joilla voidaan optimoida 3D-malleja pelialustalle. Lisäksi tutkittiin työvälineitä, joita voidaan hyödyntää mallien optimoimiseen ja joiden avulla saadaan luotua laadukkaita materiaaleja. Pelillistämiseen käytettiin Unreal Engine- ja Unity-pelimoottoreita. Mallintamiseen käytettiin SketchUp-, SketchUp pro- ja 3ds Max -ohjelmia.

Työssä tehtiin 3d-malleja ja muutettiin niitä eri tiedostoformaatteihin, joiden avulla pystyttiin tutkimaan eri formaattien vaikutusta mallien optimointiin ja pelillistämiseen. Mallina toimi temppele, jossa oli paljon polygoneja ja värikarttoja. Mallit tuotiin Unity- ja Unreal Engine-ohjelmiin. Malleja optimoitiin eri työkaluilla ja myös pelillistämiseen käytettyjen pelimoottoreiden tarjoamilla työkaluilla.

2 KOLMIULOTTEISEN MALLIN MALLINTAMINEN

Työssä käytettyjä ohjelmia olivat SketchUp, SketchUp Pro, 3ds Max, Unity ja Unreal Engine. Ohjelmia käytettiin HP Omen kannettavalla tietokoneella. SketchUp- ja 3ds Max-ohjelmilla luotiin käytetyt kolmiulotteiset mallit ja Unity, sekä Unreal Engine olivat ne pelimoottorit, joihin mallit tuotiin.

Kolmiulotteisena mallina oli temppele, joka oli mallinnettu kahdella eri mallinnusohjelmalla. Mallit olivat muuten samantyyllisiä, mutta 3ds max-ohjelmalla tehdyssä versiossa mallin katto oli mallinnettu pienellä polygonimäärällä, jotta pystyi vertailemaan, vaikuttaako mallien polygonit ja eri ohjelmien käyttö malleihin sekä tuoko se jotain ongelmia tai parannuksia

pelillistämiseen. Unreal Engine-pelimootorissa oli apuvälineinä Exporter Plugins ja Unreal Datasmith, joiden avulla mallit sai helposti tuotua Unreal Engine-ohjelmaan siten, että ohjelma ymmärsi myös paremmin mallien rakenteita ja tekstuureja.

2.1 Kolmiulotteisen mallinnuksen historiaa

Tietokonepohjainen mallintaminen alkoi 1960-luvulla ja markkinoille saapuivat ensimmäiset kaupalliset CAD-piirtämiseen soveltuvat tietokoneet ja ohjelmat. Silloin syntyi ensimmäinen virtuaalinen malli. Mallin loi William Fetter, joka suunnitteli 3D-mallin Boeing yritykselle matkustajalentokoneen ohjaamoa varten. Mallina oli ihminen, joka sovitettiin Boeing-lentokoneen ohjaamoa varten. Kolmiulotteinen mallintaminen kehittyi 1980-luvulla, jolloin CAD-suunnittelua kehitettiin ja ensimmäinen varsinainen 3D-malli tietokoneelle luotiin 1985. 3D-mallinnusohjelmat yleistyivät 1990-luvulla ja siitä asti mallinnusohjelmia on luotu lisää ja ohjelmat kehittyvät nopeasti tietotekniikan kanssa. Aikoinaan 3D- malleja piti tehdä tarkasti, koska oli tärkeää pitää polygonien määrä alhaisena, jotta muistitila säästyisi ja peli pyörisi sujuvasti ilman ongelmia. (Kontturi 2016, 11–14.)

Nykyään 3D-malleja käytetään paljon havainnollistamaan mm. taideteoksia, rakennuksia, rakennusten osia ja erilaisia laitteita. Arkkitehdit käyttävät mallintamisen ohjelmia luodessaan mallia tulevasta rakennuksesta (Rajakyläteknö 2016, 7–9). Tietokonepeleissä virtuaalinen maailma rakentuu 3D-malleista ja pelien hahmoista, jotka ovat myös 3D-malleja. Pelimaailmassa mallit ovat jo melko realistisia ja laadukkaita. Tietokoneet ovat tehokkaita pyörittämään pelimaailmaa ja malleja, jolloin saadaan luotua realistisempia pelimaailmoja ja malleja. Elokuviissa käytetään paljon 3D-malleja ja animaatioita, joilla saadaan lisää ilmettä valkokankaalla.

Nykyään virtuaaliseen maailmaan pääsee VR-laseilla ja käyttäjä saa kokea pelimaailman omin silmin. Käyttäjä näkee virtuaalisessa maailmassa 3D-malleja läheltä ja pystyy pyörittelemään niitä ”kädestä” pitäen. Tietotekniikka kehittyi jatkuvasti ja pelimaailmasta luodaan entistä enemmän realistisempaa, mutta samalla se on ”raskaampaa” tietokoneille tai konsoleille sekä muille oheislaitteille, jolloin laitteiden pitää olla suorituskyvyiltään tehokkaita.

2.2 Kolmiulotteinen mallinnus

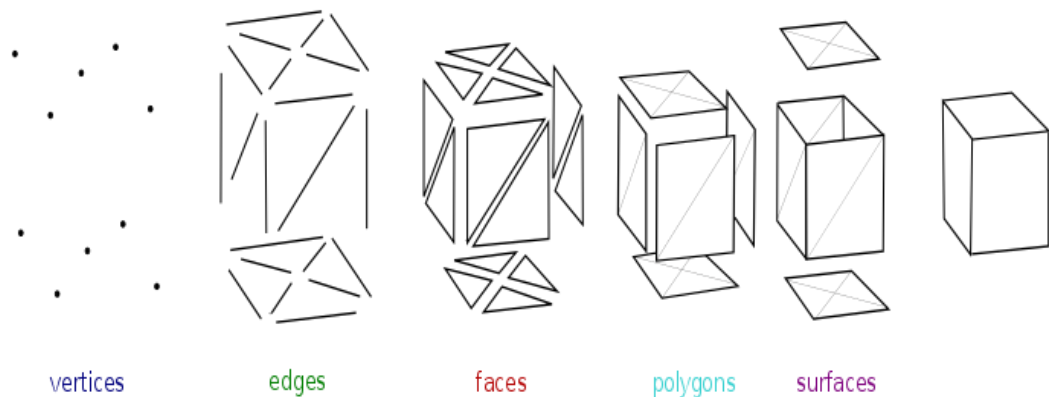
Kolmiulotteinen mallintaminen on tietokonetta käyttäen kolmiulotteisen mallin luomista halutusta kohteesta. Mallinnus alkaa mallin hahmottamisesta eli mitä halutaan mallintaa. Usein tehdään ”pohjapiirros” mallista ja tuodaan se mallinnusohjelmaan (Rajakylätekno 2016, 7–9). Mallien teko aloitetaan luomalla viivoja tai muotoja, jotka liittymällä yhteen rakentavat mallin selkärangan. Näistä sitten muodostuu itse malli. Lopuksi mallille annetaan värimaailma, joka antaa mallille ilmeen. Myös varjostus antaa mallille eloa ja näyttää aidommalta pelimoottorissa tai kuvassa. Malleille voidaan antaa valoja, jotka tuovat lisää visuaarisuutta ja antavat kohteelle lämpöä. Mallit pyritään tekemään mahdollisimman samanlaisiksi kohteen piirustusten kanssa.

Mallien verteksien määrää kannattaa seurata, koska liian ”täydellinen” malli syö tietokoneen tehoa ja voi hidastaa koneen toimivuutta. Tekstuuri- ja polygonien suuri määrä voi vaatia tietokoneelta paljon tehoa ja pahimmillaan kaataa tietokoneen (Lanusse 2015). Peliin tuotu malli voi olla liian raskas. Tämä vaatii pelimoottorin resursseja liikaa ja hidastaa pelin toimivuutta. Liian ”täydellinen” malli kannattaa muokata yksinkertaisemmaksi tai vähentää verteksien määrää. (Unity Documentation 2019).

Myös tekstuureja kannattaa muokata tai pienentää niiden määriä. Poistamalla kaikki ei vaadittavat tiedot 3D-mallista, vähentämällä kolmioiden lukumäärät ja yksinkertaistamalla mallin hierarkiaa saadaan tehostettua mallin suorittamista peleissä (Lohikoski. Rudén 2013, 8). Mallin voi myös muokata näyttämään hyvälle läheltä katsottuna ja kaukaa katsottuna näyttämään yksinkertaiselta elementiltä hyödyntäen LOD (mallin rakenteiden tarkkuus) ominaisuuksia (Valve Developer community 2013). LOD-tekniikassa mallista on muutamia eri versioita, jotka ovat tarkkuudeltaan erilaisia. Tämä vähentää pelimoottorin tehontarvetta. (Game Dev Academy 2015).

2.3 Kolmiulotteisen mallin rakenne

Kolmiulotteinen malli on kohteiden matemaattinen representaatiopinta. Kolmiulotteisen mallin mitattavia ulottuvuuksia ovat leveys X, korkeus Y ja syvyys Z. Ne auttavat mallintajaa luomaan kuvan mallista ja havainnollistamaan sen muodot. Mallit koostuvat pisteistä, reunoista kolmioista, monikulmioista ja pinnoista. (Gamedesigning 2019). Kuvassa 1 havainnollistetaan kolmiulotteisen mallin rakenne ja kuinka tietokone rakentaa mallin, joka luodaan mallinnusohjelmassa.

