Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Otso Jylhä

Unreal Studio -ohjelman hyödyntäminen tilaesittelyissä

Metropolia Ammattikorkeakoulu

Medianomi (AMK)

Viestintä

Opinnäytetyö

8.5.2019

Tekijä(t) Otsikko Sivumäärä Aika	Otso Jylhä Unreal Studio -ohjelman hyödyntäminen tilaesittelyissä 24 sivua + 1 liitettä 8.5.2019
Tutkinto	Medianomi (AMK)
Tutkinto-ohjelma	Viestinnän tutkinto-ohjelma
Suuntautumisvaihtoehto	3D-animointi ja -visualisointi
Ohjaaja(t)	Lehtori Jaro Lehtonen

Tämän tutkielman tavoitteena on tutkia Unreal Studio -ohjelman ja Datasmith-työkalun hyödyntämistä tehokkaaseen reaaliaikaisten tilaesittelyjen luontiin. Aiheesta käsitellään vaadittavat alkutoimenpiteet 3ds Max -ohjelmassa sekä viimeistelyvaiheet Unreal Studio -ohjelmassa. Erityisesti tavoitteena on tutkia tavallisen visualisointiprojektin muuttamista reaaliaikaiseksi tilaesittelyksi mahdollisimman pienellä työmäärällä ja siten, että lopputulos muistuttaisi alkuperäisiä visualisointeja mahdollisimman paljon.

Teoriaosuudessa käydään läpi myös muita vaihtoehtoja toteuttaa virtuaalisia tilaesittelyjä ja sen jälkeen projektiosuudessa keskitytään luomaan 3ds Max -visualisointiprojektista reaaliaikainen tilaesittely Unreal Studio -ohjelmalla. Tutkielma on rajattu siten, ettei ympäristön mallinnusta tai teksturointia käydä juurikaan läpi. Päätarkoituksena on siis keskittyä luomaan realistinen ympäristö jo valmiiksi luodusta kokonaisuudesta. Projektiosuus on kohdistettu PC:lle ja sen kautta virtuaalilaseille kuten HTC Vive ja Oculus Rift, joten mobiililaitteille vaadittavia optimointeja ei käydä läpi. Datasmith ja Unreal Studio toimivat hyvin ja nopeuttaa työskentelyä, mutta vielä beetavaiheessa ohjelmat eivät ole virheettömiä.

Aihe on kohdistettu pääosin teknisille 3D-graafikoille, mutta ilmankin teknistä kokemusta on mahdollista saada vahva ymmärrys työprosessista. Lukijalta odotetaan 3D-mallintamisen, teksturoimisen ja renderöinnin perusteita.

Avainsanat	reaaliaika, arkkitehtuuri, virtuaalitodellisuus, 3D, VR

Author(s) Title Number of Pages Date	Otso Jylhä Utilizing Unreal Studio in Architectural Visualization Project 24 pages + 1 appendices 8 May 2019
Degree	Bachelor of Culture and Arts
Degree Programme	Media
Specialisation option	3D Animation and Visualization
Instructor(s)	Jaro Lehtonen, Senior Lecturer

The aim of this thesis is to explore the use of the Unreal Studio and the Datasmith tool to create real-time virtual environments. This thesis will cover the initial steps in 3ds Max and finishing touches in the Unreal Studio. The main point is to study the workflow of transforming an ordinary visualization project into a real-time presentation with as little workload as possible, and so that the end results would resemble the original two-dimensional visualizations.

The theoretical part explores other options for implementing virtual space presentations, and then the project part focuses on creating a real-time presentation from the 3ds Max visualization project with Unreal Studio. The main purpose is to focus on creating a realistic real-time environment with a project that is already textured and modelled for two-dimensional visualization purposes. Therefore, the thesis will not talk a lot about creating the environment from the start. The product is targeted for PC and headset platforms such as HTC Vive and Oculus Rift, so the optimizations required for mobile devices are not considered.

The thesis is mainly for technical artists but even with less technical experience it is possible to get a strong understanding of the work process. The reader is expected to know the basics of 3D modeling, texturing and rendering.

