

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikan koulutusohjelma

2020

Rauli Haapasaari

CAD-MALLIN TUOMINEN VR-YMPÄRISTÖÖN



Rauli Haapasaari

CAD-MALLIN TUOMINEN VR-YMPÄRISTÖÖN

Kiinnostus virtuaalitodellisuuden käyttöön teollisuuden eri aloilla, kuten rakennusteollisuudessa on kasvussa. Tämän myötä myös pelimoottorien käyttö virtuaaliympäristöjen rakentamisessa on yleisempää. CAD-mallin tuominen pelimoottoriin on tyypillisesti työläs ja aikaa vievä prosessi, joten opinnäytetyön tavoitteena on perustaa nopeampi toimintatapa tuoda CAD-malli Unity-pelimoottoriin.

Opinnäytetyössä perehdyttiin käytettyihin teknologioihin ja työkaluihin sekä niiden termistöön ja historiaan. Seuraavaksi käytiin vaihe vaiheelta läpi CAD-mallin käsittely 3D-mallinnusohjelmassa ja sen tuonti Unity-pelimoottoriin, jonka jälkeen tuotiin sama malli Unityyn Pixyz-liitännäistä käyttäen. Lopuksi toimintatapoja vertailtiin keskenään.

Opinnäytetyön tuloksena syntyi toimintatapa, jolla voi potentiaalisesti nopeuttaa CAD-mallin tuontiprosessia Unityyn, ja se otettiin käyttöön asiakasprojektissa.

ASIASANAT:

VR, CAD, Unity 3d, 3D-mallinnus

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Information and Communications Technology

2020 | 38 pages

Rauli Haapasaari

BRINGING CAD MODELS TO VIRTUAL REALITY ENVIRONMENT

Interest in the use of virtual reality in various industrial areas, such as the construction industry, is growing. As a result, the use of game engines in building virtual environments is also becoming more common. Importing a CAD model into a game engine is typically a laborious and time-consuming process, so the goal of the thesis is to establish a faster workflow for the commissioner.

The thesis studied the technologies and tools used and their terminology and history. The next step was to process the CAD model in a 3D modeling program and import it into the Unity game engine, after which the same model was imported into Unity using the Pixyz plugin. Finally, the workflows were compared.

As a result of the thesis, a workflow was created that could potentially speed up the process of importing a CAD model into Unity and it was put in use in the customer project.

KEYWORDS:

VR, CAD, Unity 3d, 3D modelling

SISÄLTÖ

KÄYTETYT LYHENTEET TAI SANASTO	6
1 JOHDANTO	7
2 3D-MALLINTAMINEN JA VR	9
2.1 3D-mallintaminen yleisesti	9
2.1.1 CAD-ohjelmat	9
2.1.2 3D-mallinnusohjelmat	10
2.1.3 Blender	10
2.1.4 3ds Max	10
2.2 Virtual Reality (VR) ja pelimoottorit	11
2.2.1 Unity	12
2.2.2 Unreal Engine	12
2.2.3 CAD-malleista pelimoottoreihin	13
2.3 CAD-mallin tuomisprosessi Unity-pelimoottoriin	13
2.3.1 Pixyz	14
2.3.2 Muut vaihtoehdot tuoda CAD-malli VR-ympäristöön	15
3 TAPAUS ADE OY	16
3.1 Lähtötila	17
3.1.1 Mallin tuonti 3ds Maxiin ja optimointi	17
3.1.2 Mallin UV-kartoitus	22
3.1.3 Mallin animointi ja vienti	23
3.1.4 Mallin tuonti Unityyn	24
3.2 Työprosessin kehittäminen	26
3.2.1 Pixyz-liitännäisen ensivaikutelmat	26
3.2.2 Mallin tuonti Unityyn Pixyz-liitännäistä käyttäen	27
3.2.4 Mallin valmistelu	29
3.2.5 Mallin animointi Unityssä	30
3.2.6 Lopputulos	31
4 POHDINTA JA JOHTOPÄÄTÖKSET	33
LÄHTEET	35

KUVAT

Kuva 1. Telesphere Mask. (Virtual reality society 2017.)	11
Kuva 2. Esimerkkimalli. (GrabCad.)	16
Kuva 3. Olemassaoleva toimintatapa ja työnjako.	17
Kuva 4. 3ds Maxin tuonti-ikkuna.	18
Kuva 5. Esimerkkimallin polygon- ja verteksimäärä.	18
Kuva 6. Optimointimodifier voi hajoittaa objektin geometriaa.	19
Kuva 7. Uudelleen mallinnettu pöydänjalka.	19
Kuva 8. Esimerkkimallin hierarkia tuonnin jälkeen.	20
Kuva 9. Uudelleenjärjestetty hierarkia.	20
Kuva 10. Hierarkia geometrioiden yhdistelyn jälkeen.	21
Kuva 11. Shakkiruutumateriaali esimerkkimallissa.	22
Kuva 12. 3ds Maxin aikajana.	23
Kuva 13. 3ds Maxin vienti-ikkuna.	24
Kuva 14. Unityn tuontiasetukset.	25
Kuva 15. Metallic ja smoothness arvon muuttamisen vaikutukset.	26
Kuva 16. Pixyz-liitännäisen tuonti-ikkuna.	28
Kuva 17. Esimerkkimallin polygon-määrät eri laatuasetuksilla.	29
Kuva 18. Merge kontekstivalikossa.	29
Kuva 19. Explode kontekstivalikossa.	30
Kuva 20. Animaatiokomponenttien luonti.	30
Kuva 21. Unityn aikajana.	31
Kuva 22. Muutokset aiempaan toimintatapaan sekä uusi toimintatapa.	32

KÄYTETYT LYHENTEET TAI SANASTO

CAD	Computer Aided Design, eli tietokoneavustettu suunnittelu
Edge	Kaksi verteksiä yhdistävä suora viiva.
Interpolointi	Kahden arvon välissä olevien arvojen suoraviivainen laskeminen
Iterointi	Menetelmä, jossa työvaiheet toistetaan, kunnes saadaan haluttu tulos.
Komponentti	Unityssä objektin osa, joka määrittelee sen toiminnallisuudet.
LOD	Level Of Detail, LOD-malleissa niiden monimutkaisuus tai polygonmäärä pienenee, kun se jää kauemmas katsojasta.
Metadata	Malliin liitetty informaatio eri sovelluksissa.
Modifier	3D-mallinnusohjelmassa objektiin lisättävä osa, jolla voi manipuloida objektia eri tavoilla.
Niputus	Unityn staattinen tai dynaaminen niputus (Batching), joiden avulla renderöintiä voi nopeuttaa.
Polygon	Suorasivuinen monikulmio, joka koostuu edgeistä ja vertekseistä.
Renderöinti	2D-kuvan muodostaminen 3D-ympäristöstä
SDK	Software Development Kit, ohjelmistokehitystyökalu
Verteksi	Kolmiulotteinen piste.
VR	Virtual Reality, virtuaalitodellisuus

1 JOHDANTO

Kiinnostus virtuaaliodellisuuden käyttöön teollisuuden eri aloilla, kuten rakennusteollisuudessa on kasvussa. Tämän myötä myös pelimoottorien käyttö virtuaaliympäristöjen rakentamisessa on yleisempää. Teollisuuden aloilla suunnittelussa käytetään pääsääntöisesti CAD-työkaluja, joilla tuotteita tai rakennuksia pystyy visualisoimaan kolmiulotteisesti. Näitä CAD-työkaluilla luotuja malleja ei kuitenkaan voi tuoda pelimoottoreihin suoraan, koska pelimoottorit eivät tue niitä.

Opinnäytetyön aiheena on tuoda CAD-malli Unity-pelimoottoriin, jossa sitä voidaan tarkastella HTC-Vivellä virtuaaliympäristössä. CAD-malli tuodaan perinteisesti ensin 3D-mallinnusohjelmaan, jossa se optimoidaan ja käännetään muotoon, jota Unity tukee. Jotkin osat tästä toimintatavasta ovat tyypillisesti kuitenkin hyvin paljon aikaa vieviä ja raskaita resurssoinnin näkökulmasta varsinkin kun aikataulut ovat tiukkoja.

Keväällä 2018 alkanut työ on toiminnallinen opinnäytetyö. Opinnäytetyön tavoitteena on perustaa toimeksiantajalle uusi tai muokata olemassa olevaa toimintatapaa tuoda CAD-malli Unity-pelimoottoriin. Malli täytyy saada pelimoottoriin riittävällä optimoinnilla, jotta sitä pystyy katselemaan HTC Vivellä virtuaaliodellisuudessa. Uuden toimintatavan tarkoitus on automatisoida tai jollain tapaa nopeuttaa prosessia.

Toimeksiantajalla on pitkäkestoinen asiakasprojekti, jonka tarkoituksena on rakentaa virtuaalinen esittely-ympäristö tuotantolinjastosta ja niissä käytetyistä laitteista sekä moduuleista. Toimeksiantajan projektin ympärille oli valittu työkalut ja luotu workflow eli toimintatapa, jolla asiakkaan olemassa olevia CAD-malleja saatiin tuotua Unity-pelimoottoriin, jossa sitä tarkastellaan virtuaaliympäristössä.

Opinnäytetyö alkaa perehtymällä käytettyihin teknologioihin ja työkaluihin sekä niiden termistöön ja historiaan. Ensin käydään läpi 3D-mallintamisen yleisesti, josta siirrytään CAD-malleihin ja 3D-mallinnukseen, jonka jälkeen perehdytään pelimoottoreihin ja olemassa oleviin toimintatapoihin tuoda CAD-malli Unity-pelimoottoriin.

Työosuudessa käydään ensin vaihe vaiheelta läpi CAD-mallin tuominen 3ds Maxiin, sen optimointi, hierarkian järjestely, UV-kartoitus ja avainkehysanimointi (keyframe animation). Lopuksi malli tuodaan Unity-pelimoottoriin. Tämän jälkeen sama malli tuodaan Unity-pelimoottoriin Pixyz-liitännäistä käyttäen ja pyritään pääsemään lähes samaan tai parempaan lopputulokseen. Lopulta toimintatapaa verrataan aiempaan ja pohditaan,

onko tämä uusi toimintatapa käyttökelpoinen ja voiko sillä korvata vanhan toimintatavan kokonaan tai osittain. Opinnäytetyössä ei käydä läpi ohjelmointiosuutta tai viimeistelyä, koska siinä keskitytään CAD-mallin käsittelyyn ja tuontiin Unity-pelimoottoriin.