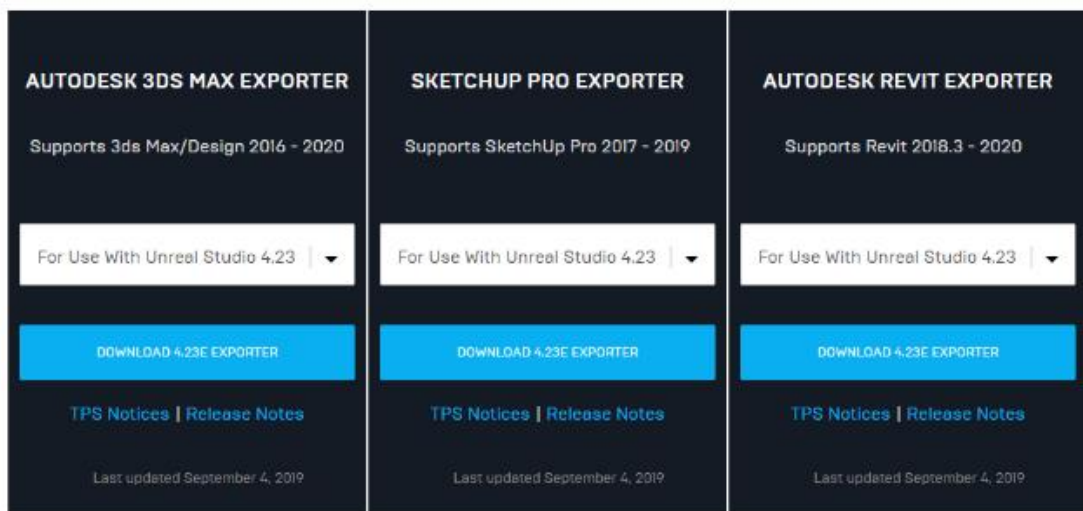
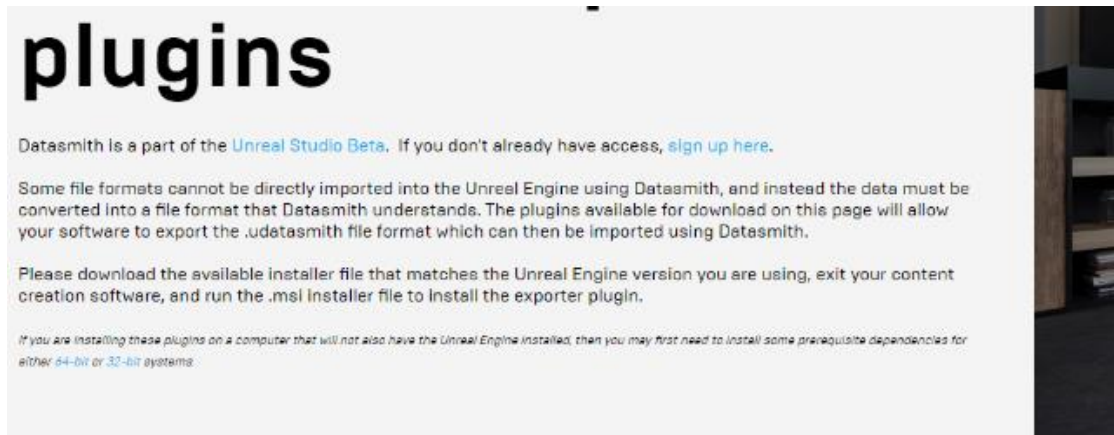


Kuva 1 Mallien rakenteet

Mallien perusta ovat verteksit (vertices), jotka määrittävät mallin rakenteen. Reunukset (edges) antaa mallille reunaviivat ja kuvastaa mallin ulkoista olemusta. Pinnat (Faces) täydentää mallin kappaleet eli antaa pisteille ja reunoille kolmiomuodon ja yhdistää reunat keskenään. Monikulmiot (Polygons) yhdistää irtaantuneet kolmiopinnat ja antaa mallille täydentävät pinnat. Esimerkiksi kaksi kolmiota, jotka yhdistyvät keskenään luovat neliön. Pinnat (surfaces) rakentaa mallia valmiiksi yhdistäen kolmiulotteisen mallin rakenteet keskenään.

2.4 Mallin pelillistämisen apuvälineitä

Unreal Engine-ohjelmaan saa Datasmith-apuvälineen, joka avaa mallit laadukkaammin ja rakentaa mallien tekstuurit. Kuva 2 esittää, kuinka 3ds Max- ja Sketchup-ohjelmaan pystyy lataamaan Datasmith-lisäosan, joka tukee Unreal Engine-ohjelmaa.



Kuva 2 Datasmith ohjelman sivusto (Unreal Engine 2019)

Unreal Engine Datasmith on ilmainen ja asentuu automaattisesti valittuun mallinnusohjelmaan. Unreal Engine-ohjelmaan voidaan ladata Unreal Engine Datasmith -apuohjelma, joka tukee Datasmith-malleja ja luo malleille mm. paremman näköiset tekstuurit ja tekee niistä muutenkin realistisemmän näköisiä. (Unreal Engine 2019)

3 3D- MALLINNUSOHJELMIEN ESITTELY

Tässä luvussa esitellään 3D-mallinnusohjelmia, niiden toimivuutta ja mitä niillä voi tehdä. Monet mallinnusohjelmat soveltuvat myös hyvin pelillistämisen tarpeisiin.

Autodesk 3ds Max

Autodesk 3ds Max on tarkka mallintamisohjelma, jolla voi tehdä kätevästi pieniä malleja esim. hahmoja ja esineitä. Ohjelmalla voi tehdä myös isompia malleja ja se tukee hyvin tuontia pelimoottoreihin. 3ds Max on suunniteltu mallien luontiin sekä luomaan yksityiskohtaisia malleja peleihin. 3ds Max vaatii Windows-käyttöjärjestelmän toimiakseen. Ohjelmalla voi luoda laadukkaita animaatioita, ja se sisältää paljon apuvälineitä mallien ja animaatioiden tekoon. 3ds Max ohjelma on hyvin suosittu monissa töissä. Ohjelmaa on käytetty myös elokuvissa ja sarjoissa. Ohjelma on kallis, mutta opiskelijoille tarjotaan kolmen vuoden ilmaista lisenssiä. (Autodesk 3ds Max 2019).

AC3D

AC3D-ohjelmaa käytetään mm. peleihin ja se avulla voidaan tehdä esimerkiksi 3D-tulostamisen malleja. Ohjelmaa voi käyttää myös Google Earth, X-Plane ja Second life-ohjelmiin. Ohjelma on hyvin helppokäyttöinen ja sisältää hyvät UV-ominaisuudet. Tämä lisää käyttömahdollisuuksia ja parantaa mallin ulkonäköä. Ohjelmaa voi käyttää 14 päivää ilmaiseksi ja hinta riippuu käyttäjämäärästä. Ohjelma toimii Windows-, Mac- ja Linux-käyttöjärjestelmissä. (AC3D 2019).

Cinema 4D

Cinema 4D -ohjelma on suunniteltu 3D mallien ja animaatioiden tekemiseen. Helppokäyttöinen, hyvää teknologiaa ja nopeasti sekä vakaasti toimiva ohjelma. Ohjelman on kehittänyt MAXON Computer GmbH Friedrichsdorffin. Ohjelmalla voidaan luoda melko helposti kolmiulotteisia visualisointeja. Yleisesti ohjelmaa käytetään arkkitehtisuunnitteluun, visualisointiin, simulointiin ja animaatioiden tekemiseen. Ohjelmaa voi käyttää Windows- ja

MacOS -tietokoneilla. Ohjelma on maksullinen, mutta ohjelmasta on saatavilla myös kokeiluversio. (Maxon 2019).

Houdini

Houdini on tunnettu ohjelma isoissa peliyhtiöissä. Mallintaminen on laadukasta ja ohjelmalla onnistuu myös animaatioiden luominen. Esimerkiksi Disney käyttää animaatioelokuvissaan Houdini-ohjelmaa. Houdini on avoin ja lähdekoodia voi muokata halunsa mukaan ja siihen voi lisätä tarvittavia lisäpaketteja. Houdini on maksullinen ohjelma, mutta siitä löytyy ilmainen versio opiskelijoille, artisteille ja harrastelijoille. Ohjelma toimii kaikilla tietokonekäyttöjärjestelmillä. (SideFX 2019).

Lightwave

Lightwave-ohjelma on mallinnusohjelma, jolla voi tehdä animaatioita ohjelmiin/elokuviin ja 3D-malleja peleihin sekä paljon muuta. Lightwave on avoin ja lähdekoodia voi muokata sekä asentaa lisäpaketteja. Lightwave on tunnettu ohjelma monissa käyttäjäpiireissä. Ohjelmalla on luotu paljon malleja ja animaatioita elokuviin, sarjoihin, animaatioihin ja peleihin. Ohjelma on maksullinen, mutta sitä voi käyttää kuukauden ilmaiseksi. Ohjelma pyörii Windows- ja Mac- tietokoneilla. (Toolfarm 2019).

Substance Maya

Monet yhtiöt käyttävät Substance Mayaa ja monissa uusimmissa peleissä on käytetty ohjelman mallintamisen välineitä ja animoimista. Isot yhtiöt käyttävät Substance Mayaa, koska sillä saa tehtyä laadukkaita malleja peleihin. Substance Mayassa on suurivalikoima värejä ja materiaaleja, joita voi sekoitella keskenään. Ohjelmaa käytetään myös tuotteiden esittelyyn ja niiden havainnollistamiseen. Ohjelma sopii myös arkkitehdeille, koska sillä saa realistisia oikean näköisiä rakennusmalleja. Ohjelma on maksullinen ja soveltuu kaikille tietokonekäyttöjärjestelmille. Ohjelmasta on tarjolla ilmainen kokeiluversio sekä opiskelijaversio kouluille. (Adobe 2019).

Trimble SketchUp

SketchUp on helppokäyttöinen mallintamisohjelma. SketchUp-ohjelmalla on Unityn kanssa yhteistyösopimus, joka mahdollistaa mallien tuonnin Unity-ohjelmaan helpommin ja suoraan. Yleisimpiä malleja, joita tehdään SketchUpilla ovat esimerkiksi rakennukset ja ajoneuvot. Ohjelmalla voi luoda muitakin malleja, mutta se vaatii enemmän aikaa ja tarkkuutta. Ohjelman sisällä on varasto, johon voi ladata omia malleja muiden käyttöön tai sieltä voi ladata muiden tekemiä malleja itselleen. Ohjelmaa voi käyttää kolmekymmentä vuorokautta ilmaiseksi, jonka jälkeen käyttö maksaa. Ohjelmaa voi käyttää Windows- ja Mac-tietokoneilla.

Muita mallinnusohjelmia

Nykyään on kymmeniä eri mallinnusohjelmia, joita voi käyttää 3D-mallien luontiin tai animoimiseen. Monissa mallintamisen ohjelmissa on omat yhteensopivuudet pelimoottoreiden kanssa, jotka helpottavat mallien tuontia. 3D-mallinnusohjelmissa on paljon eroja. Jotkut ohjelmat ovat tarkempia ja toiset ohjelmat taas ovat keskittyneet ohjelman animaatioiden toimivuuteen ja helppouteen. Jotkut ohjelmat mahdollistavat lähdekoodin muokkauksen, silloin voi muokata ohjelmaa mielensä mukaan. Joihinkin mallinnusohjelmiin on saatavana valmiina lisäosia (blugins), joita voi käyttää laajentamaan ohjelman ominaisuuksia. Niitä voi ladata ja asentaa ohjelmaan ilmaiseksi tai pientä maksua vastaan. Yleensä lisäosat ovat ilmaisia tai vaativat kirjautumisen tarjoajan sivustolle.

4 OPTIMOINTI

Tässä luvussa käsitellään, kuinka optimoida malleja ja mitä keinoja on saada malli pysymään halutun näköisenä vielä optimoinnin jälkeen. Samalla selvitetään mitä apuvälineitä ja erillisiä ohjelmia on olemassa mallien optimointiin.

Luvussa esitetään myös taulukko, jossa esitellään Unreal Engine Datasmith-työkalun tukemia tiedostomuotoja.