Keywords	real-time, architecture, virtual reality, 3D, VR

Sisällys

1 Johdanto

Tässä tutkielmassa tutkitaan 3D-visualisointiprojektin muuttamista reaaliaikaiseksi tilaesittelyksi hyödyntämällä Datasmith-työkalua ja Unreal Studio -ohjelmaa. Työprosessin alkulähtökohtana käytetään 3ds Max -mallinnusohjelmalla luotua kolmiulotteista ympäristöä, josta on luotu realistisia kuvavisualisointeja Corona-ohjelmalla. Tämä ympäristö on tarkoitus siirtää Unreal Engine -pelimoottoriin, jonka avulla sitä voidaan esitellä reaaliajassa joko virtuaalilaseilla tai tietokoneen ruudulta. Tavoitteena on tehdä tämä tehokkaasti ja hyödyntää mahdollisimman paljon jo tehtyjä malleja, materiaaleja ja valoja. Tarkoituksena on myös vertailla muita tapoja luoda reaaliaikainen tilaesittely ja vertailla työhön käytettyjä resursseja.

Unreal Studio on ohjelma, joka sisältää Datasmith-työkalun. Datasmith mahdollistaa muun muassa CAD- ja 3ds Max -tiedostojen tuonnin Unreal Engine -ohjelmaan. Tulevaisuudessa Datasmith-ohjelman on tarkoitus tukea muidenkin ohjelmien natiividataa. (Unreal Engine Documentation 2018.) Datasmith-työkalun lisäksi Unreal Studio sisältää lukuisia työprosessia helpottavia ominaisuuksia, kuten sata Allegorithmic-yrityksen tuottamaa Substance-materiaalia (Epic Games 2018).

2 Arkkitehtivisualisoinnista virtuaalitodellisuuteen

Yksinkertaisimmillaan arkkitehtivisualisointi on suunnitelman tai konseptin muuntamista asiakkaalle formaattiin, jota on helppo tarkastella ja joka antaa paremman ymmärryksen tuotteen lopullisesta visuaalisesta ilmeestä. Suunnitelmat ja konseptit voivat olla esimerkiksi kaksi- tai kolmiulotteisia piirustuksia, valokuvia tai internetsivuilta poimittuja kuvakokoelmia. Visualisoijan tehtävä on sisäistää tämä informaatio ja tuottaa siitä kokonaisuus, joka vastaa asiakkaan toiveita. Käyttötarkoituksesta riippuen projektin lopputulos voi olla realistinen kuvakooste, animaatio, 3D-printattu malli tai interaktiivinen ympäristö, jossa vapaasti liikkuminen on mahdollista. (Learn Arch Viz 2017a.)

Arkkitehtivisualisoinneissa lopputulos on yleensä useampi kuin yksi kuva ennalta valituista sijainneista. Arkkitehtien luomat suunnitelmat mallinnetaan, teksturoidaan ja valaistaan, jonka jälkeen tietokoneelle annetaan käsky laskea kuva ympäristöstä tai mallista. Tätä operaatiota kutsutaan renderöinniksi. Renderöinnissä tietokone laskee, miten

valo käyttäytyy osuessaan mallien pintaan simuloiden oikean valon käyttäytymistä. Renderöinti on hyvin raskasta, ja tietokoneen laskentateho vaikuttaa tarvittavaan aikaan. (Learn Arch Viz 2017a.) Lopullisten kuvien renderöintiin vaadittavat ajat vaihtelevat hyvin paljon, mutta usein puhutaan useista tunneista ja valaistus joudutaan laskemaan jokaiselle kuvalle uudestaan. Usein renderöintivaiheessa kuvista lasketaan isommat versiot kuin on tarpeellista, jotta kuvaa on vielä mahdollista rajata ja muokata. Tämä kasvattaa renderöintiaikoja eksponentiaalisesti. Jos kuvan leveys ja korkeus kaksinkertaistetaan, kasvaa renderöintiin vaadittava aika nelinkertaiseksi. Pelimoottoria käytettäessä valaistus joudutaan laskemaan vain kerran, jonka jälkeen kuvakohtainen laskenta-aika jää hyvin pieneksi (Ronai 2017). Jos valaistuksen renderöintiin kuluu tunti, kestää pelimoottorilla 200 kuvan animaation renderöinnissä vain hieman yli tunti, kun taas esimerkiksi Corona tai V-Ray-ohjelmilla tässä kestäisi 200 tuntia.