CAD-mallien tarkastelua virtuaalitodellisuudessa on tutkittu aiemminkin opinnäytetöissä. Esimerkiksi Keituri (2018) ja Heinonen (2019) tuovat CAD-mallin virtuaalitodellisuuteen, mutta Pixyz-liitännäisen käytöstä on hyvin vähän tietoa. Pixzys-liitännäistä käytetään Muuriaisniemen (2019) samalle toimeksiantajalle tehdyssä opinnäytetyössä Building Information Model (BIM) -datan tuomiseen mallin mukana pelimoottoriin.

2 3D-MALLINTAMINEN JA VR

2.1 3D-mallintaminen yleisesti

3D-mallintaminen on helpottanut ja parantanut suunnittelutyötä. Muotojen ja tilavaatiusten helpomman visualisoinnin lisäksi, sillä saadaan parannettua piirtotarkkuutta ja tehokkuutta. (Jacobs 2014.)

Vuonna 1963 Ivan Sutherlandin kirjoittama Sketchpad oli ensimmäinen 3D-mallintamiseen tarkoitettu ohjelma. Sillä pystyi mallintamaan yksinkertaisia objekteja kuten kuutioita ja primoja. 70- ja 80-luvulla 3D-mallinnusohjelmien käyttö alkoi yleistyä tietokoneen kalliista hinnasta huolimatta. Se johti ohjelmien ominaisuuksien, kuten tekstuuri- ja kumukartoitusten kehittymiseen ja yksinkertaisten mallien sijasta niillä saatiin aikaisiksi realistisempia lopputuloksia. 90-luvulla saapui kaupallisia mallinnusohjelmia kuten Autodeskin 3D Studio, joita käytetään vielä tänäkin päivänä (Easy Render 2019).

2.1.1 CAD-ohjelmat

Computer Aided Design (CAD) ohjelmat ovat varsinkin tuotannollisilla aloilla yleisimmin käytettyjä suunnittelutyökaluja. Niillä pyritään mahdollistamaan erittäin tarkkojen teknisten mallien tekeminen, esimerkiksi koneellista tuotantoa tai 3D-tulostusta varten. Työ aloitetaan usein luomalla tekninen 2D-piirustus, josta se laajenee 3D-malliksi lisäämällä objektin syvyyttä vaativia osia kuten materiaalin paksuutta, koloja ja säikeitä. (Plewa 2019.)

CAD-ohjelmilla tehtyjen 3D-mallien fokus on mallien oikeiden mittasuhteiden ja tarkkuuden saavuttaminen niiden reaali maailman toteuttamiskelpoisuuden varmistamiseksi (Sanders 2019). Yleisiä CAD-ohjelmia ovat muun muassa AutoCAD, Solidworks ja Catia.

2.1.2 3D-mallinnusohjelmat

Siinä missä CAD-ohjelmat keskittyvät oikeisiin mittasuhteisiin ja mallien tuotannolliseen käytettävyyteen, 3D-mallinnusohjelmilla mallintamista lähestytään abstraktimmalla tasolla. Niissä työstetään 3D-muotoja teknisten 2D-piirustusten sijaan. (Plewa 2019.)

Koska 3D-mallinnusohjelmilla on tarkoitus luoda objekteja, hahmoja tai kokonaisia maailmoja esimerkiksi pelien ja animaatioiden käyttöön, niiden mukana tulevien työkalujen fokus on enemmän artistinen, kuin tekninen. Niillä voidaan keskittyä tekstuuriin tarkkuuteen ja kuvien yksityiskohtiin, kuten hahmojen hiuksiin, puiden yksittäisiin lehtiin ja niiden liikkeeseen tuulen tai sateen vaikutuksesta. Niillä tavoitellaan yleensä mahdollisimman visuaalisesti näyttävää lopputulosta. Yleisiä 3D-mallinnusohjelmia ovat esimerkiksi 3ds Max ja Blender. (Sanders 2019.)

2.1.3 Blender

Blender on Ton Roosendaalin alun perin vuonna 1995 luoma 3D-mallinnusohjelma, joka muuttui avoimen lähdekoodin ohjelmaksi vuonna 2002. Vapaaehtoiset kehittäjät ympäri maailman ovat kehittäneet työkalua siitä lähtien Ton Roosendaalin johtamana (Blender 2013). Blenderin ominaisuuksiin kuuluu muun muassa 3D-mallinnus ja animointi, simulointi, renderöinti ja liikkeiden seuranta (motion tracking).

Avoimen lähdekoodin ohjelmana Blender on ilmainen ja sillä onkin arviolta 1-3 miljoonaa aktiivista käyttäjää ja se ladattiin vuonna 2019 yli 10 miljoonaa kertaa (Siddi 2020). Suosioon vaikuttaa hinnan lisäksi myös yhteisölähtöisyys ja siihen löytyy paljon tutoriaaleja, joten sillä on helppo aloittaa 3D-mallinnus (Plewa 2018).

2.1.4 3ds Max

Autodeskin 3D Studio on 1990-luvulla julkaistu ensimmäinen hinnaltaan edullinen 3D-mallinnusohjelma, joka tunnetaan nykyään nimellä 3ds Max. Toiminnallisuudet olivat julkaisun aikoihin ennennäkemättömiä. 3D Studio saikin valtavan suosion ja sitä käytettiin jo ennen 2000-lukua monissa TV- ja elokuvastudiossa, sekä pelistudioissa, kuten esimerkiksi Blizzardilla. (InspirationTuts 2019.)

3ds Max onkin edelleen yksi suosituimmista 3D-mallinnusohjelmista ammattikäytössä ja sitä käytetään niin hahmojen mallinnukseen ja animointiin, kuin fotorealististen kuvien renderointiin (Petty). Toimeksiantajalla oli valmiiksi lisenssit 3ds Maxiin ja sitä käytetään toimeksiantajan muissakin projekteissa. Tästä syystä sitä käytetään myös tässä opin- näytetyössä.

2.2 Virtual Reality (VR) ja pelimoottorit

Kielitoimiston sanakirjan mukaan, virtuaalitodellisuus on ”tietokonesimulaation tuotta- mien aistimusten avulla luotu keinotekoinen ympäristö, keino-, lume-, tekotodellisuus” (Kotimaisten kielten keskus 2018). Tästä määrityksestä huolimatta virtuaalitodellisuuden voidaan ajatella olevan mikä tahansa ratkaisu, jolla aivoja voidaan huijata luulemaan epätosi todeksi (Virtual reality society 2017a).

Varhaisempia nykyisien VR-lasien kaltaisia toteutuksia olivat jo 1800-luvun alkupuolella kehitetyt stereoskoopit. 1950-luvulla Morton Heilig kehitti Sensoraman, jolla pyrittiin sti- muloimaan kaikkia aisteja. Mortonin seuraava inovaatio oli hänen 1960 kehittelemä Te- lesphere Mask (Kuva 1), joka oli ensimmäinen HMD (head mounted display). (Virtual reality society 2017a.)



Kuva 1. Telesphere Mask. (Virtual reality society 2017.)

Vaikka virtuaalitodellisuus on ollut olemassa jo pitkään, on se todella tullut kuluttajien olottuville vasta muutaman viime vuoden aikana. Tähän on vaikuttanut niihin tarvittavien

laitteiden hinnanlasku. HMD-laitteita on saatavilla aina muutamien eurojen hinnasta alkaen, mistä johtuen virtuaalitodellisuuden käyttö on vahvassa kasvussa. (Virtual reality society 2017b.)

HMD-laitteilla käytettävä virtuaaliympäristö on pohjimmiltaan ohjelma tai peli, joka ohjelmoijan on kehitettävä. Tätä tarkoitusta varten markkinoilta löytyy valmiita kehitysympäristöjä, joita kutsutaan pelimoottoreiksi. Näistä kaksi yleisintä ovat Unity ja Unreal Engine (Sparks 2017).

2.2.1 Unity

David Helgason perusti Unityn vuonna 2004, josta lähtien se on kasvanut pienestä studiosta yhdeksi maailman suosituimmaksi pelimoottoriksi. Sillä on 45 %:n osuus pelimoottorimarkkinoista ja yli 4,5 miljoonaa rekisteröitynyttä kehittäjää. (iGlobe Partners 2019.) Unitysta on tullut suosittu erityisesti mobiili ja VR sekä AR (Augmented Reality) -kehittäjien keskuudessa, jonka takia yli 50% mobiili- ja 60% VR sekä AR -sovelluksista on kehitetty sillä (Rowe 2019).

Unityn vahvuuksia ovat sen opetuskirjasto ja käyttäjien tekemät tutoriaalit (Sparks 2017), kattava kauppapaikka josta on saatavilla 3D malleja ja SDK:ita (Rowe 2019), sekä yleinen käytön helppous. Tämän takia sitä käyttävät varsinkin yhden miehen yritykset ja pienet kehitystiimit. (Creative Bloq 2019.) Se tukee myös integraatioita kaikkien suurempien VR ja AR laitteiden kanssa (Rowe 2019). Unity on käytössä toimeksiantajan asiakasprojektissa ja myös muissa projekteista, joten tästä syystä opinnäytetyö on tehty käyttäen tätä pelimoottoria.

2.2.2 Unreal Engine

Epic games aloitti kehittämään Unreal peliä vuonna 1998, joka johti Unreal Enginen ensimmäisen version luontiin. Pian Unrealin ensimmäisten esittelyjen jälkeen, muut studiot

alkoivat kiinnostua pelimoottorista ja pyysivät lupaa käyttää sitä omien peliensä tekemiseen. Tämä johti pelimoottorin lisensointiin ja sen käytön laajentumiseen pelialalla. (Thomsen 2012.)

Unreal Enginen suurimpana etuna pidetään sillä luotujen sovelluksien visuaalisesti korkea laatu. Tästä syystä syystä Unrealin pelimoottoria käytetään usein suurissa peleissä ja isojen studioiden tuotannossa. (Creative Bloq 2019.)

2.2.3 CAD-malleista pelimoottoreihin

Koska virtuaalitodellisuuden käyttö on aiemmin keskittynyt peli ja viihdemaailmaan, muun muassa animointiin käytetyistä 3D-mallinnusohjelmista malleja voidaan helposti tuoda pelimoottoreiden käytettäväksi. Virtuaalitodellisuuden käyttö on kuitenkin alkanut levitä myös useille teollisuuden aloille (Easy Render 2019). Virtuaalitodellisuuden avulla visualisoimalla, tuotantoprosesseja voidaan helpommin tehostaa koskematta linjastoihin ja tuotteita optimoida jo kehityksen aikaisessa vaiheessa (ARTS experts GmbH 2019). Teollisilla aloilla 3D-mallintamista on kuitenkin tehty CAD-ohjelmilla, sillä niiden alkuperäinen tarkoitus on ollut niiden tuotannollinen suunnittelu. Ongelmaksi muodostuu, että toisin kun 3D-mallinnusohjelmilla, CAD-ohjelmilla tehtyjä kuvia ei voi suoraan käyttää pelimoottoreissa ja kääntämiseen vaadittava toimintatapa on pidetty monimutkaisena ja aikaa vievänä (Baldwin 2017).