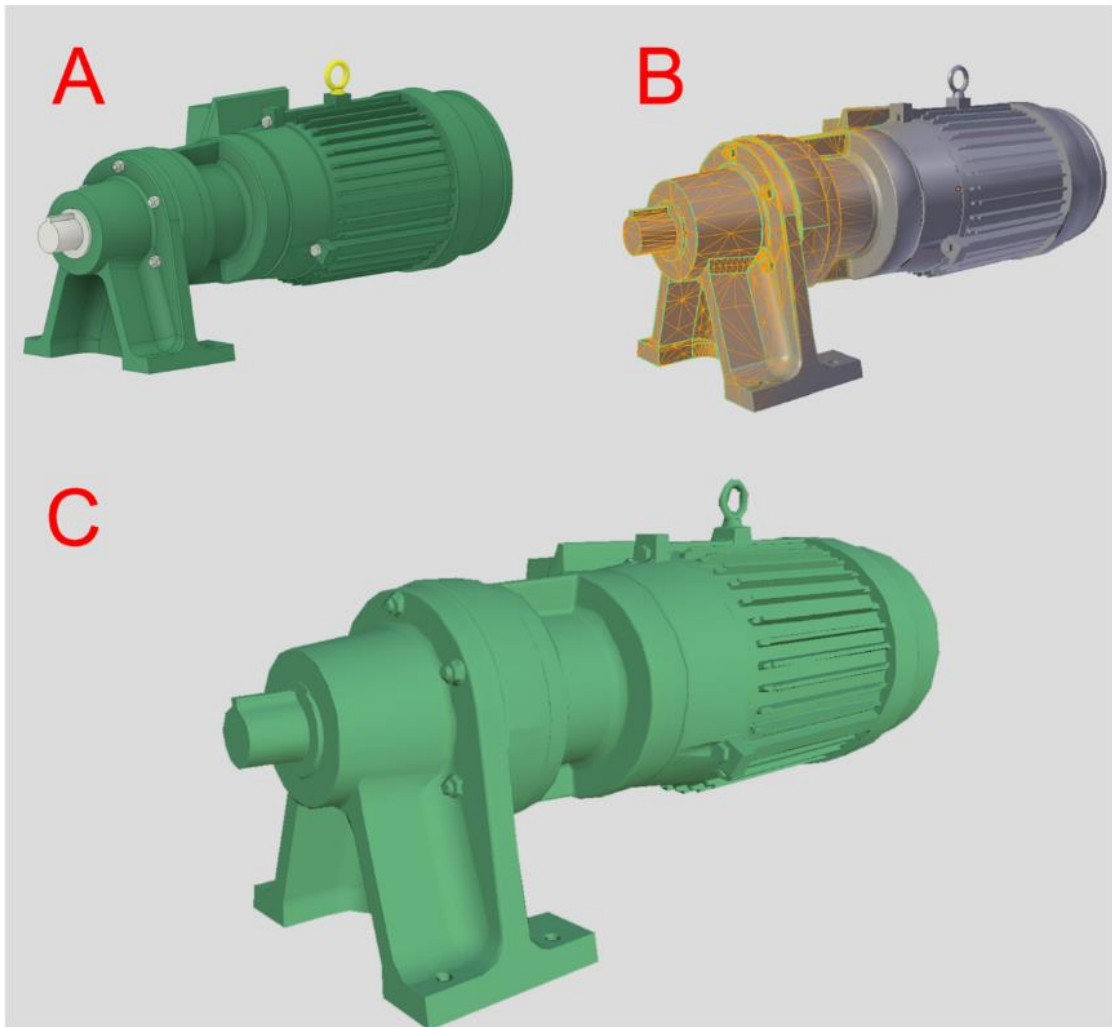
4.1 Keinoja mallien optimoimiseen

3D-malleissa kannattaa käyttää (PBR) renderointia. PBR on fysiikkaperusteista renderointia, jolla tarkoitetaan realistisempaa grafiikkaa ilman laskentaa vaativaa säteenseurantaa. Säteenseuranta tarkoittaa tietokoneen luomaa kuvaa ja sen valosäteiden kulkua. PBR tuo esiin mallin väreille, epätasaisuudelle ja kuoppaisuudelle paremman näkyvyyden. (Microsoft Docs 2019).

PBR:llä luodaan tekstuurikarttoja ja ne korvaavat suuren määrän erilaisia materiaaleja ja pintoja. Kun mallia käsitellään, eikä siihen tarvita tarkkoja yksityiskohtia, voidaan suorituskykyä parantaa käyttämällä yhtä väriä, jolloin ylimääräiset tekstuuritiedot ja pintainformaatiot jäävät pois. Mallin pinnan heijastusta eli karheutta pienentämällä voidaan myös parantaa mallin suorituskykyä. Pienentämällä mallin tarkkuutta, voidaan vaikuttaa muistin kulutukseen ja tiedoston kokoon. JPG-pakkaus ja PNG-tiedostomuoto pienentävät tiedostokokoa, mutta eivät vaikuta suoritusmuistiin. (Microsoft Docs 2019).

Suuri kolmio- tai kärkipiste-lukumäärä voi pienentää suorituskykyä. Jos tiedetään mallin käyttötarkoitus etukäteen, niin pisteiden määrää voidaan tarvittaessa pienentää. Suurempaa pistetiheyttä voidaan käyttää mallin tärkeimmillä alueilla tai tärkeimmissä yksityiskohdissa.

Hienoja geometrisiä yksityiskohtia ja materiaalivärejä voidaan korvata normaali-, väri- ja ORM-kartoilla, jolloin säästetään kolmioiden määrässä ja tiedoton koossa. Kuvassa 3 havainnollistetaan mallien rakenteita ja erilaisia esitystapoja. (Microsoft Docs 2019).



Kuva 3 Microsoft harjoitusmalli (Microsoft 2019)

Yhdistämällä materiaaleja tai vähentämällä materiaalikutsuja ja kehyksien määrää, voidaan nopeuttaa mallin toimintaan. Hitaat latausajat voivat johtua suuresta määrästä erillään olevia materiaaleja. Atlas-tiedosto (pinnan konfigurointitiedosto) voi nopeuttaa materiaalikutsuja, mikä tehostaa mallin toimivuutta ja latausaikaa. Mallin hierarkian monimutkaisuudet eli ryhmittelyt, ja esimerkiksi suuret määrät solmuja ja liitoksia vaikuttavat mallin suoritus aikaan heikentävästi. Suoritus aika voi parantaa yksinkertaistamalla mallin hierarkioita, poistamalla tarpeettomia solmuja ja yhdistämällä verkon silmiä mahdollisuuksien mukaan. (Microsoft Docs 2019).

Mallin vilkkuminen tapahtuu, kun geometriset pinnat ovat samassa tasossa tai lähes samassa tasossa. Se on yleinen ongelma varsinkin, jos malli on animoitu tai liikkuu näytöllä paikasta toiseen. Tämä ongelma johtuu Z-puskuroinnista, kun kaksi pintaa on lähes samalla etäisyydellä kamerasta.

Vilkkuvuusongelmat saadaan ratkaistua geometrinen pintojen etäisyyden kasvatuksella pienellä määrällä.

Githubissa on työväline, joka parantaa mallien toimivuutta pelialustalla. Ohjelma on automaattinen ja tekee korkeampia monikulmioverkkoja, mutta samalla tasaisia kolmioita. Kolmion asetteluja pystyy korjaamaan. Tälläkin ohjelmalla on rajansa, kun aina voi olla topologeja, joita tietokone ei pysty käsittelemään. Topologioiden määrään pystyy vaikuttamaan mallintaja, vähentämällä manuaalisesti topologioiden määrää. Se on aikaa vievää, mutta samalla tehokas tapa. Mallintaja pystyy päättämään mallinsa näytävyyden tekemällä itse korjaukset. Ohjelmaa kehitetään ja sen oletetaan olevan parempi kuin maksulliset ohjelmat. Ohjelma takaa 60-70%:n kevennyksen malleille ja saman laadun, minkä mallintaja pystyy toteuttamaan monien tuntien työllä. Ohjelma on ilmainen ja ladattavissa Githubissa. (Gamedev 2019).

4.2 Mallien optimointi ulkopuolisilla ohjelmilla

Ulkopuoliset ohjelmat pystyvät optimoimaan mallia, simuloimaan mallia ja sen toimintaa. Tämä helpottaa mallin tuontia pelimoottoreihin. (Daniel 2019). Ulkopuoliset ohjelmat voivat myös jossain tapauksissa aiheuttaa vahinkoa malleille. Esimerkiksi Unreal Datasmith-ohjelma tekee SketchUp-malleista muovisen näköisiä ja jättää osan mallin tekstuureista pois. Kuvassa 4 näkyy muovisen näköinen lopputulos.



Kuva 4 Datasmithillä tuodun mallin muovinen olemus

Malliin voi tehdä korjauksia tuomalla tekstuurit manuaalisesti, mutta tällöinkin niiden laatu yleensä kärsii ja materiaalit voivat muuttua liian isoiksi. SketchUp-malleja ei myöskään kannata ryhmittää (group) omassa ohjelmassa, koska ryhmät pysyvät Unreal-ohjelmassa ja ryhmiteltyjä rakenteita on hankalampi korjata. Kuvassa 5 näkyy ryhmittelyn hankaluus, jolloin värit voivat mennä mallissa väärin.



Kuva 5 Ryhmitetty malli, jonka värit eivät näy oikein

Nykyisin löytyy myös erilaisia ohjelmia, jotka optimoivat 3D-malleja. Ohjelmat voivat olla muokattavissa, eli käyttäjä pystyy ohjelmoimaan ohjelmaa ja tekemään siihen haluttuja muutoksia. Muutokset voivat olla ohjelman suoritukseen tai sen toimivuuteen vaikuttavia asioita. Ohjelmat pystyvät siivoamaan malleja ylimääräisistä kolmioista, muokkamaan niitä ja parantamaan niiden toimivuutta. Ohjelmat pystyvät myös korjaamaan mallin virheitä ja elävöittämään mallin rakennetta ja värillisyyttä. (Lohikoski, Rudén 2013, 24).

Meshlab-ohjelmalla mallintaja pystyy optimoimaan mallia ja muokkaamaan mallin skaalausta, mittakavaa, sijaintia sekä suuntaa. Mallin pystyy esimerkiksi keskittämään uudelleen, jolloin se sopii paremmin valmiiseen käyttöön. Ohjelma on monipuolinen ja antaa käyttäjälle paljon erilaisia työvälineitä, joilla käyttäjä pystyy parantamaan mallien toimintaa ja ulkoasua. (Meshlab 2019).

4.3 Ohjelmat jotka tukevat Datasmith-työvälinettä

Seuraavassa taulukossa 1 havainnollistetaan Datasmith-työkalun tukemia tiedostoformaatteja ja yhteensopivuutta eri ohjelmien kanssa.