Virtuaaliympäristön luonti pelkän staattisen kuvakoosteen sijaan hyödyttää projektin jokaista osapuolta kuten arkkitehtejä, suunnittelijoita, 3D-artisteja ja asiakkaita. Edes hienoimmilla kuvilla ei ole mahdollista saada aikaan yhtä immersiivisiä kokemuksia kuin virtuaalilaseilla (Maheut 2017). Pelimoottoria käytettäessä valaistuksen laskemisen jälkeen kuvakulmia, kuvan resoluutiota ja materiaaleja voidaan vielä muuttaa reaaliajassa. Tämä nopeuttaa suunnitteluprosessia huomattavasti. (Ronai 2017)

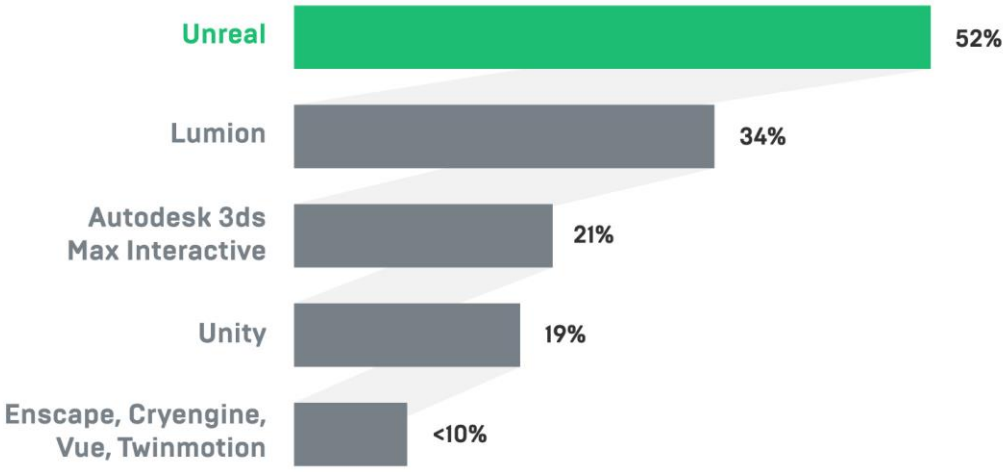
Virtuaalimallin luonti antaa myös asiakkaalle paremman mahdollisuuden vaikuttaa projektin lopputulokseen. On esimerkiksi mahdollista luoda ympäristö erilaisten värimaailmojen ja huonekalukokonaisuuksien kokeiluun, jota asiakas tai sisustussuunnittelija voi reaaliajassa muokata. Tällöin parhaan mahdollisen kokonaisuuden löytyminen on vaivattomampaa. (LNG Studios 2016.) Mahdollisuus liikkua tilassa vapaasti ja katsoa jokaiseen suuntaan voi myös ennalta ehkäistä odottamattomia ongelmia tai väärinymmärryksiä (Parkin 2018).

3 Vaihtoehtoiset työkalut

3.1 Vaihtoehtoiset moottorit

Unreal Engine ei ole ainoa vaihtoehto reaaliaikaisen ympäristön luomiseen. Muita suosittuja alustoja on lukuisia, joista merkittävimmät käydään työssä läpi. Puolueetonta ja

luotettavaa kantaa ”parhaasta” moottorista tilojen reaaliaikaiseen visualisointiin on mahdotonta löytää, ja tekijöistä riippuen on mahdollista päästä näyttäviin lopputuloksiin. Alla oleva kuvio 1 voi kuitenkin antaa osviittaa sopivimmista työkaluista.



Kuvio 1. Prosentuaaliset osuudet suosituimmista reaaliaikamoottoreista arkkitehtivisualisointien luontiin vuonna 2018 (Pimetel 2018).

CGArchitect Internet -sivuston tekemän kyselyn perusteella voidaan havaita selkeää kasvua reaaliaikaisten moottorien käytössä. Lähes jokaisen näiden suosio oli kaksinkertaistunut vuosien 2016 ja 2018 kyselyjen välillä. (Mottle 2016. Mottle 2018.) Osa Unreal Enginen suosiosta voidaan selittää ilmaisen Unreal Studio -ohjelman beetaversion julkaisulla, joka mahdollistaa saumattoman tiedonsiirron Datasmith-työkalun avulla visualisointiohjelmasta pelimoottoriin. Vuonna 2017 julkaistu Unreal Studio tukee lukuisia tiedostomuotoja ja on erityisesti suunnattu arkkitehtivisualisointiin laajalla materiaali ja valaistustuella. (Pimetel 2018.)