2.3 CAD-mallin tuomisprosessi Unity-pelimoottoriin

Unity ei suoraan tue CAD-tiedostomuotoja, koska CAD-mallien geometria koostuu NURBS-käyristä. NURBS-käyrät ovat tarkkojen pintojen muotojen visualisointiin käytettäviä matemaattisia käyriä, joiden renderöinti reaaliaikaisesti olisi mahdotonta. Tästä syystä CAD-malli täytyy kääntää 3D-mallinnusohjelmalla polygongeometriaksi. (Klementtinen 2012, 39.)

Polygonit ovat suorasisivuisia muotoja, jotka määrittyvät kolmiulotteisten pisteiden, eli verteksien ja niitä yhdistävien suorien reunaviivojen, eli edgejen avulla (Autodesk 2015). Käännösprosessin jälkeen se täytyy vielä viedä Unity-pelimoottorin tukemaan muotoon.

Pelkkä käännösprosessi ei kuitenkaan aina riitä. CAD-malli saattaa olla vielä liian raskas tuotavaksi virtuaaliodellisuuteen. CAD-malleissa on suoraan tuotuna helposti satojatusiansia verteksejä ja esimerkiksi mobiililaitteilla verteksimäärä pitäisi pystyä pitämään alle 100 000:n. Moderni pöytäkone pystyy kerrallaan käsittelemään miljoonia verteksejä, mutta virtuaalialustalle renderöidessä kaikki piirretään tuplana, koska molemmille silmille on oma kameransa (Unity Docs 2020; Porter 2017). Tästä syystä CAD-mallia pitää vielä optimoida 3D-mallinnusohjelmassa.

Optimointi käsitteenä on melko laaja ja esimerkiksi tässä opinnäytetyössä tehty geometrioiden yhdistäminen olisi osa optimointia, mutta koska vaiheet erotellaan toisistaan, optimoinnilla tarkoitetaan pääsääntöisesti vain polygon tai verteksimäärän vähentämistä.

2.3.1 Pixyz

PixyzPlugin on Unityn liitännäinen, jolla saa tuotua lähes minkä tahansa CAD-mallin suoraan Unityyn. Liitännäisellä voi sen lisäksi muun muassa optimoida mallia, kääntää mallin normaaleja, siirtää objektin origoa tai luoda Level of Detail (LOD) versiot mallista. Liitännäisen sääntömoottoria (Rule Engine) käyttäen näitä prosesseja pystyy myös automatisoimaan. (Pixyz plugin 2020)

Pixyz -brandin alla on myös muita 3D datan käsittelyyn erikoistuneita ohjelmia, kuten PixyzStudio, joka on erillinen ohjelmansa, jossa on PixyzPluginin toiminnallisuuksien lisäksi lisätoimintoja ja on suunniteltu mallien käsittelyyn. (Martin E. 2019)

PixyzPluginin lisenssimaksu on 1 000 €/v (83,3 €/kk) ja PixyzStudion lisenssi maksaa 1 800 €/v (150 €/kk) ja molemmissa on mahdollista ostaa lisenssi vain vuodeksi kerrallaan. Tästä syystä, sekä sen hetkisestä tarpeesta tuoda CAD-malleja pelkästään Unityyn, päätimme ottaa testikäyttöön PixyzPluginin.

CAD-malleja pystyy tuomaan suoraan Unity-pelimoottoriin myös muita liitännäisiä käyttäen, mutta Pixyz oli ainoa, jossa oli edellämäinittuja ominaisuuksia, joten tilanteeseen sopivia varsinaisia muita vaihtoehtoja ei ollut.

2.3.2 Muut vaihtoehdot tuoda CAD-malli VR-ympäristöön

CAD-malleja pystyy katselemaan myös käyttämättä pelimoottoria. Joissain CAD-suunnitteluohjelmissa ja 3D-mallinnusohjelmissa on omat ominaisuudet mallin katseluun virtuaalilaseilla vaihtelevilla interaktion tasoilla. Esimerkiksi Solidworksissä ja Autodeskin 3D-mallinnusohjelmissa itsessään on mahdollisuus katsella photorealistisia renderöintejä CAD-mallista. On muitakin vaihtoehtoja, kuten Magic Leapillä toimiva Onshape, jolla pystyy katsomaan CAD-mallia AR-laseilla ja HTC Viveä käyttävä Siemens NX, jolla pystyy tuomaan CAD-mallin suoraan virtuaaliympäristöön ja niitä pystyy joissain määrin myös käsittelemään. (Keane 2019.)

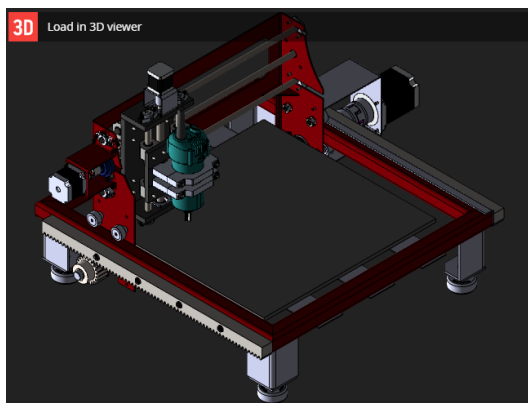
VR toiminnallisuudet ja interaktiot ovat näissä kuitenkin melko rajallisia, joten toimeksiantajan asiakasprojektissa vaadittuja toiminnallisuuksia ei näillä vaihtoehdoilla pysty luomaan. Tästä syystä näitä vaihtoehtoja ei tutkittu enempää.

3 TAPPAUS ADE OY

Toimeksiantaja Ade Oy aloitti pitkäkestoisen asiakasprojektin, jonka tarkoituksena on rakentaa virtuaalinen esittely-ympäristö tuotantolinjastosta ja niissä käytetyistä laitteista sekä moduuleista. Projektissa on ns. aula omana Unityn scenenä, josta pääsee katselemaan eri esittelyjä, jotka ovat myös omia scenejään Unityssä. Kun asiakas haluaa uuden ympäristön esiteltäväksi, luodaan vanhan pohjalta uusi scene Unityyn, ja vaihdetaan esiteltävä tuote uuteen asiakkaan CAD-mallin pohjalta tehtyyn malliin. Tämän jälkeen lisätään aulaan painike, jota painamalla tätä uutta linjaa pääsee katselemaan. Koska uusi ympäristö luodaan vanhan pohjalta, VR:n toiminnallisuuksia eli liikkumista ja interaktiivita tai projektin asetuksia ei tarvitse säätää tai ohjelmoida uudestaan.

Tehtävänä oli mallien tuontityöstä vastaaminen ja sen mukana olemassa olevien työtapojen kehittäminen. Projektissa oli aiemmin luotu toimintatapa, jonka mukaan asiakkaan olemassa olevia CAD-malleja saatiin tuotua VR-ympäristöön. Tässä toimintatavassa on kuitenkin kohtia, jotka vievät hyvin paljon aikaa, jotta lopputulos on riittävän hyvälaatuisen ja mahdollisimman kevyt VR-ympäristössä. Aikataulut ovat usein hyvin tiukat, joten pienikin ajan säästäminen jossain toimintatavan vaiheessa antaa lisää aikaa muihin vaiheisiin, joka puolestaan parantaa tuotteen laatua kokonaisuudessa.

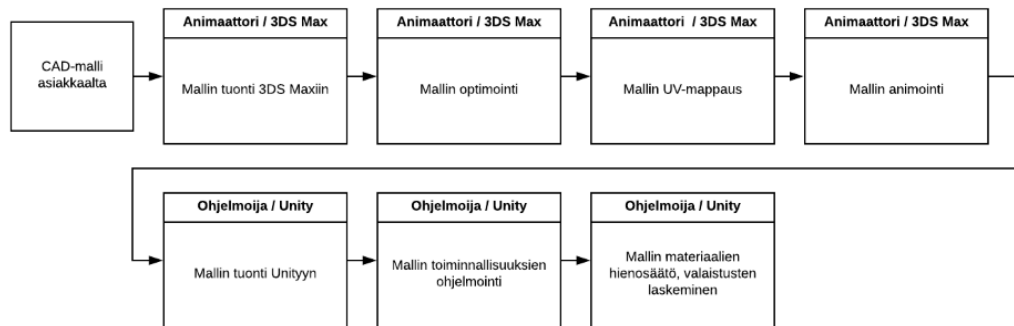
Tässä opinnäytetyössä käytän esimerkeissä GrabCad-sivustolta ladattua ilmaista mallia, koska varsinaiset asiakasprojektin mallit ovat vaitolovelvollisuuden alaisia. (Kuva 2.) Esimerkissä on käytetty Unityn versiota 2019.1.3f1 ja 3ds Max 2020.



Kuva 2. Esimerkkimalli. (GrabCad.)

3.1 Lähtötila

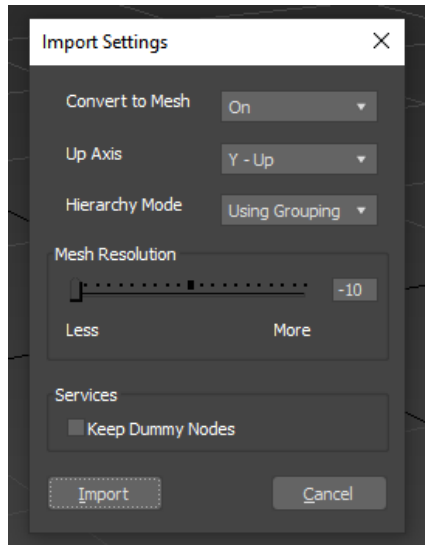
Tapaus alkoi tilanteesta, jossa projekti oli jo aloitettu ja workflow eli toimintatapa luotu. Työkaluina käytettiin 3D-mallinnuksessa 3ds Maxia ja pelimoottorina Unityä. Ohjelmointikielenä Unityssä käytetään C#. Asiakkaalta saatu CAD-malli tuodaan 3ds Maxiin, jossa se ensin optimoidaan polygonimäärän laskemiseksi, järjestetään hierarkiat animointia varten ja yhdistetään geometriaa materiaalin mukaan, jonka jälkeen malli UV-kartoitetaan, jotta tekstuurit menevät oikein paikoilleen, ja luodaan animaatiot. Seuraavaksi malli viedään fbx-muotoon, jotta sen saa tuotua Unityyn. Malli tuodaan Unityyn ja ohjelmoidaan sille toiminnallisuudet, esimerkiksi laukaisimet, milloin animaatiot täytyy käynnistää. Koska 3ds Maxin materiaalit eivät suoraan käänny Unityyn, ohjelmoijan pitää vielä hienosäätää mallin materiaalit ja scenen valaistukset tapauskohtaisesti. (Kuva 3.)