Taulukko 1 Datasmithin tukemat tiedostoformaatit ja ohjelmat (Unreal Engine 2019)

Tiedosto	Tiedoston pääte	Versio	Työstämisen tyyppi	Värimaailma	Reunojen korjaus	Kokoominen
3D ACIS	.sat	R21 -	Alkuperäinen	Ei	Kyllä	Kyllä
3DSXML	.3dxml	FS 2.0	Alkuperäinen	Ei	Ei	Ei
Autodesk 3DS Max	.uDatasmith	2016-2019	Tuotu apuohjelma	Kyllä	Ei	Kyllä
Autodesk Inventor	.iam, .ipt	2018-	Alkuperäinen	Ei	Kyllä	Ei
CADDS	.gbf	4 & 5	Alkuperäinen	Ei	Kyllä	Kyllä
CATIA V5	.catpart, .catproduct, .cgr	R10 – R24	Alkuperäinen	Ei	Kyllä	Kyllä
ESRI CityEngine	.uDatasmith	--	Tuotu apuohjelma	-	-	-
I-DEAS	.unv	NX6 -	Alkuperäinen	Ei	Kyllä	Kyllä
Intial Graphics Exchange Specification (IGES)	.iges, .igs	5.2, 5.3	Alkuperäinen	Ei	Kyllä	Kyllä
JT Open	.jt	10 -	Alkuperäinen	Kyllä	Kyllä	Kyllä
MDL	Mdl	2018	Alkuperäinen	Ei	Ei	Ei
Parasolid (x_t)	.x_t	20 -	Alkuperäinen	Ei	Kyllä	Ei
PTC Creo (Pro/ENGINEER)	.asm.1, .neu, .prt.1	V13 – Creo 2.0	Alkuperäinen	Kyllä	Kyllä	Kyllä
Rhino 3D	.3dm	--	Alkuperäinen	Ei	Kyllä	Ei
ROBCAD	.rf	--	Alkuperäinen	Kyllä	Kyllä	Kyllä
Siemens NX	.prt	V11 – NX9	Alkuperäinen	Ei	Kyllä	Kyllä
SketchUp	.uDatasmith	2017, 2018	Tuotu apuohjelma	Kyllä	Ei	Kyllä
SolidWorks	.sldasm, .sldprt	2017 -	Alkuperäinen	Ei	Kyllä	Kyllä
STEP	.step	AP203, AP214, AP242	Alkuperäinen	Kyllä	Kyllä	Kyllä

DeltaGen	Fbx	2017, 2018	Alkuperäinen	Kyllä	Ei	Kyllä
VRED	Fbx	2019	Alkuperäinen	Kyllä	Ei	Kyllä
AxF	Axf	2018	Alkuperäinen	Ei	Ei	Ei

Taulukon 1 ensimmäinen sarake esittää tuetun ohjelman nimen. Toinen sarake esittää ohjelman tiedostotyyppiä. Kolmas sarake esittelee ne ohjelmien versiot, joita Datasmith tukee. Neljännessä sarakkeessa näkyy tapa, jolla ohjelma toimii mallintamisen ohjelmassa. Viides sarake kertoo, sisältääkö formaatti värikarttoja ja viimeiset sarakkeet kertovat, pystyykö ohjelma korjaamaan mallin konflikteja ja pakkaako ohjelma mallin.

5 MALLIEN PELILLISTÄMINEN

Tässä luvussa esitellään, kuinka malleja pelillistetään ja millä tavalla niitä tuodaan pelillistämishjelmiin. Tässä yhteydessä esitellään Unity- ja Unreal Engine-pelimoottoreiden käyttö pelillistämisessä, niiden mahdollisuudet ja niiden toiminnot mallien tuomiseen. Lisäksi esitellään niitä ongelmia, joita ilmeni tekemissäni esimerkeissä.

5.1 Mallit pelimaailmassa

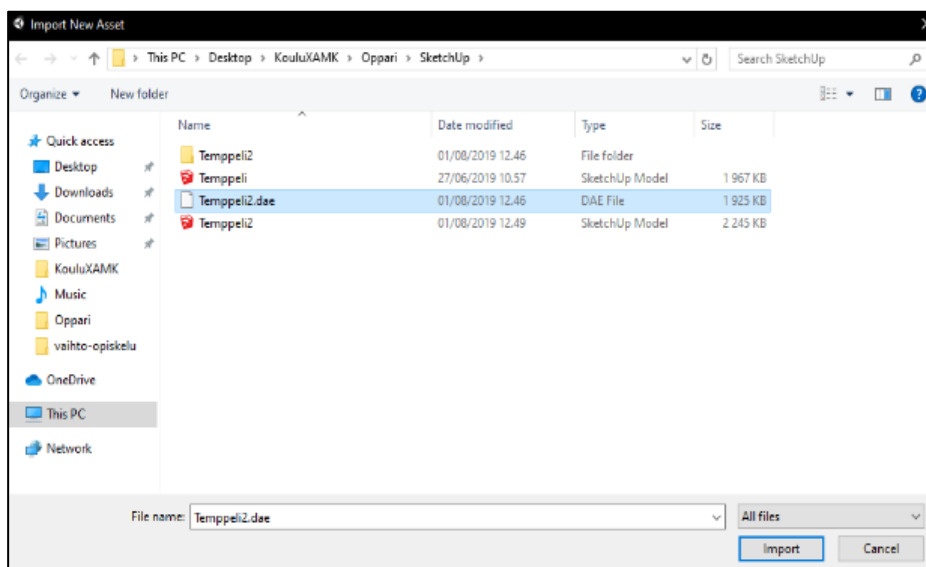
Pelimaailmassa mallit näkyvät sen mukaisesti, mitä pelaajan halutaan näkevän ja mitä pelimaailmaan on sijoitettu. Oleellisia kohtia kannattaa tehdä näyttäväksi ja korostaa niiden näkyvyyttä. Malleja tuotaessa pelimaailmaan, kannattaa miettiä, mitä pelajaa tulee näkemään ja mitkä alueet on rajattu pelaajalta. (Lohikoski. Rudén 2013, 8.) Näillä keinoilla pystyy suunnittelemaan mallia ja toteuttamaan laadukkaasti mallista erityisesti ne alueet, jotka näkyvät pelaajalle pelimaailmassa. Samalla pystytään minimoimaan mallin tiedoston kokoa, joka vähentää pelin kokoa ja säästää koneen tehoa ja muistia. Eli kaikkea ei tarvitse mallista tehdä tarkasti valmiiksi, jos siitä näkyy pelkästään rajattu alue.

Jos esimerkiksi mallinnetaan rakennus, josta näkyy vain rakennuksen edusta, ja takaseinä sekä osa sivuseinästä ei näy pelaajalle, voidaan mallintaa vain etuseinä ja osa sivuseinästä pelimaailmaan. Tällöin voidaan säästää koneen

tehoja ja muistia jättämällä tarpeettomat osat pois. Mallista puuttuvia osia voidaan peittää toisilla malleilla tai koristeilla, jolloin voidaan hämätä pelaajaa.

5.2 Mallien tuonti Unity-pelimoottoriin

Malleja voidaan tuoda kahdella eri tavalla Unity-pelimoottoriin. Ensimmäinen keino on ”raahata” malli suoraan peliin ja toinen keino on valita pelimoottorista tuontivalikko, etsiä malli koneelta ja valita mallin paketti, kuten kuvassa 6 näytetään.



Kuva 6 Mallin tuonti Unity-pelimoottoriin

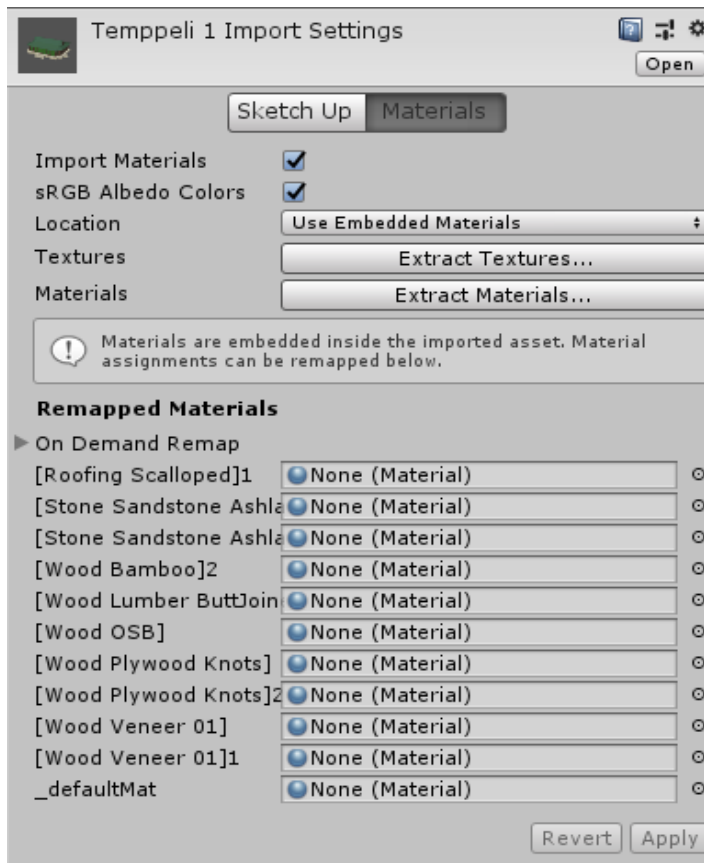
Malli menee suoraan Unityn ”asset” kansioon, josta pelimoottori hakee halutun mallin käyttöön. Mallien tiedot vaihtelevat ja mallin tiedostotyyppi vaikuttaa mallin materiaaliin sekä siihen pitääkö mallin asetuksia muuttaa pelimoottorissa.

Unity-ohjelmassa SketchUp-malleissa tiedostotyyppi vaikuttaa materiaalien sisältöön ja siihen, miten värimaailmaa joudutaan korjailemaan pelimoottorissa sekä, miten hyvin malli pitää rakenteet kiinni toisissaan. Mallissa voi tulla ongelmia, joita ei saata nähdä ulkoapäin, kuten kuvassa 7. Mallin sisällä voi olla ristiriitoja, jolloin mallin sisälle jää aukkoja, jotka pilaavat mallin pelimaailmassa.