3.1.1 Unity

Erityisesti pelialalla suosittua Unity-ohjelmaa voi käyttää myös reaaliaikaisissa arkkitehtivisualisoinneissa. Lukuisien tiedostotyyppien tuki mahdollistaa lähes saumattoman tiedonkulun mallinnusohjelmasta pelimoottoriin ja lopuksi vielä päätelaitteelle. Päätelaitteina Unity tukee tietokoneiden lisäksi niin mobiililaitteita, virtuaalilaseja kuin useimpia konsoleita. (Unity n.d. a.) Mikäli projektin tavoitteena on realistinen lopputulos, eivät

moottorin perusominaisuudet välttämättä riitä. Muun muassa värikorjaukseen, jälkikäsittelyyn ja efekteihin on ladattava lisäosat muista lähteistä. Tässä kuitenkin auttaa laaja valikoima liitännäisiä Unityn Asset Store -palvelusta, josta löytyy niin maksullisia kuin ilmaisia työkaluja. (Franczak 2017.) Unity on C#-ohjelmointikielipohjainen, joten mukautettuja toiminnallisuuksia tehtäessä se vaatii ymmärrystä ohjelmoinnista. (Unity n.d. b.) Valmiita pohjia on mahdollista ostaa ja käyttää, jolloin on mahdollista pärjätä ilman ohjelmointia.

3.1.2 Twinmotion ja Lumion

Twinmotion ja Lumion ovat käyttäjäystävällisiä moottoreita reaaliaikaisen grafiikan tuottamiseen. Unity-pelimoottoriin verrattuna ei ohjelmointitaitoja vaadita, joten nämä ohjelmat soveltuvat arkkitehtien sekä 3D-artistien käyttöön. Tämä kuitenkin rajoittaa mahdollisuuksia tehdä tilaesittelyyn omia efektejä tai ominaisuuksia. Reaaliajassa laskettava valaistus mahdollistaa välittömän palautteen ja interaktiivisuuden, mutta laadullisesti on vaikea päästä samalle tasolle kuin esimerkiksi ohjelmilla V-Ray tai Corona. Twinmotion ja Lumion tukevat lukuisia CAD ja 3D-tiedostomuotoja, mikä mahdollistaa joustavan tiedonsiirron ohjelmien välillä. Ympäristöjen luontiin ohjelmat tarjoavat suuren määrän valmiiksi tehtyjä elementtejä ja materiaaleja, joita on mahdollista kustomoida sään ja vuodenaikojen vaihtuessa. Virtuaalilaseille on molemmilla alustoilla mahdollista luoda 360 asteen panoraamoja, mutta vain Twinmotion tukee täysin interaktiivista virtuaalimaailmaa HTC-Vive- ja Oculus Rift -laitteilla. (Higgwe n.d.)

3.1.3 Panotour

Panotour on ohjelma, jolla voi luoda tilaesittelyjä 360 asteen kuvista. Tilaesittely on mahdollista ladata internettiin, mikä helpottaa ympäristön esittelyä asiakkaille. Internet-sivu pohjaista tilaa voi tarkastella tietokoneen ruudulta tai esimerkiksi Oculus Go ja Gear VR -virtuaalilaseilla. Tiloissa ei voi kuitenkaan liikkua täysin vapaasti, koska esittely koostuu kuvista, jotka on otettu ennalta määritellyistä pisteistä. Näiden pisteiden välillä on kuitenkin mahdollista liikkua, mikä auttaa havaitsijaa tilan tulkinnassa. (Dukes 2016.)

3.2 Vaihtoehtoiset tiedostomuodot datansiirrossa

Usein pelimoottorit eivät tue mallinnusohjelmien natiivitiedostomuotoa, ja tällöin täytyy projekti tallentaa muotoon, jota pelimoottori ymmärtää. Kolmiulotteisen datan tallentamiseen on lukuisia jopa satoja tiedostomuotoja. Suosituimmat tiedostomuodot tiedonsiirrossa ovat usein avoimeen lähdekoodiin perustuvia, mikä helpottaa tiedostomuodon ymmärtämistä ja auttaa kehittäjiä lisäämään tuen omiin ohjelmiinsa. Usein tiedostomuodot on kehitetty tiettyihin tarkoituksiin, joten ne tukevat eri asioita. Esimerkiksi STL-tiedostomuoto, joka on laajalti käytössä 3D-tulostuksessa, ei sisällä tietoa mallin väristä tai tekstuureista. (Chakravorty 2019.)