Kuva 3. Olemassaoleva toimintatapa ja työnjako.

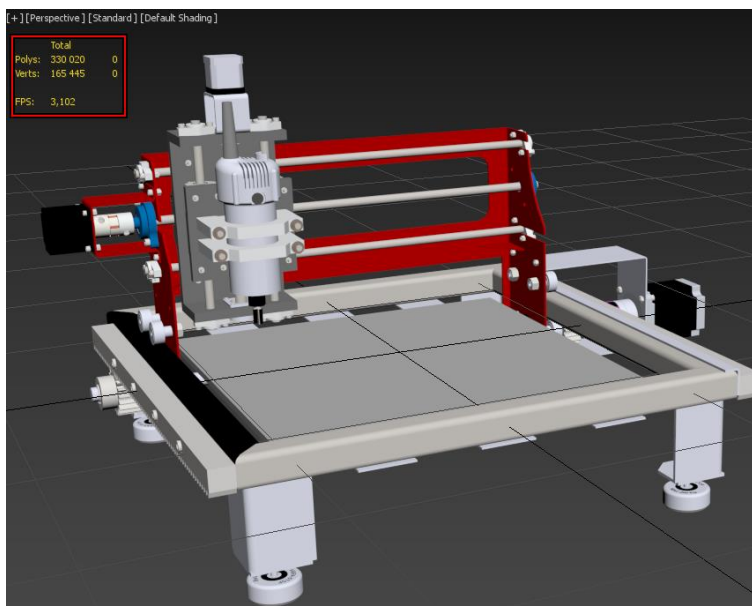
3.1.1 Mallin tuonti 3ds Maxiin ja optimointi

Asiakkaalta saatu CAD-malli tuodaan ensin 3ds Maxiin sen tuontityökalulla, jossa pystyy säätämään mallin hierarkian ja tarkkuuden asetuksia. Asetuksista voi myös määrittellä yläakselin. (Kuva 4.)



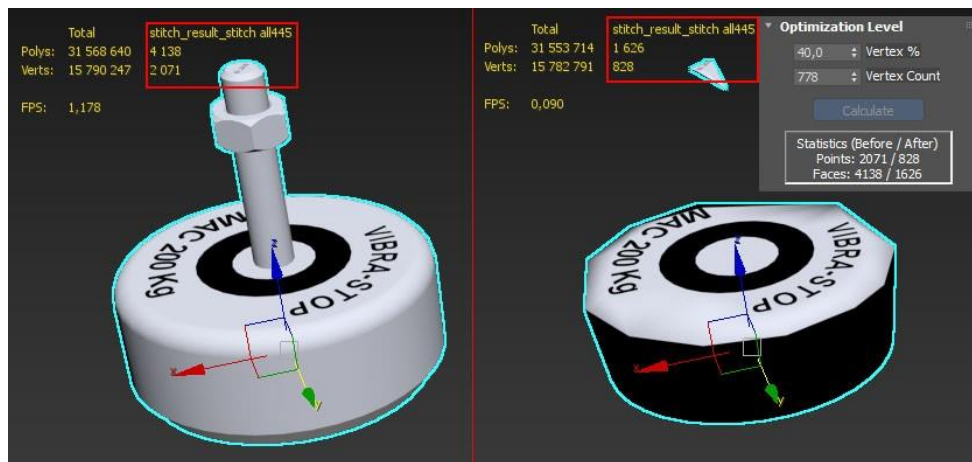
Kuva 4. 3ds Maxin tuonti-ikkuna.

Kun malli on tuotu halutuilla esiasetuksilla 3ds Maxiin, se on kuitenkin usein liian raskas suoraan vietäväksi VR-sovellukseen. Tästä syystä sitä on optimoitava eli geometriaa täytyy yhdistellä ja polygonimäärää vähentää. Kuvassa 5 nähdään, että minimiasetuksilla tuodussa CAD-mallissa on vielä yli 330 000 polygonia, mikä on ongelma kun samankaltaisia malleja on useita. Esimerkiksi jos Unityyn tuodaan kokonaisia tuotantolinjastoja ja ympäristöjä.

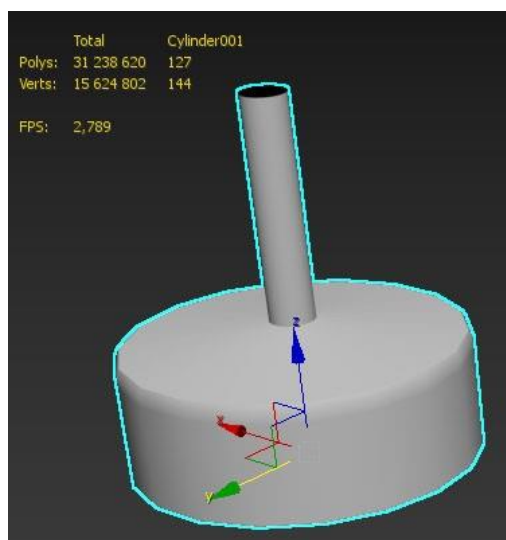


Kuva 5. Esimerkkimallin polygon- ja verteksimäärä.

Optimointiin on useita eri tapoja ja algoritmeja, joita voidaan käyttää hyödyksi, mutta optimointityökalut täytyy valita tapauskohtaisesti. 3ds Maxissa on Optimizer ja ProOptimizer-modifierit, joilla saa potentiaalisesti nopeastikin vähennettyä geometrian polygonmäärää. Tämä tosin helposti hajottaa mallin geometriaa, joten niitä ei pysty kaikissa tapauksissa käyttämään halutun optimoinnin tason saavuttamiseksi. (Kuva 6.) Joitain vähemmän tärkeitä yksityiskohtia mallista on helppo ja nopea mallintaa uudelleen tai jättää pois kokonaan. (Kuva 7.)

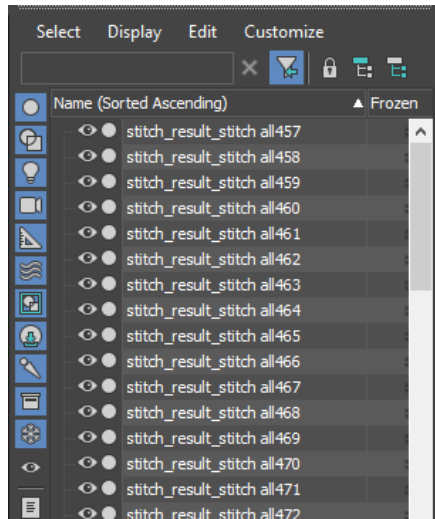


Kuva 6. Optimointimodifier voi hajottaa objektin geometriaa.



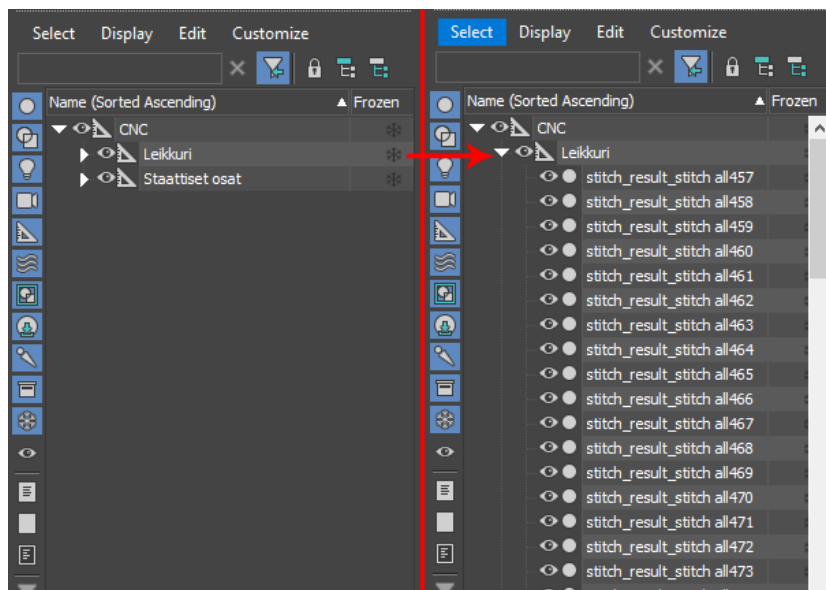
Kuva 7. Uudelleen mallinnettu pöydänjalka.

Mallin polygonmäärän vähentämisen jälkeen tarkastellaan sen hierarkiaa. Staattisissa malleissa joita ei animoida, hierarkialla ei ole niin suurta merkitystä. Jos malliin tulee animaatioita, hierarkian järjestely helpottaa niin animaattorin kuin ohjelmoijankin työtä. Joissain tapauksissa se on myös toiminnan kannalta pakollista. Esimerkkimalli animoidaan ja sen hierarkia on Kuvan 8 mukainen, joten sitä täytyy selkeyttää ja varmistaa, että oikeat osat liikkuvat animaatioissa.



Kuva 8. Esimerkkimallin hierarkia tuonnin jälkeen.

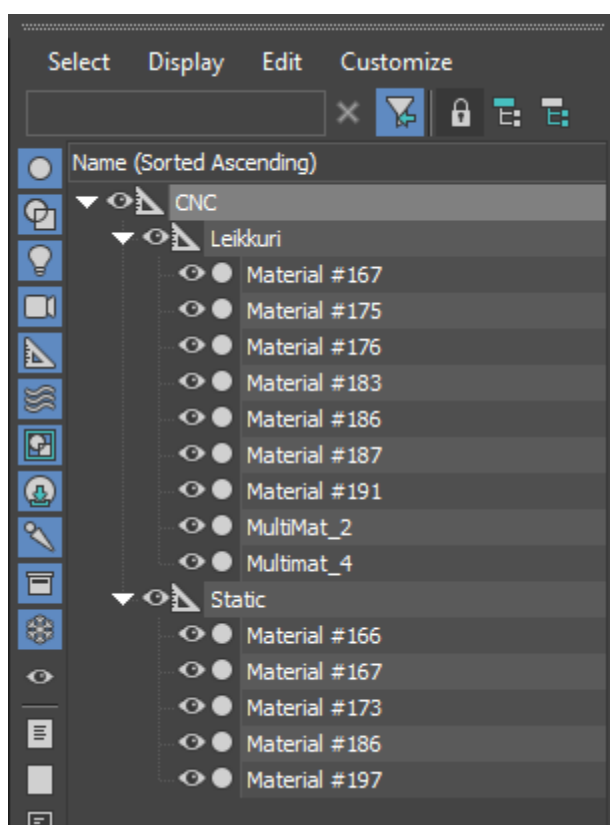
Koska esimerkin animaatio tulee olemaan hyvin yksinkertainen (vain yksi liikkuva osa), hierarkian voi jakaa yksinkertaisimmillaan staattisiin ja liikkuviin osiin. (Kuva 9.)