Kuva 7 Mallin mahdollisia ongelmia

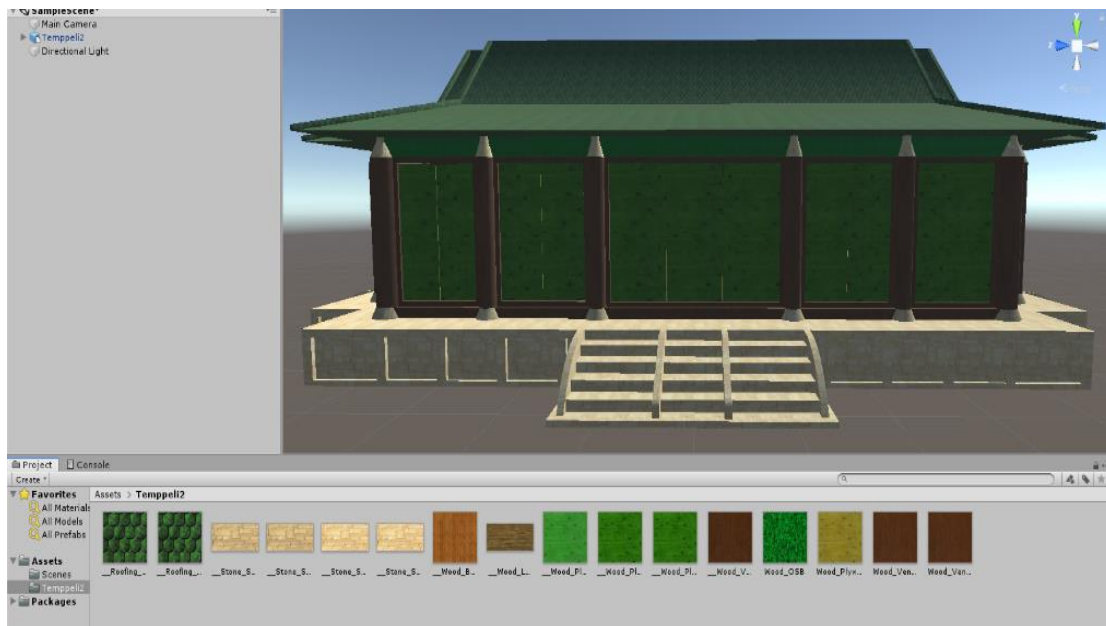
Mallien materiaalien korjaaminen tehdään Unity materials- välilehdellä. Kuvassa 8 havainnollistetaan id:t ja valikot.



Kuva 8 Unityn materiaalikorjaus

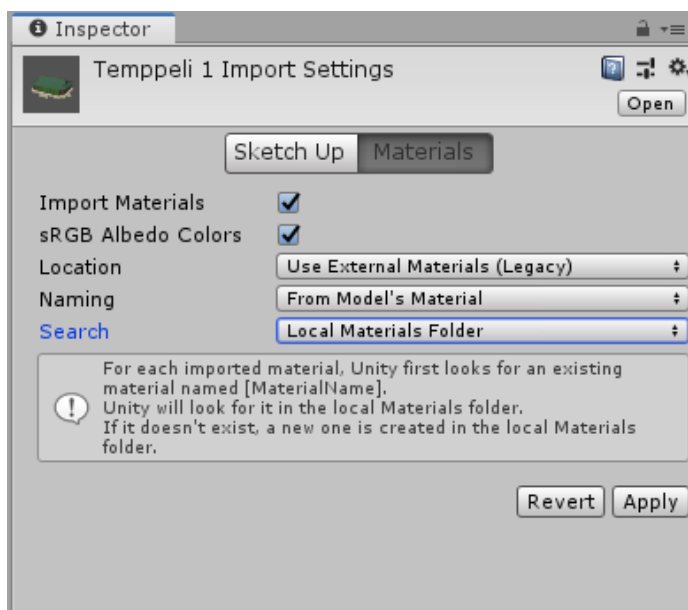
Ohjelma antaa mallin id-nimet valmiiksi, jolloin valitaan tiedostoista oikeat materiaalit. Suoraan raahattu malli tuo mallin unityyn, mutta joskus materiaalit

pitää käydä manuaalisesti asettelemassa paikoilleen. Materiaalien nimet sekä ulkoasun voi tarkistaa tuoduista materiaaleista, kuten kuvassa 9 näkyy.



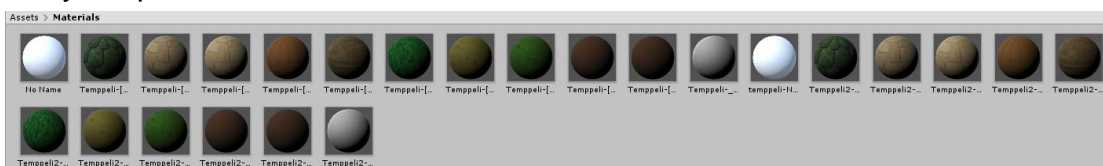
Kuva 9 Mallin värikartta

Helpommalla pääsee, kun valitsee "Materials" -välilehdeltä "Location"-kentästä arvon "Use external materials (legacy)". Tällöin käytetään luotuja materiaaleja. Valitaan "Naming"-kenttään "From model's material" ja "Search"-kenttään "Local materials folder". Tällöin materiaalit etsitään paikallisesta materiaalikansiosista. Kuvassa 10 näkyy Unity Materials-ikkunassa edellä esitetyt valinnat.



Kuva 10 Mallin materiaalien haku Unityssä

Ohjelma sijoittaa materiaalit malliin automaattisesti ja tuo oikeat materiaalit oikeille paikoille. Mallien värikarttaa voi tarkistella värikansiosta, jonka SketchUp luo mallien valmistumisesta. Kuvassa 11 materiaalit näkyvät Unityssä palloina.



Kuva 11 Mallin materiaalit Unityssä

3ds Max ohelman mallit voi tuoda Unityyn monella eri tyylillä. Mallit voi raahata suoraan Unity-ohjelmaan tai tuomalla ne importing-toiminnon kautta. Malleista puuttuu värit, jolloin ne pitää erikseen tuoda ohjelmaan.

3ds Max-mallien materiaalit vääristyvät usein Unityssä, koska ne esim. venyvät mallissa ja näyttävät epäluontevilta. Mallin piilotetut elementit tulevat myös mallin mukana, jolloin ne pitää poistaa mallista manuaalisesti. Esimerkkinä tästä on kuvassa 12 näkyvät valkoiset neliöt.

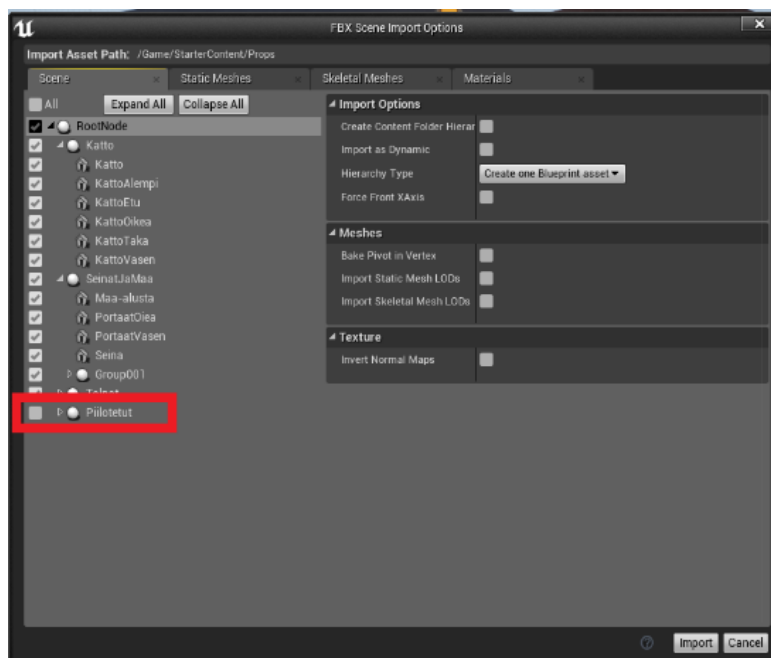


Kuva 12 Mallin piilotetut elementit Unityssä

Mallien kamerat voivat myös tulla mallin mukana, kun mallia tuo Unity-pelimoottoriin. Mallin kamera pitää yleensä poistaa, jotta Unityn omat kamera-asetukset eivät menisi sekaisin.

5.3 Mallien tuonti Unreal Engine-pelimoottoriin

Unreal Engine-pelimoottoriin voi tuoda malleja suoraan, mutta asetuksia kannattaa muokata ensin. ”Scene” -osiossa on painikke ”Expand all”, joka avaa mallien tiedot auki ja ”Collapse all” -painike pienentää näkyvyyden. Jos mallissa on piilotettuja elementtejä, ne kannattaa ottaa valikossa pois päältä, koska mallien leikkaukset saattavat jäädä näkyviin. Kuvassa 13 näkyy huomioidulla neliöllä piilotettu elementti, joka kannattaa jättää valitsematta.

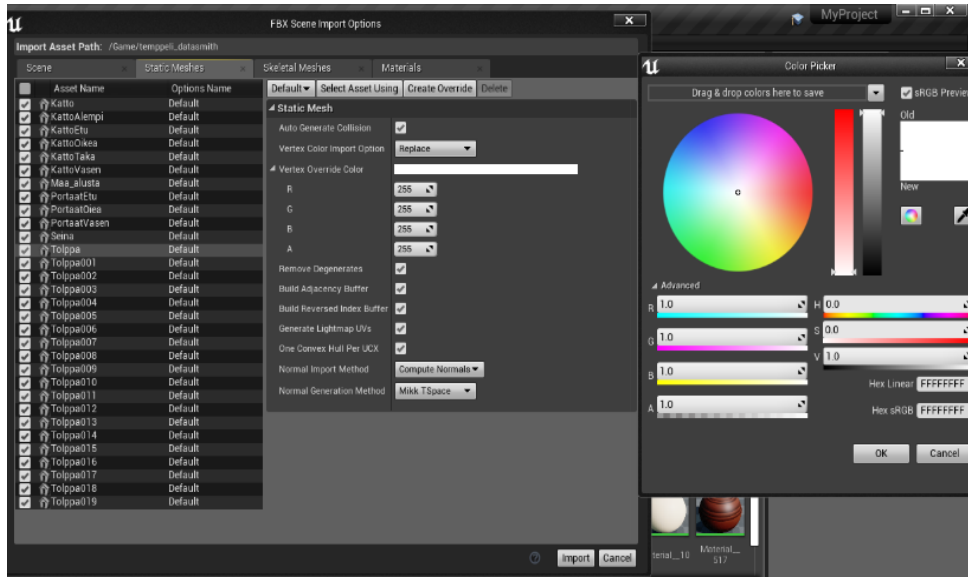


Kuva 13 Mallin piilotetut elementit Unrealissa

Leikkaus-elementit voi poistaa jälkikäteen, mutta se on pitkäjänteistä työtä ja kuluttaa aikaa, koska niitä pitää valita ja poistaa yksitellen. Mallia tuodessa näkyy välilehdet ja aloituslehtenä on ”Scene”. ”Import options” - tuontivaihtoehdossa on ensimmäisenä vaihtoehtona ”Create Content Folder Hierarchy” eli halutaanko tehdä erillisen kansion tuoduille elementeille ja rakentaa kansioista polku rakenteisiin sekä väreille. ”Import as dynamic” - valinnalla voi määrittää, tuodaanko objektit dynaamisina vai staattisina. ”Hierarchy type” -painikkeesta voi valita mallille hierarkia tyyppin. Valittuna oleva ”Create one Blueprint asset” vaihtoehto luo tuottavasta mallista yhden Blueprint-assetin (mallin).