3.2.1 3D Object File (OBJ)

Alias Systems korporaation (entiseltä nimeltään Alias Wavefront) kehittämä 3D Object File on laajalti tuettu yksinkertainen tiedostomuoto, joka pitää sisällään tiedon kolmiulotteisen mallin tärkeimmistä ominaisuuksista. Näitä ovat mallin pisteiden sijainnit, UV-tekstuurikoordinaatit, mallin pisteiden luomat tasot ja niiden pintanormaalit. Formaatin yksinkertaisuudesta seuraa kuitenkin rajoituksia. Animaatiota, objektihierarkiaa, monimutkaisia materiaaleja tai useampaa UV-tekstuurikoordinaattikanavaa ei ole mahdollista siirtää OBJ-tiedostomuotoon. (Simmons n.d.)

```
# Blender v2.79 (sub 0) OBJ File: ''
# www.blender.org
mtllib cube.mtl
o Cube_Cube.001
v -1.000000 -1.000000 1.000000
v -1.000000 1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v -1.000000 1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v 1.000000 1.000000 1.000000
v 1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
vt 0.250043 0.500000
vt 0.250043 0.250043
vt 0.500000 0.250043
vt 0.500000 0.500000
vt 0.749957 0.500000
vt 0.749957 0.749957
vt 0.500000 0.749957
vt 0.500000 0.999913
vt 0.250043 0.999913
vt 0.250043 0.749957
vt 0.000087 0.749957
vt 0.000087 0.500000
vt 0.500000 0.000087
vt 0.250043 0.000087
vn -1.0000 0.0000 0.0000
vn 0.0000 0.0000 -1.0000
vn 1.0000 0.0000 0.0000
vn 0.0000 0.0000 1.0000
vn 0.0000 -1.0000 0.0000
vn 0.0000 1.0000 0.0000
usemtl None
s off
f 1/1/1 2/2/1 4/3/1 3/4/1
f 3/4/2 4/5/2 8/6/2 7/7/2
f 7/7/3 8/8/3 6/9/3 5/10/3
f 5/10/4 6/11/4 2/12/4 1/1/4
f 3/4/5 7/7/5 5/10/5 1/1/5
f 8/13/6 4/3/6 2/2/6 6/14/6
```

Kuvio 2. Kuutio tallennettuna OBJ-tiedostomuotoon.

3.2.2 Filmbox (FBX)

Kaydaran kehittämä ja Autodeskin omistukseen vuonna 2006 siirtynyt tiedostomuoto FBX (Filmbox) on laajalti tuettu ja monipuolisempi kuin 3D Object File -formaatti. FBX täyttää OBJ-tiedostomuodon puutteet UV-tekstuurikoordinaatti, animaatio ja objektihierarkiatuilla (Simmons n.d). Lisäksi tiedostoon on mahdollista tallentaa tietoa yksinkertaisista kameroista, valoista ja materiaaleista (Autodesk FBX 2017). Valitettavasti FBX ei tue arkkitehtivisualisoinneissa laajalti käytettyjä V-Ray tai Corona materiaaleja, joten tiedonsiirron jälkeen kaikki materiaalit on luotava uudestaan (Nichols 2017).

3.2.3 COLLADA (DAE)

Khronos-yhdistyksen kehittämä Collada-tiedostoformaatti on ollut laajassa käytössä peli- ja elokuvateollisuudessa. Sen tavoitteena oli tulla standarditiedostomuodoksi kolmiulotteisen datan tallentamisessa, ja vuonna 2013 se sai oman ISO (International Organization for Standardization)-merkinnän. Tämän seurauksena Collada on ollut laajalti tuettu ja suosittu tiedostomuoto. Collada ei ole kuitenkaan pysynyt 3D-ohjelmien kehityksessä mukana, ja käyttäjät ovat siirtyneet Filmbox ja 3D Object File -tiedostomuotojen käyttöön. (Chakravorty 2019.)

4 Projekti: Visualisointiprojektin muuttaminen virtuaaliseksi tilaesittelyksi

Erityisesti arkkitehtivisualisoinneissa on tärkeää, että koko ympäristö on mahdollista siirtää pelimoottoriin kadottamatta tietoa materiaaleista tai valaistuksesta. Täydellisissä olosuhteissa ympäristö näyttäisi siis täysin samalta mallinnusohjelman tuottamissa kuvissa ja pelimoottorissa. Tätä varten Epic Games on julkaissut Unreal Studio -ohjelman ja Datasmith-työkalun.