Kuva 9. Uudelleenjärjestetty hierarkia.

Hierarkiaan luodaan tyhjät Dummy-objektit, joiden lapsiobjekteiksi geometriat on asetettu. Täten mallissa voi animoida pelkästään yhtä objektiä ja animaatiot eivät hajoa, jos mallien geometriaan tulee jälkikäteen jotain muutoksia. Myös Unityn puolella ohjelmoija pystyy helposti merkkamaan oikeat osat staattisiksi valojen laskua varten. Seuraavaksi yhdistetään geometrioita materiaalin mukaan Unityn niputusta varten.

Osien hierarkinen erottelu helpottaa myös geometrioiden yhdistämistä. Sen sijaan että tässä vaiheessa kävisi yksitellen osia läpi, on liikkuvat osat helppo eristää muusta mallista. Kun haluttu osa on eristetty, voidaan siitä ottaa materiaali materiaalieditorin Pick Material From Object -toiminnolla, jonka jälkeen voidaan valita kaikki geometria, joka käyttää kyseistä materiaalia valitsemalla kontekstimenusta Select by Material. Kun oikeat osat on saatu valittua, ne on jälleen helppo eristää, jonka jälkeen ne saadaan yhdistettyä toisiinsa objektin "Modify" valikosta käyttäen "Attach List" -toimintoa. Kuvassa 10 näemme materiaalien mukaan yhdistetyn mallin hierarkian.



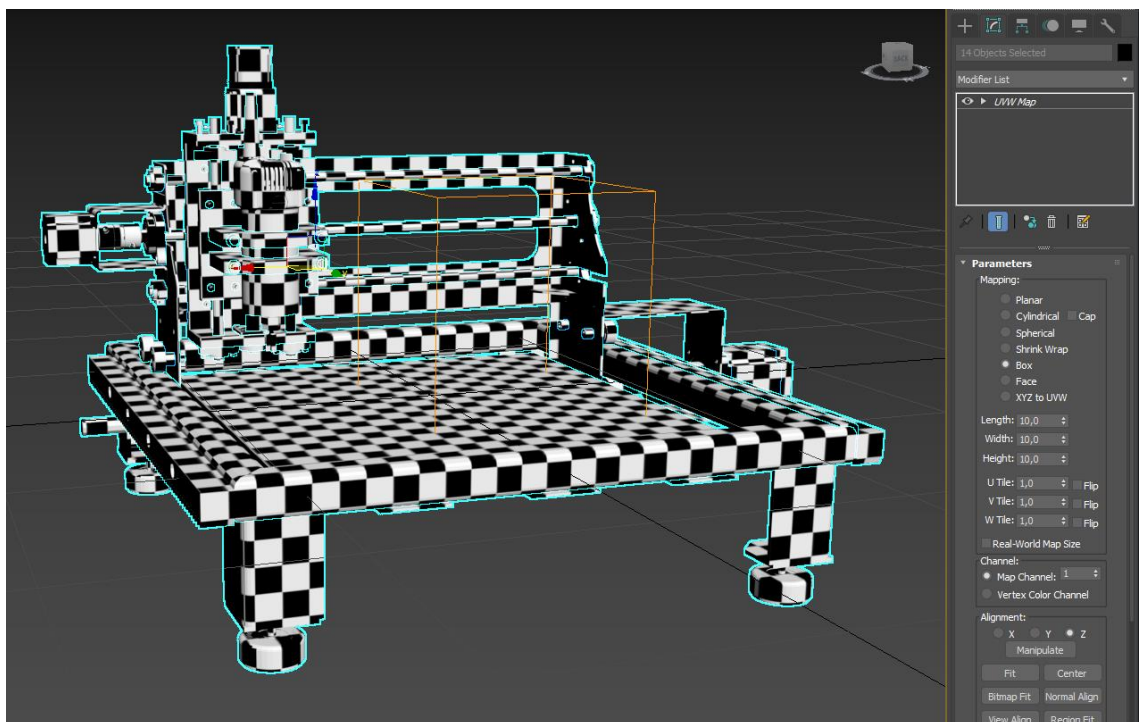
Kuva 10. Hierarkia geometrioiden yhdistelyn jälkeen.

Haluttujen osien yhdistämisen jälkeen malli on valmis UV-kartoitusta varten. Jos mallin UV-kartoittaa ennen kuin geometria on muokattu valmiiksi, kartat voivat hajota. Tästä

syystä on tärkeää, että optimoinnit ja muokkaukset tehdään ennen kuin malli UV-kartoitetaan.

3.1.2 Mallin UV-kartoitus

Koska asiakasprojektin malleissa on käytetty pääsääntöisesti jo olemassa olevia materiaaleja, joissa on saumattomat tekstuurit, UV-karttoja ei ole ollut tarvetta luoda kovin tarkasti. 3ds Maxin automaattiset UV-kartoitustyökalut ovat luoneet riittävän laadukkaita tuloksia. Tarkoituksena on ollut, että samoja materiaaleja pystyisi käyttämään mahdollisimman monessa eri mallissa. Tästä syystä esimerkissä ei perehdytä tarkemmin UV-kartoitukseen, vaan käytetään 3ds Maxin UVW Map -modifieria. Kuvassa 11 nähdään shakkiruutukuviollisen tekstuurin asettuminen malliin sekä kyseisen modifierin asetukset.



Kuva 11. Shakkiruutumateriaali esimerkimallissa.

Mallin UV-kartoituksen jälkeen voidaan siirtyä animointiin. Asiakasprojektissa malleilla on hieman monimutkaisemmat animaatiot, mutta kuitenkin pääsääntöisesti x, y, ja z akselin lineaarisia liikkeitä. Esimerkimallin leikkuri animoidaan kulkemaan vasemmasta laidasta oikeaan.

3.1.3 Mallin animointi ja vienti

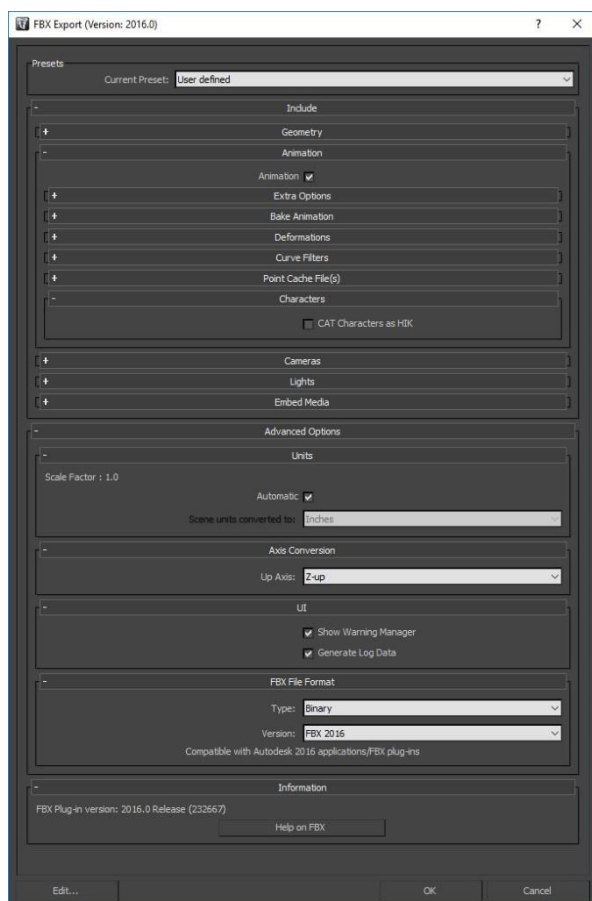
Projektissa osa asiakkaalta tulleista malleista ovat staattisia, joita ei tarvitse animoida. Osa malleista ja kokonaisuuksista kuitenkin sisältää animoituja elementtejä, esimerkiksi tuotantolaitteistoa, jotka liikkuvat samaan tapaan kuin oikeat laitteetkin.

3ds Maxissa avainkehysanimointi tapahtuu sille luotujen työkalujen avulla. Avainkehysanimaatioissa asetetaan avaimia (key) haluttuihin kohtiin aikajanalla. Avaimet määrittelevät muun muassa objektin sen hetkisen position tai orientaation. Kun avaimia on asetettu, objektin animaatio-ohjain osaa interpoloida objektin liikkeen avaimien välillä. Esimerkkimallin animaatioissa asetetaan leikkuriobjektin aloituspisteen avain animaation alkuun, ja kohdistetaan aikajana 60. avainkehysten kohdalle. Seuraavaksi siirretään leikkuriobjekti liikeratansa oikeaan laitaan ja asetetaan uusi avain. 3ds Maxin animaatiot toteutuvat oletusasetuksilla 30 fps (frame per second) nopeudella, joten leikkuri liikkuu nyt aloituskohdastaan liikeratansa oikeaan laitaan kahdessa sekunnissa. Aikajana kohdistetaan 65. avainkehysten kohdalle, kopioidaan edellinen avain ja liitetään se aikajanalle valittuun kohtaan. Seuraavaksi kopioidaan ensimmäisen avain ja liitetään se aikajanelle 125. avainkehysten kohtaan ja yksinkertainen animaatio on valmis. Kuvassa 12 nähdään 3ds Maxin avainkehysanimaatiotyökalun ja animaation avainkehysten sijainnit aikajanalla.



Kuva 12. 3ds Maxin aikajana.

Mallin valmistumisen myötä malli vietään fbx-muotoon 3ds Maxin vientityökalua käyttäen. Vientiasetuksista voi vielä säätää tarvittavia asetuksia, kuten yksikköjen määrittely metriseen tai imperiaaliseen järjestelmään tai halutaanko animaatiot viädä myös mallin mukana. (Kuva 13.)



Kuva 13. 3ds Maxin vienti-ikkuna.

3.1.4 Mallin tuonti Unityyn

Unityn puolella asiakasprojektissa on jo valmiiksi luotu ympäristö ja VR-toiminnallisuudet, jotka kopioidaan aina uuteen kokonaisuuteen. Tämä mahdollistaa sen, että kaikkia asetuksia ei tarvitse säätää aina uudestaan, kun uutta linjastoa tehdään. Tästä syystä myöskään tässä opinnäytetyössä ei perehdytä niihin, vaan ainoastaan mallin tuomiseen Unityyn ja sen käsittelyyn Unityssä.