”Force front XAxis” -valinnalla mallien akselia voi muuttaa. ”Bake Pivot in Vertex” -valinnalla FBX-tiedoston alkuperäisiä solmuja käytetään mesh-verkon luomiseen. ”Import Static Mesh LODs” -valinnalla mallin voi tuoda LOD

näkyvyyden eli "Level Of Detail". Tämä mahdollistaa mallin erilaisen tarkkuuden eri etäisyyksiltä., jolloin ohjelma tekee muutamia erillisiä rakenteita mallista. "Import Skeletal Mesh LODs" -hahmoille on oma LOD-näkyvyys, joka tekee saman asian, kuin esineille. "Ivert Normal Maps" -valinta tuo y-akselin normaalina karttana ja kääntää sen käänteiseksi kartaksi Unrealiin. Kuva 14 havainnollistaa mallin tuonti-ikkunan.



Kuva 14 Mallin tuonti Ureal Engine -pelimoottoriin

"Static Meshes" -valinnalla alla sijaitsee mallin rakenteet ja valintapainikkeet. Valintapainikkeista voi valita, mitä osia otetaan mukaan ja mitä jätetään pois mallista. "Default" -valinnan alta voi valita, mitä asetuksia haluaa käyttää mallissa. "Select Asset Import Using" -valinnalla voi valita kaikki mallin rakenteet ja "Create override" -valinnalla voi luoda uuden asetuksen ja antaa sille nimen. "Auto Generate Collision" -valintapainikkeesta voi luoda automaattisesti törmäystunnisteet mallille. "Vertex Color Import Option" -valikosta voi valita verteksin värien tuontiasetukset.

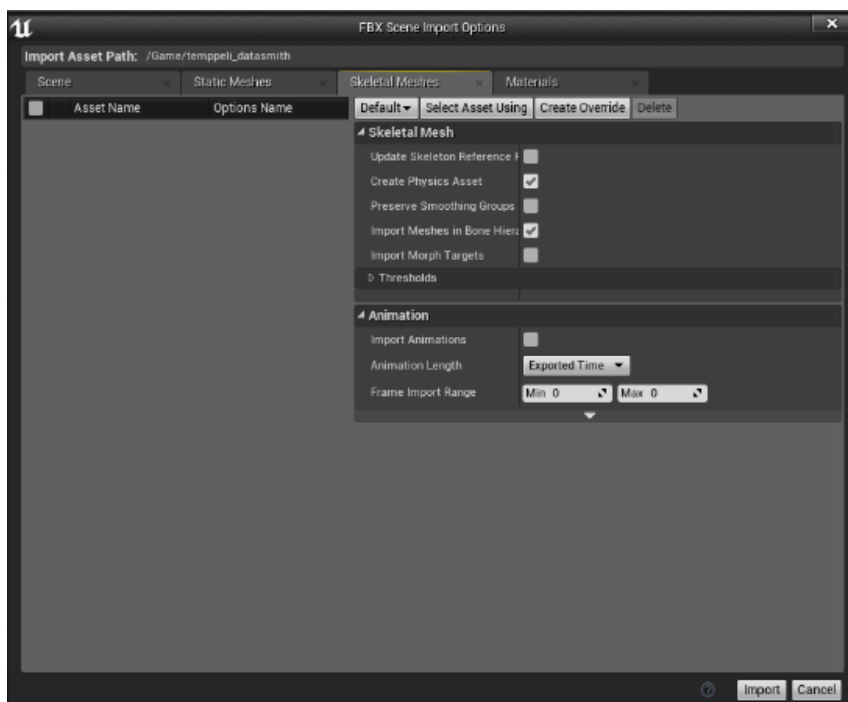
"Vertex override color" -valinnassa väriä voi vaihtaa haluamaansa väriin ja oletuksena on valkoinen. Väriä klikatessa avautuu värikartta, josta voi valita mieleisensä väriin ja vaalentaa/tummentaa väriä. "Drag & drop colors here to save" -toiminnolla valitsemansa väriin voi tallentaa raahaamalla väri ko valinnan päälle. "Remove Degenerates" -valinta poistaa rikkoutuneet kolmiot, jotka voivat pilata mallia. Tämä valinta suositellaan pitämään päällä. "Build Adjacency Buffer" -valinnalla voi poistaa mallista tyhjät kohdat, sekä päällekkäiset osat. Isoissa malleissa kehoitetaan ottamaan painike pois päältä.

”Build Reversed Index Buffer” -valinta luo peilikuvan muunnoksesta.

”Generate lightmap uvs” -valinta luo UV-valokartan. ”One Convex hull per ucx” -valinta yhdistää osia.

”Normal Import method” -valinta määrää kuinka mallin normaalit käsitellään.

”Compute Normals” -tarkoittaa, että pelimoottori laskee normaalit tuontivaiheessa. ”Import Normals” -tuo normaalit sellaisenaan ja ”Import normals and tangents” tuo normaalit ja tangentit. ”Normal Generation Method” -määrittelee normaalien luonti tavan. Oletuksena on ”Mikk TSpace” -eli käytä MikkTSpace-menetelmää generoimaan normeja ja tangentteja.

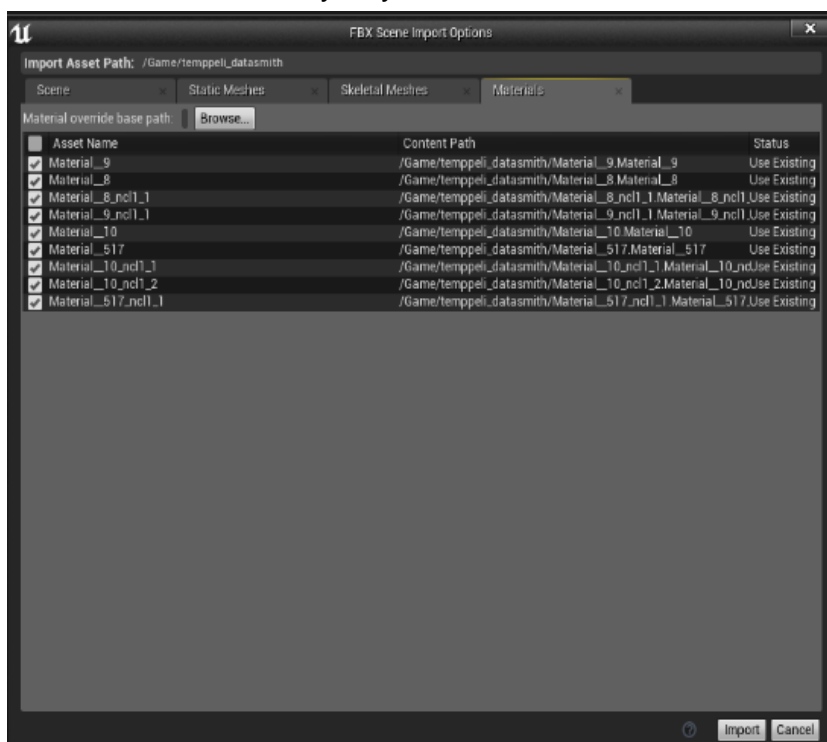


Kuva 15 Mallin luuranko Urealissa

”Skeletal meshes” -on sama valikko kuin aikaisemmassa välilehdessä ”Static meshes”. -Valikossa voi pitää tuonnin perusasetukset päällä, valita kaikki mallin materiaalit ja luoda uusia sääntöjä. ”Update Skeleton Reference Pose” -päivittää luurangolle tuotavan verkon perusasettoon. ”Create physics asset” -valinta mallin fyysiset assetit. ”Preserve Smoothing Groups” -valinta yrittää säilyttää alkuperäiset verkon tasoitusryhmät. ”Import Meshes in Bone Hierarchy” -valinnalla tuodaan sisäkkäiset luurankoverkot. ”Import morph targets” -valinta luo oliolle Unreal muodon tuotuihin verkkoihin.

”Animation” -valinnat antavat valita mallin animaatioihin liittyviä tuontiasetuksia. ”Import Animations” -valinta tuo animaatiot mallin mukana FBX-tiedostosta. ”Animation Length” -valinta määrää animaation pituuden tuontiasetukset. ”Frame import range” -valinnoilla määritellään tuotavat animaation ruudut. Kuva 15 havainnollistaa ”Skeletal Meshes” -asetusikkunaa Unrealissa.

Materiaali-valikossa sijaitsee mallin omat materiaalit, joita on käytetty mallissa. Mallin materiaalit voi tuoda valittuun kansioon ja ne pystytään tuomaan aikaisempien materiaalien päälle. Kuva 16 näyttää materiaalivalikon ja sen, mitä materiaalia on käytetty missäkin rakenteessa.

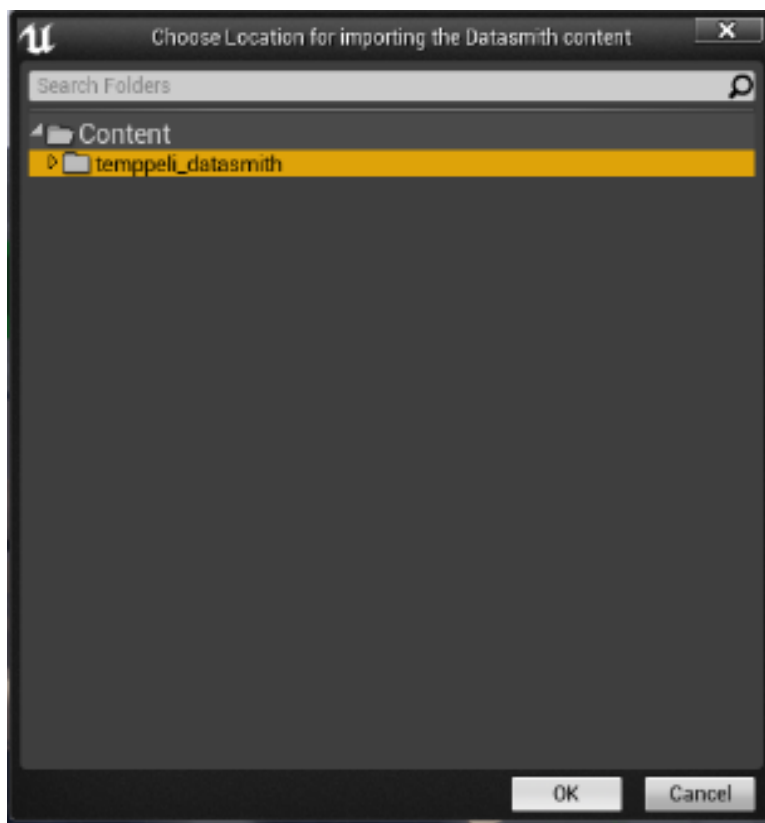


Kuva 16 Mallin materiaalit Unrealissa

Unreal Engine-pelimoottoriin voidaan tuoda malleja suoraan, koska niissä on omat asetuksensa, jotka pitävät mallien rakennetta ja värejä sellaisina kun ne mallinnuksessa on annettu.

Unreal Engine-pelimoottoriohjelmassa pystyy lataamaan myös Dataasmith-lisäosan, joka muuttaa 3D-mallit sopiviksi Unrealia varten. Käyttäjän pitää ensin liittyä beta-ohjelmaan, jotta saa oikeudet käyttää lisäosaa.