4.1 Alkutoimenpiteet

Jo projektin alussa on hyvä miettiä, millä mittakaavalla ympäristö luodaan. 3ds Max -ohjelmassa on mahdollisuus valita käytettävä mittayksikkö, kun taas Unreal Engine on senttimetripohjainen, joten järkevintä olisi tehdä projekti alusta asti senttimetreissä. Jos tila on luotu käyttäen muuta mittayksikköä, on se suositeltavaa muuttaa senttimetripohjaiseksi jo mallinnusohjelmassa. Mittakaavan lisäksi Unreal Engine saattaa aiheuttaa

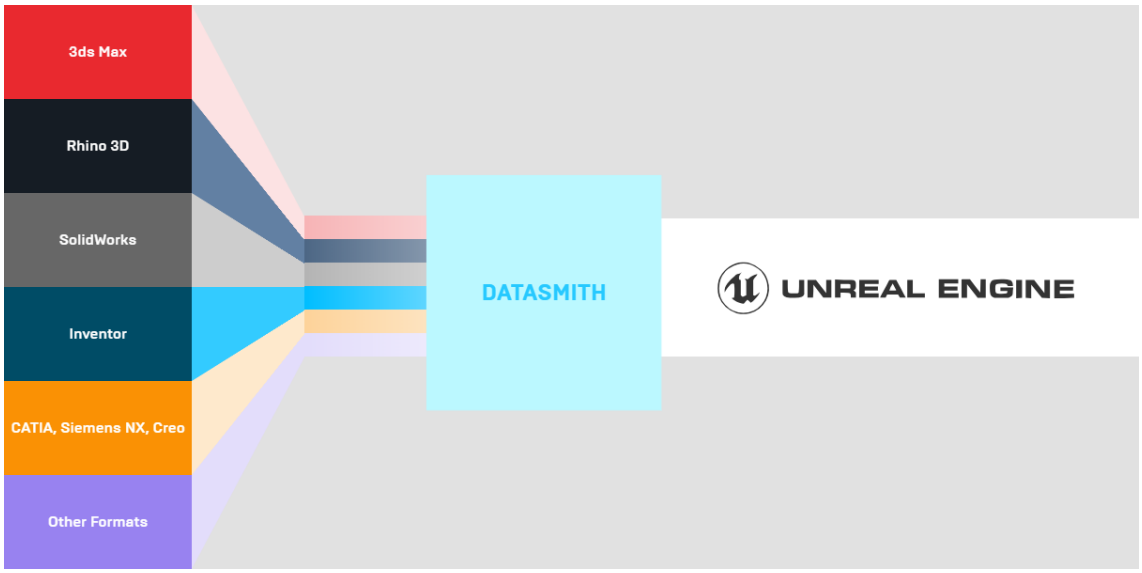
graafisia häiriöitä, jos projekti on luotu hyvin kauas 3D-avaruuden keskipisteestä. Nämä laskuvirheet ovat useimmissa tapauksissa tietokoneen laskennassa tapahtuvia pyöristysvirheitä, jotka ovat korjattavissa siirtämällä ympäristö lähemmäs 3D-avaruuden keskipistettä eli origoa. (Doyle 2018b, osa 6.0.)

Unreal Engine on reaaliaikaista kuvaa tuottava pelimoottori, jonka takia laajoja ja raskaita 3D visualisointiprojektia täytyy keventää. Mikäli päätelaitteena on tavallinen tietokoneen ruutu, on suositeltava ruututaajuus noin 30-60 ruutua sekunnissa. Virtuaalilaseja käytettäessä suositeltava ruututaajuus on vähintään 90. Tätä matalammat ruututaajuudet saattavat aiheuttaa pahoinvointia. (Doyle 2018a, osa 2.1.)

Usein visualisointiprojekteissa käytettävät mallit ovat hyvin yksityiskohtaisia ja tämän takia liian raskaita reaaliaikaiseen tarkasteluun, joten on suositeltavaa luoda niille useampia versioita eri katsontaetäisyyksiä varten. Näitä versioita kutsutaan LOD-objekteiksi (Level of Detail), ja tarkoituksena on näyttää esineestä kevyempi versio tarkasteltaessa sitä kaukaa. Tämä toimenpide vähentää malleissa tarvittavien yksityiskohtien määrää kuvien piirtoon ja kasvattaa kuvatajuutta. LOD-tasoja on mahdollista luoda mallinnusohjelmissa, mutta Unreal Engine -ohjelmalla on myös mahdollista luoda kevennetyt versiot objekteista automaattisesti. Automaattisen prosessin jälkeen on vielä mahdollista asettaa kynnysarvot versioiden vaihtoon ja niiden kevennysprosentit. (Unreal Engine 2019.) Muita raskaita objekteja, kuten suuria nurmikenttiä ja metsikköjä, on suositeltavaa korvata reaaliaikaiseen katseluun tarkoitetuilla versioilla (Doyle 2018a. osa 3.3).