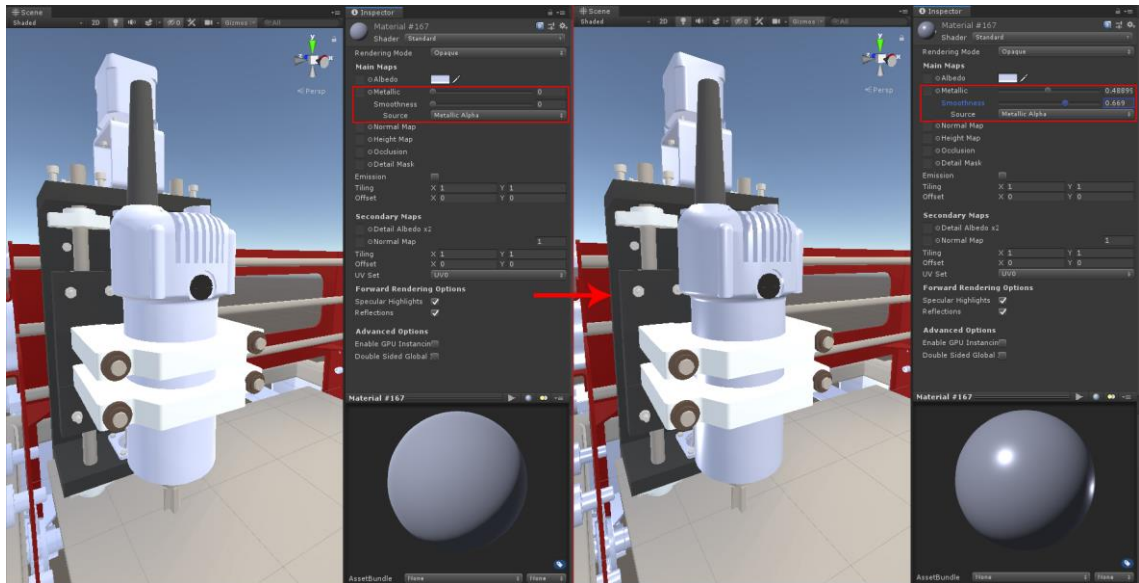
Kun malli on tuotu Unityyn, pystyy säätämään sen tuontiasetuksia (Kuva 14). Tuontiasetuksista pystyy vielä tekemään säädöksiä muun muassa mallin skaalaan. Asetuksissa pystyy vielä kääntämään käytetyt yksiköt tai laskemaan mallin normaaleja uusiksi. Asetusten Materials-osiosta pystyy poimimaan malliin mahdollisesti liitetyt tekstuurit ja materiaalit. Mallille pystyy myös asettamaan vanhojen materiaalien tilalle jo olemassa olevia materiaaleja, kuten asiakasprojektissa on tehty. Esimerkkiprojektissa ei ole valmiita materiaaleja mallille, joten ne poimittiin mallista.



Kuva 14. Unityn tuontiasetukset.

Toisin kuin monissa muissa ohjelmissa, näitä tuontiasetuksia pystyy säätämään myös jälkikäteen, jos sille on tarvetta. Esimerkiksi jos mallin normaaleissa tai pehmennyksessä (smoothing) ilmenee jotain pientä ongelmaa, ne pystyy laskemaan uusiksi sen sijaan, että käyttäisi mallista tuotua dataa.

Kun Unityyn tuo mallin ja sille luodaan materiaalit, on niiden metallisuus (Metallic) ja pehmeys (Smoothness) -arvot oletuksena 0. Tästä syystä niitä täytyy tarpeen mukaan säätää (Kuva 15). Mallissa on paljon metallisia osia, joten niiden täytyy heijastaa valoa ja materiaalista riippuen myös ympäristöä.



Kuva 15. Metallic ja smoothness arvon muuttamisen vaikutukset.

Asiakasprojektissa on käytettävät materiaalit valmiina, joten materiaaleja tarvitsee poimia tai säätää vain tapauksissa, joissa esiteltävässä laitteessa käytetään jotain uutta materiaalia, joka on ollut harvinaista.

3.2 Työprosessin kehittäminen

Tutkimustyö aloitettiin syksyllä 2018 ja silloin vaihtoehdot olivat erittäin niukat ja lähes kaikki lähteet suosittelivat CAD-mallin tuontia ensin 3D-mallinnusohjelmaan ja toimimaan samalla toimintatavalla, joka käytössä jo oli. Vastaa tuli Unityn blogikirjoitus jossa kerrottiin Pixyz -liitännäisestä, jolla voisi tuoda CAD-malleja suoraan Unityyn. Liitännäisellä pystyisi myös muun muassa optimoimaan ja UV-kartoittamaan mallit. (Martin E. 2018)

3.2.1 Pixyz-liitännäisen ensivaikutelmat

Pixyz-liitännäisestä testattiin ensin sen kokeiluversiota, jonka sai käyttöön seitsemäksi päiväksi. Pienempien ja yksinkertaisempien mallien tuonti olikin erittäin nopeaa. Tuonti kesti minutteja, vaikka niille luotiin UV-kartat teksturointia ja valokarttoja varten. Muutama monimutkaisemmassa testimallissa kuitenkin prosessi kesti hyvin kauan tai Unity

kaatui. Geometrian yhdistely ja automaattinen Unityn prefabien luominen sujui lähes ongelmitta, mutta pian ilmeni ongelma, että liitännäinen ei tue Unityn kumoamis- ja uudelleen teko (undo, redo) -toimintoja. Tämä tarkoitti sitä, että jos geometrian yhdistelyssä tai mallin käsittelyn muussa vaiheessa tekee virheen, joutuu mallin tuomaan uudestaan Unityyn ja aloittamaan prosessi alusta. Tämä kierrettiin luomalla mallista varmuuskopioita, jotka kasvattavat projektin kokoa.

Optimoinnin taso tuntui riittävältä PC:llä suoritettaviin VR-sovelluksiin, jos CAD-malleja ei ole liian montaa näkyvissä samaan aikaan. Mobiili VR-sovelluksissa mallit ovat vielä liian raskaita, joten malleja täytyy vielä optimoida lisää 3D-mallinnusohjelmassa. Myös joissain tilanteissa optimointialgoritmi tuntui hajoittavan etenkin pyöreitä muotoja helpommin, joten optimoinnin määrää piti joko vähentää, tai korjata muodot mallinnusohjelmassa.

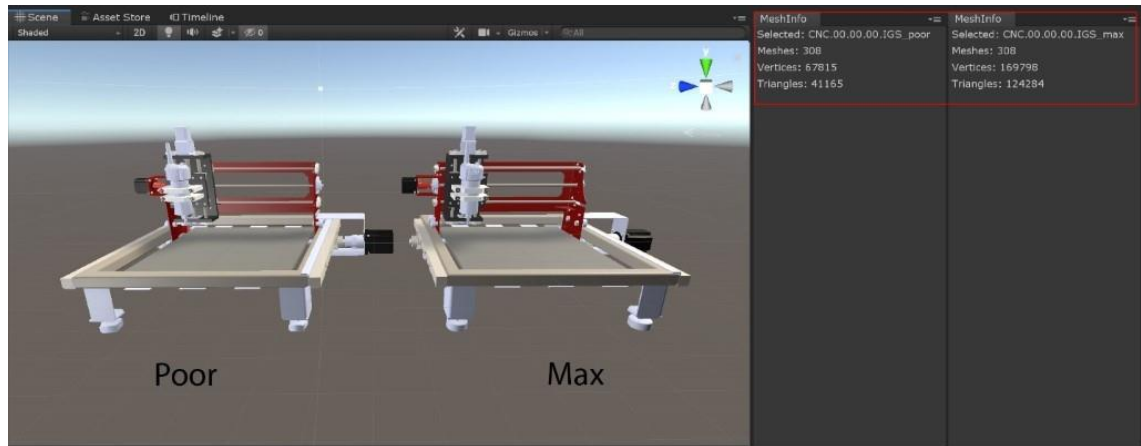
3.2.2 Mallin tuonti Unityyn Pixyz-liitännäistä käyttäen

Asiakkaalta saatu CAD-malli tuodaan Unityyn suoraan Pixyzin Unityliitännäistä käyttäen. Kun Unityssä avaa Pixyz-valikon ja valitsee "Import Model", aukeaa kuvan 16 mukainen tuonti-ikkuna. Tässä vaiheessa voi valita, millä laadulla geometriat tuodaan Unityyn, luodaanko mallille UV-kartat niin tekstuureille kuin valokartoille ja säätämään niiden resoluutiota sekä kuinka tiiviisti kartat pakataan (padding). Jos CAD-mallissa on liitettyä metadataa, voi senkin tuoda halutessaan ja automaattisesti luoda LOD-versiot mallista.



Kuva 16. Pixyz-liitännäisen tuonti-ikkuna.

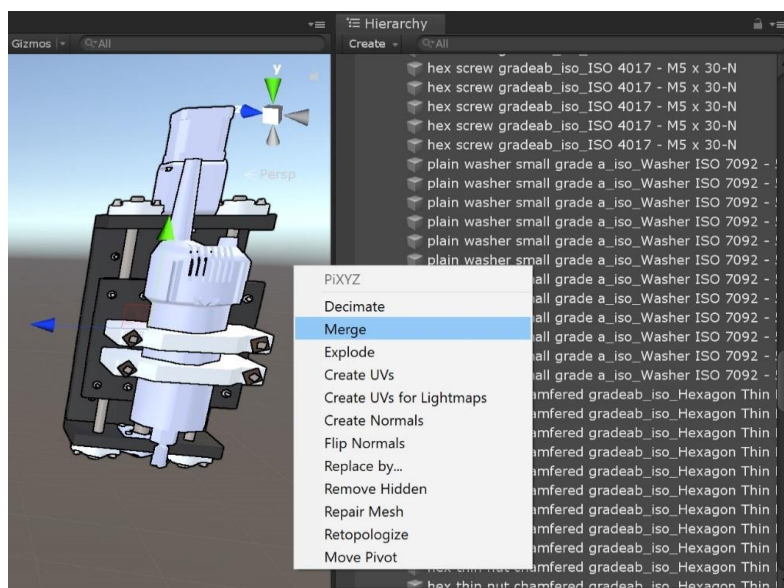
Esimerkkimalli tuotiin kuvan 16 mukaisilla asetuksilla Unityyn. Vertailun vuoksi esimerkkimalli tuotiin myös maksimilaadulla. Käytössä olevasta tietokoneesta, CAD-mallista ja valituista asetuksista riippuen tuonti saattaa kestää kauankin, mutta esimerkkimallin tuonnissa kesti noin 3 minuuttia molemmilla laaduilla. Kuvassa 17 nähdään esimerkkimallin polygonimäärät molemmilla asetuksilla.