Kuva 17 ikkunakuva näyttää tuonnin helppouden.

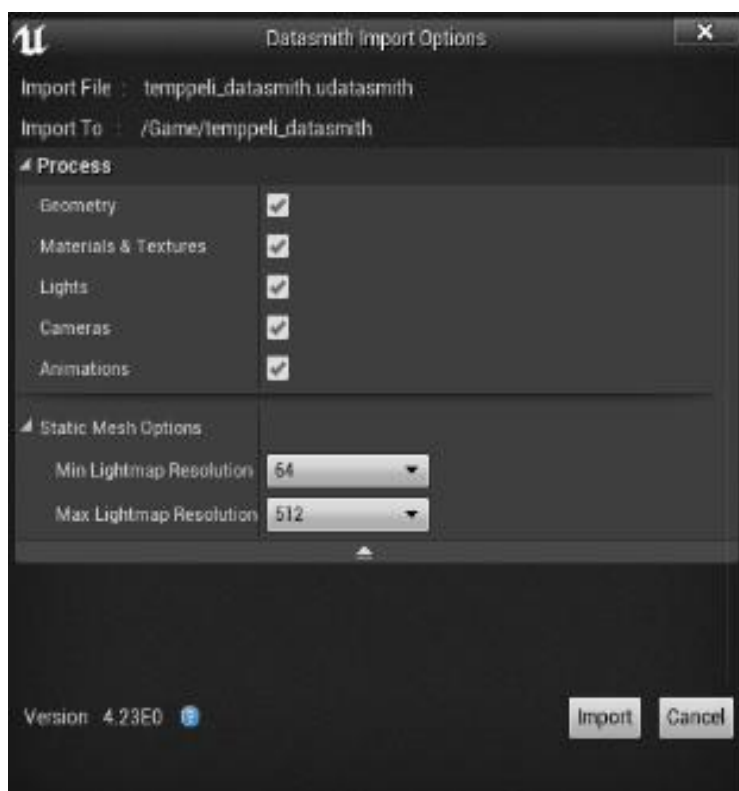


Kuva 17 Mallin tuominen Datasmith lisäosan avulla

SketchUp-malleja ei pysty suoraan tuomaan Unreal Engine-pelimoottoriin. Käyttäjä joutuu päivittämään SketchUp Pro-ohjelman, johon voi ladata Datasmith-lisäosan Unreal Enginen nettisivuilta. Lisäosa on ilmainen ja se asentuu suoraan SketchUp Pro-ohjelmaan. Lisäosa tukee uusinta versiota Unreal Enginestä. Se toimii 3ds Maxin ja SketchUp Pron uusimmissa versioissa. 3ds Max tai SketchUp-mallia tuotaessa Datasmith tiedostona valitaan kansio, mihin malli tuodaan. Mallille kannattaa luoda oma kansio selkeyden takia.

Tuonnin yhteydessä valitaan, mitä tuodaan. "Geometry" -valinnalla tuodaan mallin geometria. "Materials & textures" -valinnalla tuodaan mallin materiaalit ja tekstuurit. "Lights" -valinnalla tuodaan mallin valoastukset. "Cameras" -valinnalla tuodaan malliin liitetyt kamerat. "Animations" -valinnalla tuodaan mallin mahdolliset animaatiot. "Min lightmap resolution" -antaa minimaalisen resoluution automaattisille valokartoille. "Max lightmap resolution" -antaa vastaavasti maksimaalisen resoluution automaattisille valokartoille. "Chord tolerance" -valitsee luotujen kolmioiden välisen enimmäisetäisyyden alkuperäisellä pinnalla. "Max edge length" -antaa generoitujen kolmioiden

reunusten enimmäispituuden. Pienemmät arvot luovat enemmän kolmioita. "Normal tolerance" -antaa oletus normaalitoleranssin luotujen kolmioiden välille. "Stitching technique" -on nidontatekniikka, joka määrää, miten pintoja liitetään toisiinsa. Nidonta voi vaikuttaa mallin monimuotoisuuteen. "Stitching heal" -parantaa nidontaa, "Stitching none" -jättää nitomatta ja "Stitching sew" - antaa paikata nidonnan. Kuva 18 havainnollistaa mallin tuontiasetuksia Datasmith-lisäosassa.



Kuva 18 Mallin tuontiasetukset Datasmith-lisäosassa

Kun asetukset on määritelty ja tuonti aloitetaan painamalla import-nappia, ohjelma rakentaa mallin tuontiasetuksissa määritetyllä tavalla. Lopputuloksena malli on valmis malli niillä asetuksilla, jotka mallille on annettu mallin luonnin yhteydessä. Unreal Engine-pelimoottori luo mallille omat kansiot, joissa on värikartat ja mallin osat. Datasmith tiedostoissa on ongelmana, että se ei välttämättä tunnista kaikkia mallin materiaaleja. Kuvassa 19 tuodut mallit ovat kaikki tuotu erillään, mutta niissä kaikissa on sama malli ja samalla ohjelmalla tehtynä.



Kuva 19 Mallit tuotuina eri asetuksilla

Rakennuksen materiaaleina on puu ja marmori, mutta ensimmäisestä rakennuksesta puuttuu puu, ja ohjelma on tuonut pelkät värit kattoon ja seiniin. OBJ- ja FBX-tiedostoissa on kaikki materiaalit ja värit oikein. Ongelmana tuotaessa muita tiedostoja kuin Datasmith tiedostoja ovat valikoiden määrät ja kaikkien valintojen merkitys.

Jos haluaa tuoda mallin ilman lisäosia, kannattaa malli tuoda OBJ- tai FBX-tiedostona, joita Unreal Engine tukee melko hyvin. Valitettavasti SketchUp-ohjelmassa ei tue tiedostomuotoja, joita Unreal Engine tukee. Tällöin ne pitää muuntaa toiseen muotoon joillain muulla ohjelmalla tai on hankittava Pro-versio ohjelmasta. Tällöin Ohjelmasta voi tuoda .OBJ- ja FBX-tiedostoja tai ohjelmaan pystyy asentamaan Datasmith-apuohjelman. Tuotaessa Datasmith mallia Unreal Engineen, voi tulla ongelmia. Esimerkiksi värit voivat muuttua ja materiaalien kuviot voivat suurentua liikaa.

FBX-tiedoston tuonti Sketchup mallista voi olla joskus haastavaa. Joitakin elementtejä voi jäädä pois, jolloin mallista puuttuu osia, kuten kuvassa 20 on nähtävissä.



Kuva 20 Mallin rakenne on tullut virheellisesnä Unreal Engine-pelimoottoriin

Esimerkin mallissa edget ovat pirstoutuneet ja kiinnityskohdat ovat jääneet vajavaisiksi. Mallia joutuu mahdollisesti korjaamaan mallinnusohjelmassa. Toinen syy rakenteiden virheisiin on voinut tulla mallin export-toiminnossa, jolloin joitakin elementtejä on jäänyt tiedostosta pois. Malli kannattaa kokeilla tuoda uudestaan export-toiminnon kautta ja siirtää import-toiminnalla Unreal-ohjelmaan, jotta nähdään, onko vika ollut export-toiminnossa, vai onko mallissa virheitä. Mallin rakenteessa voi olla ristiriitoja, joita pitää korjata ennen tuontia.

5.4 Kolmiulotteisten mallien tiedostomuodot

Tiedostomuodot määräytyvät yleensä mallinnusohjelmien mukaan. Malleja voi export- ja import -toiminnalla siirtää myös yleisillä tiedostomuodoilla. Kaikki ohjelmat eivät tue kaikkia tiedostomuodoja. Kuitenkin 3D-mallinnusohjelmat tukevat yleensä oman tallennusmuotojensa lisäksi mallien tuontia ja vientiä (import/export) muissa tallennusmuodoissa. Esim. SketchUp käyttää .skp-muotoa, mutta export-toiminnolla tiedoston voi tallentaa myös .dae- tai .kmz-muodossa. SketchUp Pro-ohjelmassa on laajempi mahdollisuus viedä export-toiminnolla tiedostoja eri muodoissa. SketchUp Pro:n tukemia tallennusmuodoja ovat .skp, .3ds, .swg, .dxf, .dae, .fbx, .itc, .kmz, .obj, .stl, .uDatasmith, .wrl ja .xsi.

Autodesk 3ds Max-ohjelmalla on .max perustiedostomuotona sekä lisäksi muita tuettuja tiedostomuotoja ovat .chr, .Fbx, .3ds, .abc, .ai, .ase, .ass, .dae, .dwf, .dwg, .dxf, .flt, .htr, .igs, .obj, .pxproj, .sat, .stl, .svf, .wrl ja uDatasmith.

5.5 Yleisesti 3D-mallien käytöstä pelimoottoreissa

3D-mallit ovat yleisiä esimerkiksi peleissä ja monet pelialan yritykset tekevät mallinsa ja rakentavat pelimaailmansa itse. Malleja pystyy myös ostamaan ja lataamaan netistä ja muokkaamaan omanlaisiksi.

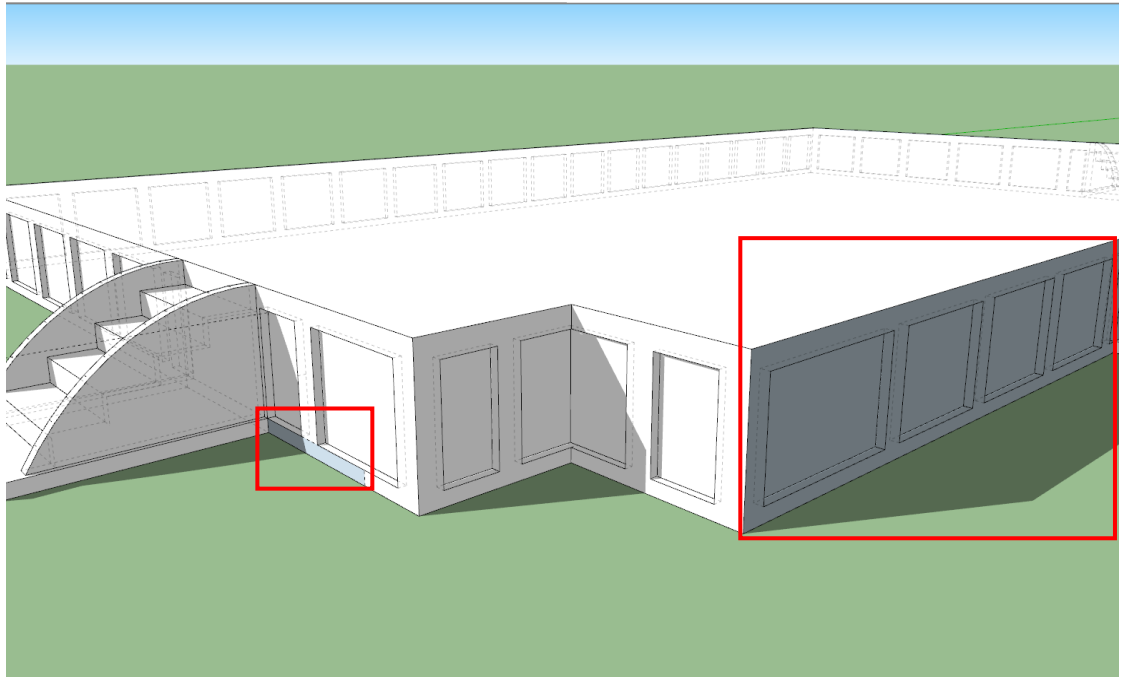
Mallit pyritään tekemään yleensä mahdollisimman aidon näköisiksi.