4.2 Datasmith-työkalun käyttö

Tarvittavien alkutoimenpiteiden jälkeen projekti viedään 3ds Max -ohjelmasta Unreal Engine -ohjelmaan käyttämällä Datasmith-työkalua. Verrattuna yleiseen FBX-tiedostoformaattiin Datasmith suoraviivaistaa työprosessia. Se muun muassa tunnistaa instanssiobjektit, luo automaattiset valaistus-UV-koordinaatit ja säilyttää projektihierarkian ja tasot. Yleisimpien säteenseurantaohjelmien (mm. V-Ray ja Corona) materiaalit Datasmith osaa muuttaa Unreal Engine -pohjaisiksi materiaaleiksi, ja pinnan epätasaisuutta simuloivat kohoumakartat muuttuvat reaaliaikaisessa valaistuksessa käytetyiksi normaalikartoiksi. Valo objekteissa siirtoprosessissa siirtyy valon voimakkuus, väri ja IES-profiili. Datasmith-työkalun tarkoitus on siis siirtää projekti mallinnusohjelmasta mahdollisimman samannäköisenä Unreal Studio -ohjelmaan. (Doyle 2018c. osa 1.1.)



Kuvio 3. Lukuisia tiedostomuotoja tukeva Datasmith on silta 3D-ohjelman ja Unreal Enginen välillä.

3ds Max -projektin muuttaminen Datasmith-tiedostomuotoon tapahtuu valitsemalla Datasmith-tiedostomuoto 3ds Max -ohjelman ”vie” valikosta. Kun vienti on tehty, käyttäjä saa ilmoituksen erilaisista ongelmista, kuten tuntemattomista tekstuureista tai materiaalityypeistä, joita Datasmith ei tue. Nämä ongelmat eivät usein ole kriittisiä, mutta saattavat vaikuttaa lopputulokseen, jos niitä ei korjata. Mikäli projekti on hyvin raskas, vienti saatetaan joutua tekemään useammassa erässä. (Doyle 2018c. osa 2.1.)

4.3 Projektin viimeistely Unreal Studiossa

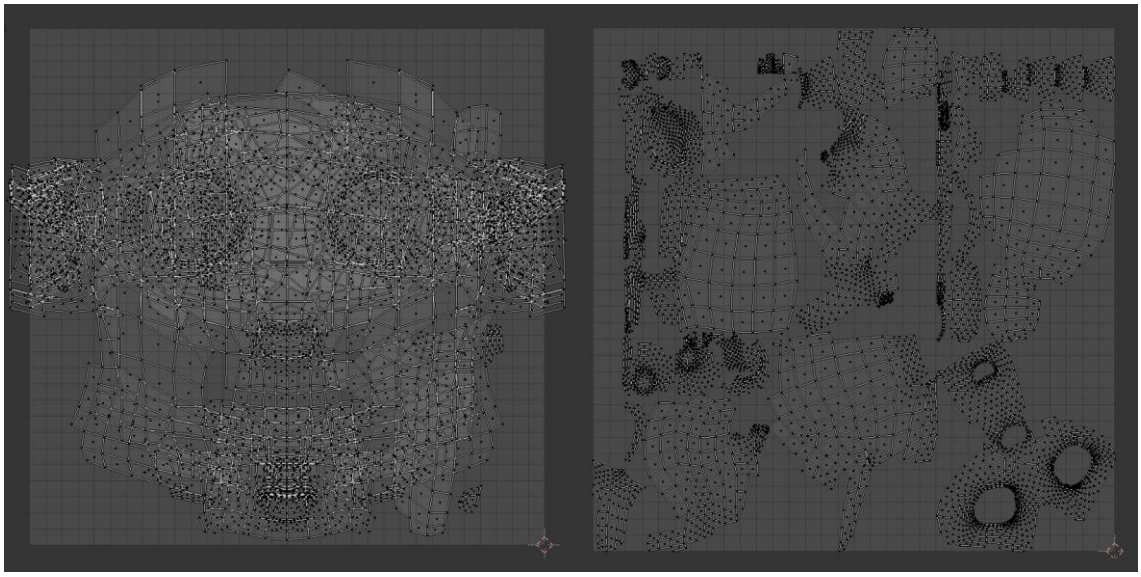
Unreal Studio on avoimessa beetatestauksessa oleva lisäosa Unreal Engine -pelimoottoriin. Studion beetalisenssi on täysin rojaltivapaa, ja sillä saa kaikki ominaisuudet Unreal Studio ja Unreal Engine -ohjelmista. Beetaversion on ilmainen syyskuuhun 2019 asti, jonka jälkeen Studiosta julkaistaan virallinen maksullinen versio, joka maksaa 49 dollaria kuukaudessa. (Unreal Engine n.d. a.)