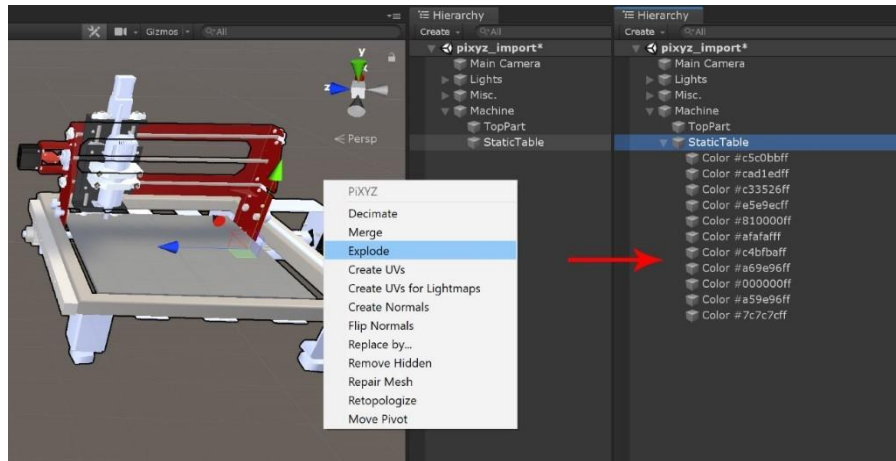
Kuva 17. Esimerkkimallin polygon-määrät eri laatuasetuksilla.

3.2.4 Mallin valmistelu

Kuten 3ds Maxissakin, täytyy myös Unityssä mallin hierarkia säätää kuntoon animoinnin helpottamiseksi sekä yhdistää geometriaa Unityn niputusta varten riippuen siitä, miten alkuperäinen tiedosto on luotu. Jos malleja on käytetty simulointia varten, voi niissä usein olla hierarkiat asetettu jo valmiiksi oikein. Geometrian yhdistäminen Pixyzillä tapahtuu valitsemalla halutut objektit, avaamalla valikko hiiren oikealla napilla ja valitsemalla valikosta "Merge" (Kuva 18). Unityn niputusta varten mallin saa eroteltua materiaalin mukaan valitsemalla valikosta "Explode" (Kuva 19).



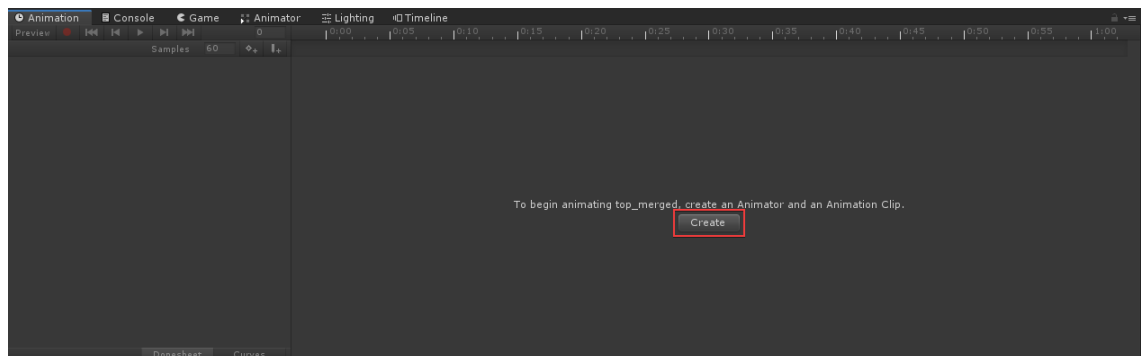
Kuva 18. Merge kontekstivalikossa.



Kuva 19. Explode kontekstivalikossa.

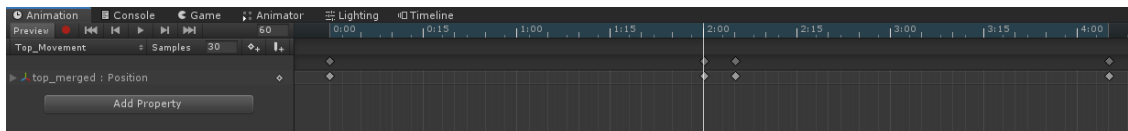
3.2.5 Mallin animointi Unityssä

Jotta Unityssä voi objektia animoida, täytyy sillä olla Animator-komponentti. Animator-komponenttiin täytyy lisätä referenssi animaatio-ohjaimeen (Animation Controller). Kun nämä löytyvät, voidaan luoda uusi animaatioklippi. Jos objekti on valittuna animaatioikkunan ollessa auki, eikä näitä komponentteja objektista tai sen vanhemmista (parent) löydy, voi Unityssä luoda ne automaattisesti painamalla Create-painiketta (Kuva 20).



Kuva 20. Animaatiokomponenttien luonti.

Avainkehysanimaation luonti tapahtuu hyvin samalla lailla kuin 3ds Maxissa. Avainkehysä asetetaan aikajanelle (tässä tapauksessa objektin positio) ja moottori interpoloi objektin position kehysten välillä. Aikajanassa eroavaisuutena 3ds Maxiin verrattuna on se, että oletuksena aikajana näkyy sekunteina kehysten sijaan. Myös animaation oletusnopeus on 60 fps. Kuvassa 21 nähdään aikajana ja avainten asettelu.



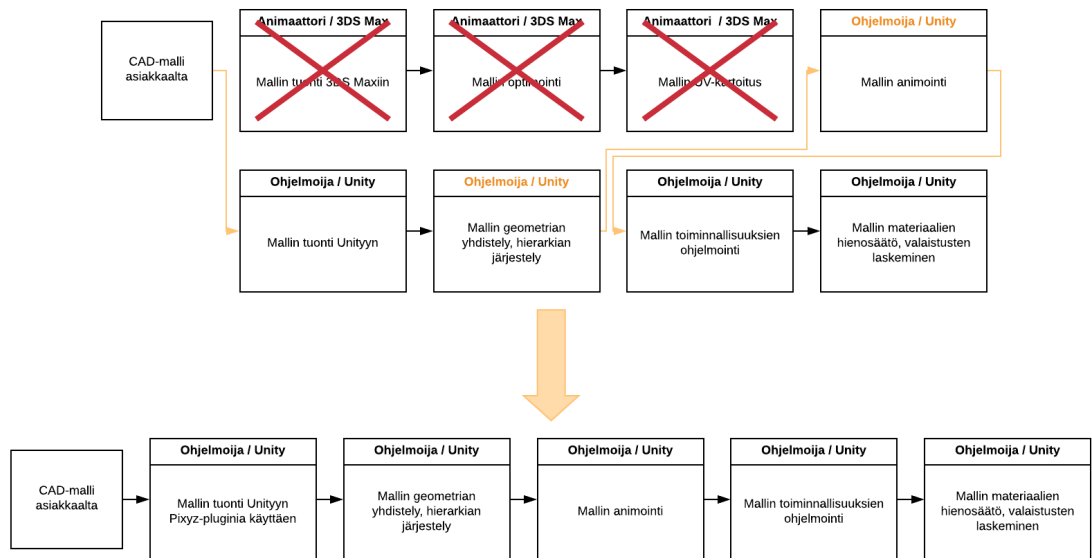
Kuva 21. Unityn aikajana.

Mallin animaation valmistuttua workflow jatkuisi samalla tavalla kuin aiempikin. Ohjelmoijan pitäisi vielä ohjelmoida mallille toiminnallisuudet, ja laskea valaistus sekä tarvittaessa hienosäätää materiaaleja.

3.2.6 Lopputulos

Ideaalisessa tapauksessa tuotava CAD-malli on jokin staattinen mekaaninen laite tai tuote, jossa on mahdollisimman vähän pyöreitä muotoja ja se tuotaisiin projektiin, jossa VR-toiminnallisuudet eivät kuluttaisi niin paljoa tietokoneen resursseja. Tällöin malleja voisi joko tuoda tarkemmalla resoluutiolla, tai näyttää niitä useampia kerrallaan. Suorituskyvyn vaikutukset näkyvätkin asiakasprojektissa isompien linjastojen kohdalla, jos linjastoa tarkastelee kuvakulmasta, jossa koko linjasto tai iso osa linjastosta näkyy kerralla. Valaistuksen laatu ja varjojen piirto kuitenkin vaikuttavat enemmän suorituskykyyn kuin mallien polygonien määrä.

Uudessa toimintatavassa mallin UV-kartoitus on jätetty kokonaan pois, koska Pixyz-liitännäinen tekee sen automaattisesti. Mallin geometriankin pystyisi staattiseen malliin yhdistämään automaattisesti materiaalin mukaan. Jos osa mallista animoidaan, tarvitsee liikkuvat osat eristää ennen yhdistämistä. Animointiosuuden siirtyminen ohjelmoijalle myös edellyttää ohjelmoijalta jonkin tason tuntemusta avainkehysanimoinnista. Kuvassa 22 nähdään muutokset vanhaan toimintatapaan sekä uusi toimintatapa selkeytettyinä.



Kuva 22. Muutokset aiempaan toimintatapaan sekä uusi toimintatapa.

Unityn Pixyz-liitännäisellä pystyykin monissa tapauksissa säästämään aikaa huomattavasti. Joissain tilanteissa pystyy jättämään manuaalisen optimoinnin ja UV-kartoituksen pois, ja sen myötä korvaamaan animaattorin osuuden työstä kokonaan. Mallien käsittely Unityssä puolestaan lisäksi ohjelmoijan työmäärää. Animaattorin pois jääminen projektista suoraviivaistaisi projektin etenemistä. Tämä toimintatapa otettiin käyttöön asiakasprojektiin ja toimeksiantaja pystyi resursoimaan animaattorin toiseen projektiin tai työtehtävään. Liitännäisen 1 000 euron vuosimaksu voi tuntua kalliilta, mutta jos 3D-mallintajan keskipalkka on 2 949 €/kk, tarvitsee liitännäisen säästää mallintajalta keskimäärin 4,4 h/kk maksaakseen itsensä takaisin (Palkkavertailu, 2019).

4 POHDINTA JA JOHTOPÄÄTÖKSET

Opinnäytetyön tavoitteena oli luoda toimintatapa, jolla saisi nopeutettua CAD-mallin tuomisen prosessia Unity-pelimoottoriin. Uudessa toimintatavassa CAD-malli tuotiin Unityyn käyttäen Pixyz-liitännäistä. Pixyz-liitännäinen oli asiakasprojektin tekohetkellä uusi ja innovatiivinen työkalu ja vastaavanlaisia muita vaihtoehtoja ei ollut saatavilla. Toimeksiantajan asiakasprojektin tapauksessa Pixyz-liitännäisellä saatiinkin suoraviivaistettua toimintatapaa ja tämä toimintatapa otettiin projektissa käyttöön.