Nykypäivänä monet peliyhtiöt pyrkivät realistisiin malleihin ja pyrkivät saamaan hahmot näyttämään elokuvamaisilta saaden siten aitouden tunnetta peliin.

Useat yritykset käyttävät useita mallinnusohjelmia luodessaan malleja, koska kaikissa ohjelmissa on omat etuutensa ja jotkut ohjelmat ovat erikoistuneet tietynlaisiin malleihin. Useat yritykset ovat myös muokanneet mallinnusohjelmia ja parantaneet niiden toimintoja sekä tuoneet omia ”lisäosia” ohjelmiin. Yleensä pelimaailma rakentuu 3D-malleista ja sen maalustasta, johon lisätään perusaineita, kuten puita, nurmikkoa, maata ja vettä, joilla saadaan pelimaailma elämään tavoitteidensa mukaisesti.

5.6 Työn vaiheet ja ongelmat

Ensimmäinen vaihe oli tehdä 3D-mallit, joita hyödynnettiin opinnäytetyössä. Mallintamisessa meni paljon aikaa, koska mallin aihe piti valita ja mallintaa malli kahdella eri mallinnusohjelmalla. Malleja piti tallentaa eri tiedostoformaateissa, jotta pystyi tutkimaan niitä pelialustalla ja toteamaan niiden toimivuuden. SketchUp-mallia piti korjata, koska mallia mallintaessa polygonit (monikulmiot) ja edget (reunukset) olivat siirtyneet vääriin kohtiin, jolloin mallista puuttui kohtia, kun sen pelillisti. SketchUp-ohjelmassa pääsee näkemään mallin virhekohdat, mikä helpottaa ja nopeuttaa korjausta. Kuvassa 21 näkyy tummemmat kohdat, jotka kertovat virheestä mallissa. Virhe mallissa voi vaikuttaa myöhemmin pelillistämiseen.



Kuva 21 Mallin virheiden sijainnit SketchUp-ohjelmassa

Mallin tummat kohdat näyttävät ristiriidat, jolloin niitä korjaamalla malli toimii pelimaailmassa oikein, eikä jätä mallista elementtejä pois. Mallien staattisia tietoja pystyi seuraamaan, kuten edges (reunus) ja faces (pintojen) määrää. SketchUp-mallissa edget olivat 8428 ja faces 3574.

Pelimaailmassa luotiin pelkkä maa-alusta, johon mallit tuontiin. Valotuksia korjattiin pelialustalla. SketchUp-mallien kanssa oli hankaluuksia, koska Unreal-ohjelma ei tue suoraan sen tiedostoformaattia, jolloin jouduttiin lataamaan SketchUp Pro-versio ja siirtämään mallit sen kautta. Myös Unreal Datasmith toimii ainoastaan SketchUp-ohjelmien Pro-versiossa. SketchUp Pro on maksullinen ja sitä voi käyttää ilmaiseksi vain kuukauden. Aikaa ei ollut paljon, jotta mallit saisi toimimaan oikein eikä tulisi ongelmia lisenssin sulkeuduttua. 3ds Max-ohjelman kanssa välttyttiin suuremmilta ongelmilta, koska sen tukemissa formaateissa viedyt mallit toimivat suoraan Unity- ja Unreal Engine-pelimoottoreissa. 3ds Max-ohjelmaa voivat opiskelijat käyttää ilmaiseksi, joten lisenssien sulkeutumisesta ei tarvinnut huolehtia. Mallien kokoja muutettiin mallinnusohjelmissa ja tutkittiin, vaikuttaako mallin koko mallin näkymiseen ja toimimiseen, kun malli tuodaan pelimoottoriin.

6 YHTEENVETO

Työvälineinä oli Unreal Engine- ja Unity-pelimoottoriohjelmat, joilla testattiin mallien optimointia ja toimivuutta. Pientä eroa oli ohjelmissa ja niiden toimivuudessa. Kummassakin pelimoottorissa mallit näyttivät hyviltä. Unreal Datasmith lisäosaa käytettiin mallien tuonnissa Unreal Engine-pelimoottoriin. Unityssä toimii paremmin SketchUp-mallit, koska valmistajat ovat tehneet sopimuksen keskenään. 3ds Max-ohjelman mallia jouduttiin korjaamaan Unity-pelimoottorilla. Unity-pelimoottori ei tunnistanut 3ds Max-mallin upotuksia ja toi upotetut mallit virheellisesti näkyviin. Unreal Engine-pelimoottoriin oli hankalampi saada mallit ja mallien väritykset oikein. SketchUp-malleja sai tuotua Pro versiosta. Lataamalla Datasmith lisäosan SketchUp Pro-ohjelmaan, saa mallit tuotua helpommin Unreal Engineen.

Pelimaailmassa 3D-mallit ovat keskeinen asia ja ilman niitä ei pysty tekemään 3D- tai 2½D-pelejä. Mallintamisessa pitää olla tarkka ja huolellinen, koska muussa tapauksessa pitää mallia korjata jälkikäteen ja tämä voi viedä paljon aikaa. Mallinnusohjelman monipuolisia työvälineitä kannattaa käyttää, jotta mallintaminen nopeutuu ja helpottaa työskentelyä. On hyvä myös osata luoda valotukset uudelleen sekä korjata valmiita elementtejä. Pelillistämisessä pitää mallin rakenteet ja materiaalit sekä muut osat osata tuoda oikein. On tärkeää valita oikea 3D-mallin tiedostomuoto. Sen pitää olla yhteensopiva pelillistämiseen käytettävän pelimoottorin kanssa. Havaintoni mukaan parhaimpia tiedostomuotoja tuontiin ovat OBJ- sekä FBX-formaatit. Itse pelillistämisen vaihe on lopulta ehkä helpoin osuus, koska pelimoottoreissa on omia pitkälle kehittyneitä apu- tai työvälineitä, joilla saadaan 3D-mallit tuotua laadukkaina ja melko automaattisesti pelimaailmaan.

LÄHTEET

Ac3D. 2019. Features. WWW-dokumentti. Saatavissa: <https://www.inivis.com/> [viitattu 3.9.2019].

Adobe. 2019. Substance. WWW-dokumentti. Saatavissa: <https://www.substance3d.com/> [viitattu 12.9.2019].

Autodesk. 3ds Max. 2019. Features. WWW-dokumentti. Saatavissa: <https://www.autodesk.com/products/softimage/overview> [viitattu 10.8.2019].

Daniel. 2019. How to optimize your 3d model for faster Lumion performance. Lumion. WWW-dokumentti. Saatavissa: <https://lumion.com/blog/how-to-optimize-your-3d-model-for-faster-lumion.html> [viitattu 5.3.2020].

Game Dev Academy. 2015. What is LOD? (Level of Detail & LOD Group). Youtube. Video. Saatavissa: <https://www.youtube.com/watch?v=mlkIMgEVnX0> [viitattu 10.8.2019].

Gamedesigning. 2019. 3D Game Modeling: Beginners Guide. WWW-dokumentti. Saatavissa: <https://www.gamedesigning.org/learn/3d-modeling/> [viitattu 17.9.2019].

Kontturi, H. 2016. 3d-Mallinnuksen ja tietomallinnuksen hyödyntäminen rakennusallalla. Opinnäytetyö. Kevät 2016. Seinäjoen Ammattikorkeakoulu. Rakennustekniikka. Verkkosivuja 66. PDF-dokumentti. Saatavissa: <https://www.theseus.fi/bitstream/handle/10024/131811/3D%20mallinnus%20ja%20tietomallinnus.pdf?sequence=1&isAllowed=y> [viitattu 26.6.2019].

Lanusse, I. 2015. 3 Tips to Optimize your 3D Scenes. Wolox. WWW-dokumentti. Saatavissa: <https://medium.com/wolox-driving-innovation/3-tips-to-optimize-your-3d-scenes-864b498fcb0> [viitattu 19.11.2019].

Lohikoski, L. Rudén, E. 2013. Optimization of 3D Game Models. Södertörns högskola. Institutionen för naturvetenskap, miljö och teknik. Kandidatuppsats. PDF-dokumentti. Saatavissa: <https://www.diva-portal.org/smash/get/diva2:708048/FULLTEXT01.pdf> [viitattu 4.3.2020].

Maxon. 2019. Polygonal modeling. WWW-dokumentti. Saatavissa: <https://www.maxon.net/en/products/cinema-4d/features/modeling/polygon-modeling/> [viitattu 3.8.2019].

Meshlab features. Meshlab. WWW-dokumentti. Saatavissa: <http://www.meshlab.net/> [viitattu 11.2.2020].

Microsoft Docs. 1.10.2019. Best practices for converting and optimizing real-time 3D models. WWW-dokumentti. Saatavissa: <https://docs.microsoft.com/en-us/dynamics365/mixed-reality/import-tool/best-practices> [viitattu 17.11.2019].

Rajakyläteknö. 2016. Mallinnuksen perusteet. Powerpoint-diasarja. Päivitetty 3.11.2016. Saatavissa: <https://rajakylatekno.files.wordpress.com/2016/11/3d-mallinnuksen-perusteet-2016.pptx> [viitattu 30.7.2019].

SideFX 2019. Houdini. WWW-dokumentti. Saatavissa: <https://www.sidefx.com/products/houdini/> [viitattu 6.8.2019].

Toolfarm. 2019. NewTek LightWave 3D. WWW-dokumentti. Saatavissa: https://www.toolfarm.com/buy/newtek_lightwave_3d/ [viitattu 3.9.2019].

Unity Documentation. 2019. Unity Documentation. WWW-dokumentti. Saatavissa: <https://docs.unity3d.com/Manual/index.html> [viitattu 30.7.2019].

Unreal Engine. 2019. Datasmith export plugins. WWW-dokumentti. Saatavissa: <https://www.unrealengine.com/en-US/datasmith/plugins> [viitattu 28.6.2019]

Unreal Engine. 2019. Datasmith Supported Software and File Types. WWW-dokumentti. Saatavissa: <https://docs.unrealengine.com/en-US/Engine/Content/Importing/Datasmith/SupportedSoftwareAndFileTypes/index.html> [viitattu 5.9.2019]

Valve Developer Community. 2013. LOD Models. WWW-dokumentti. Saatavissa: https://developer.valvesoftware.com/wiki/LOD_Models [viitattu 28.8.2019].