Ideaalisissa tilanteissa Pixyz-liitännäisellä ja sen myötä manuaalisen optimoinnin ja UV-kartoituksen automatisoinnilla pystyy potentiaalisesti säästämään huomattavan määrän aikaa. Näitä ideaalisia tilanteita ei kuitenkaan kovin usein tapahdu, joten liitännäisellä ei useinkaan pysty täysin korvaamaan 3D-mallinnuksen tarvetta. Optimointialgoritmi saattaa vähentää polygonmäärää liian tehokkaasti hajoittaen geometriaa, tai malli pitäisi saada vähemmän tehokkaaseen VR-ympäristöön, jolloin malli jää optimoinninkin jälkeen vielä liian raskaaksi tai epämuodostuu liikaa.

Ajan varsinaiseen säästämiseen vaikuttaa moni tekijä. Kokenut animaattori tai 3D-mallintaja voi kyetä nopeastikin optimoimaan ja UV-kartoittamaan malleja huomattavasti paremmalla visuaalisella laadulla kuin automatisoitu algoritmi. Ihminen osaa varmemmin myös erottaa tärkeät yksityiskohdat, joiden täytyy olla tarkkoja ja vähemmän tärkeät yksityiskohdat, joiden tarkkuutta voi huomattavasti laskea tai poistaa kokonaan. Uudehkon liitännäisen mukana tulevat myös uudet ongelmat, kuten kumoamis ja uudelleen tekominaisuuksien puute. Aiemmin animaattorille kuuluneet animointi ja muut mallin käsittelyyn kuuluvat osat puolestaan kuluttavat aikaa ohjelmoijalta. Jos projektissa on vain yksi ohjelmoija, lisääntynyt työmäärä myös hidastaa projektin etenemistä.

Vaikka asiakasprojektissa tässä opinnäytetyössä kehitetty toimintatapa otettiin käyttöön, sitä ei pysty käyttämään kokonaisvaltaisesti CAD-mallien tuonnissa Unityyn, vaan toimintatapa täytyy valita tapauskohtaisesti. Varsinkin pienemmissä projekteissa, joissa esitellään CAD-malleja interaktiivisissa esittely-ympäristössä tämä voisi olla validi toimintatapa.

Uutta toimintatapaa ja Pixyz-liitännäistä voi myös soveltaa vain osittain, vaikka iteratiivisessa kehityksessä. Tällöin malli tuodaan projektin alkuvaiheessa jo paikanpitäjäksi antamaan parempaa vaikutelmaa lopullisesta visuaalisesta ilmeestä samalla, kun animaattori optimoi vielä lopullista mallia.

Opinnäytetyössä keskityttiin pääsääntöisesti mallien geometrian optimointiin eli polygonmäärän vähentämiseen ja LOD-mallien käyttöä ei tutkittu. LOD-malleja käyttämällä voi myös parantaa VR-sovelluksen suorituskykyä, mutta samalla pitää käytössä tarkempaa korkean resoluution mallia, kun sitä tarkastellaan lähietäisyydeltä. Näihin ei kuitenkaan opinnäytetyössä perehdytty, sillä projektin aiemmassa toimintatavassa niitä ei 3ds Maxissa luotu.

Toimintatavassa on vielä jatkokehittävää. LOD-mallien käyttäminen voisi parantaa visuaalista laatua ja Pixyz-liitännäisen päivitykset parantavat optimoinnin laatua ja nopeutta. Pixyzin osalta voisi myös harkita Pixyz Studion käyttöä Unity-liitännäisen tai 3ds Maxin sijaan, vaikka se toisikin uuden ohjelman toimintatapaan. Pixys Studion vahvuutena ovat laajemmat toiminnallisuudet, sekä suuren CAD-datan käsittelyyn suunniteltu ympäristö, joten mallien käsittely saattaa olla nopeampaa ja helpompaa.

Pelitekniikat kuitenkin kehittyvät nopeasti ja tuotannon alat alkavat kiihtyvällä vauhdilla kiinnostumaan virtuaalisesta visualisoinnista. Sen myötä myös pelimoottorien käytöstä visualisoinnin työkaluna. Lisääntynyt kiinnostus kasvattaa kysyntää tehokkaammille tavoille tuoda jo olemassa olevaa dataa näkyville uusilla tavoilla, jonka ansiosta Pixyzin kaltaisia liitännäisiä ja työkaluja saattaa tulla useampikin seuraavan vuosikymmenen aikana. Lisääntynyt kilpailu luo myös olemassa oleville yrityksille paineita parantaa omia työkalujaan kilpailukyvyyn säilyttämiseksi. Myös tietokoneet ja renderöintitekniikat kehittyvät vauhdilla vähentäen optimoinnin tarvetta tai tarve poistuu kokonaan.

LÄHTEET

ARTS experts GmbH 2019. Augmented Reality and Virtual Reality in manufacturing and industry. Viitattu 6.1.2020 <https://arts.eu/blog/augmented-reality-and-virtual-reality-in-manufacturing-and-industry>.

Autodesk 2015. Introduction to polygons. Viitattu 8.5.2020 <https://knowledge.autodesk.com/support/maya-lt/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/MayaLT/files/Polygons-overview-Introduction-to-polygons-htm.html>.

Baldwin, J. 2017. How CAD is Currently Playing into the VR Industry. Viitattu 6.1.2020 <https://blog.grabcad.com/blog/2017/08/14/cad-currently-playing-vr-industry>.

Blender 2013. History. Viitattu 22.5.2020 <https://www.blender.org/foundation/history/>.

Creative Bloq 2019. Unity vs Unreal Engine: which game engine is for you? Viitattu 6.1.2020 <https://www.creativebloq.com/advice/unity-vs-unreal-engine-which-game-engine-is-for-you>.

Easy Render 2019. A Brief History of 3D Visualizations: The Ins and Outs. Viitattu 6.1.2020 <https://www.easyrender.com/3d-visualization/a-brief-history-of-3d-visualizations-the-ins-and-outs>.

GrabCad. Esimerkkimalli. Viitattu 26.5. <https://grabcad.com/library/cnc-router-500-x-500-mm-1>.

Heinonen, N. 2019. Monialustaisen laajennetun todellisuuden sovelluksen toteutus. Viitattu 26.5.2020 <https://www.theseus.fi/handle/10024/169542>.

iGlobe Partners 2019. Unity, Forerunner in global game development software. Viitattu 6.1.2020 <https://www.iglobepartners.com/portfolio-unity.html>.

Jacobs, A. 2014. The advantages of 3D design. Viitattu 6.1.2020 <https://plusgroups.com/the-advantages-of-3d-design>.

Keane, P. 2019. VR in CAD: Where Are We Now? Viitattu 23.5.2020 <https://www.engineering.com/DesignSoftware/DesignSoftwareArticles/ArticleID/19116/VR-in-CAD-Where-Are-We-Now.aspx>.

Keituri, M. 2018. VR-tarkastelusolun suunnittelu. Viitattu 26.5.2020 <https://www.theseus.fi/handle/10024/145621>.

Klementtinen, N. 2012 Teollisuuden 3D-mallin optimointi reaaliaikarenderöintiin. Viitattu 8.5.2020 <http://www.theseus.fi/handle/10024/52758>.

- Kotimaisten kielten keskus 2018. Virtuaalitodellisuus. Viitattu 6.1.2020 <https://www.kielitoimistonsanakirja.fi/netmot.exe?SearchWord=virtuaalitodellisuus&dic=1&page=results&UI=fi80&Opt=1>.
- Martin, E. 2018. Unity Partners with PiXYZ Software to unlock CAD data for real-time development. Viitattu 5.5.2020 <https://blogs.unity3d.com/2018/03/06/unity-partners-with-pixyz/>.
- Martin, E. 2019. Streamline your data workflows with PiXYZ 2019.1. Viitattu 6.5. 2020 <https://blogs.unity3d.com/2019/05/08/streamline-your-data-workflows-with-pixyz-2019-1/>.
- Muuriaisniemi, L. 2019. Rakennussuunnittelu virtuaalitodellisuudessa : operaatiohuoneen suunnittelutyökalu. Viitattu 26.5.2020 <https://www.theseus.fi/handle/10024/267151>.
- Palkkavertailu. Viitattu 6.5.2020 <https://palkkavertailu.com/palkka/3d-mallintaja>.
- Petty, J. Ei julkaisupäivää. What is 3ds Max & What is it Used For? Viitattu 22.5.2020 <https://conceptartempire.com/what-is-3ds-max/>.
- Pixyz Plugin. Viitattu 5.5.2020 <https://www.pixyz-software.com/plugin/>
- Plewa, K. 2018. Battle of software 2020: Blender vs Maya. Viitattu 22.5.2020 <https://www.sculpteo.com/blog/2018/12/17/battle-of-software-blender-vs-maya/>.
- Plewa, K. 2019. CAD vs 3D modeling software: what is the difference? Viitattu 6.1.2020 <https://www.sculpteo.com/blog/2019/03/19/cad-vs-3d-modeling-software-what-is-the-difference>.
- Porter, T. 2017. VR Optimization Tips from Underminer Studios. Viitattu 8.5.2020 <https://software.intel.com/content/www/us/en/develop/articles/vr-optimization-tips-from-underminer-studios.html>.
- Rowe, S. 2019. Learn VR Development: Tips, tricks, and guides to develop VR and AR applications. Viitattu 6.1.2020 <https://circuitstream.com/blog/programming-development-guides>.
- Sanders, A. 2019. Key Differences Between AutoCAD and Other 3D Programs. Viitattu 6.1.2020 <https://www.lifewire.com/autocad-vs-other-3d-programs-140526>.
- Siddi, F. 2020. Blender by the Numbers – 2019. Viitattu 22.5.2020 <https://www.blender.org/press/blender-by-the-numbers-2019/>.
- Sparks, M. 2017. Skills Needed to Create a Serious VR Game with a Game Engine. Viitattu 6.1.2020 <https://learningsolutionsmag.com/articles/2396/skills-needed-to-create-a-serious-vr-game-with-a-game-engine>.
- Thomsen, M. 2012. History of the Unreal Engine. Viitattu 6.1.2020 <https://www.ign.com/articles/2010/02/23/history-of-the-unreal-engine>.

Unity Docs. Viitattu 8.5.2020 <https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html>.

Virtual reality society 2017a. History Of Virtual Reality. Viitattu 6.1.2020
<https://www.vrs.org.uk/virtual-reality/history.html>.

Virtual reality society 2017b. The ultimate guide to virtual reality headsets. Viitattu 6.1.2020
<https://www.vrs.org.uk/the-ultimate-guide-to-virtual-reality-headsets>.