Unreal Engine -projektin voi luoda täysin tyhjästä tai käyttää valmiita mallipohjia. Nämä pohjat sisältävät mm. valmiiksi luotuja ominaisuuksia hahmon liikuttamiseen. Jos projektin päätelaitteena toimivat virtuaalilasit, on kannattavaa valita pohja, joka sisältää tarvittavat ominaisuudet virtuaalitodellisuutta varten. Tietokoneen ruudulta tarkasteltavaan projektiin on järkevää valita pohja, jossa hahmon liikuttaminen tapahtuu näppäimistöllä ja hiirellä. Jos valittu pohja on Unreal Engine -pohja, täytyy projekti muuttaa Unreal Stu-

dio -pohjaksi, ja tämän voi tehdä ohjelman lisäosat-valikosta. Kun projektipohja on muutettu Unreal Studio -projektiksi, voi siihen tuoda Datasmith-tiedostoja. Datasmith-tiedoston tuonnissa on valittava kansio, jonne tiedostot puretaan. (Doyle 2018d. osa 6.)

4.3.1 Valaistus

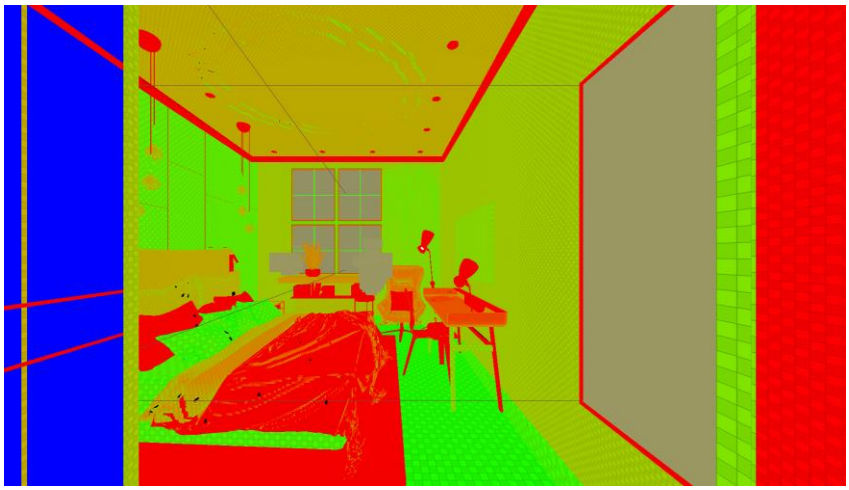
Kun visualisointiprojekti on onnistuneesti tuotu Unreal Engine -ohjelmaan, voidaan alkaa valaista ympäristöä. Unreal Engine tukee reaaliajassa laskettavaa ja esilaskettua valaistusta, mutta nykypäivän näytönohjaimet eivät vielä pysty toistamaan todellisuuteen pohjautuvaa valaistusta reaaliajassa. Tästä johtuen on valaistus laskettava etukäteen. Unreal Engine tallentaa esilasketun valaistuksen erillisiin valokarttoihin. Valokartta tallennetaan jokaiselle mallille erikseen, ja tätä varten jokainen malli tarvitsee toissijaiset UV-tekstuurikoordinaatit. Valokartan UV-koordinaatisto eroaa tavallisesta UV-koordinaatistosta siten, että kartan tasot eivät saa olla toistensa päällä ja jokaisen koordinaatin täytyy olla nollan ja yhden välillä. Datasmith osaa luoda valokartat automaattisesti, mutta monimutkaisemmissa objekteissa saatetaan ne joutua luomaan käsin mallinnusohjelmassa. (Doyle 2018d osa 7.)



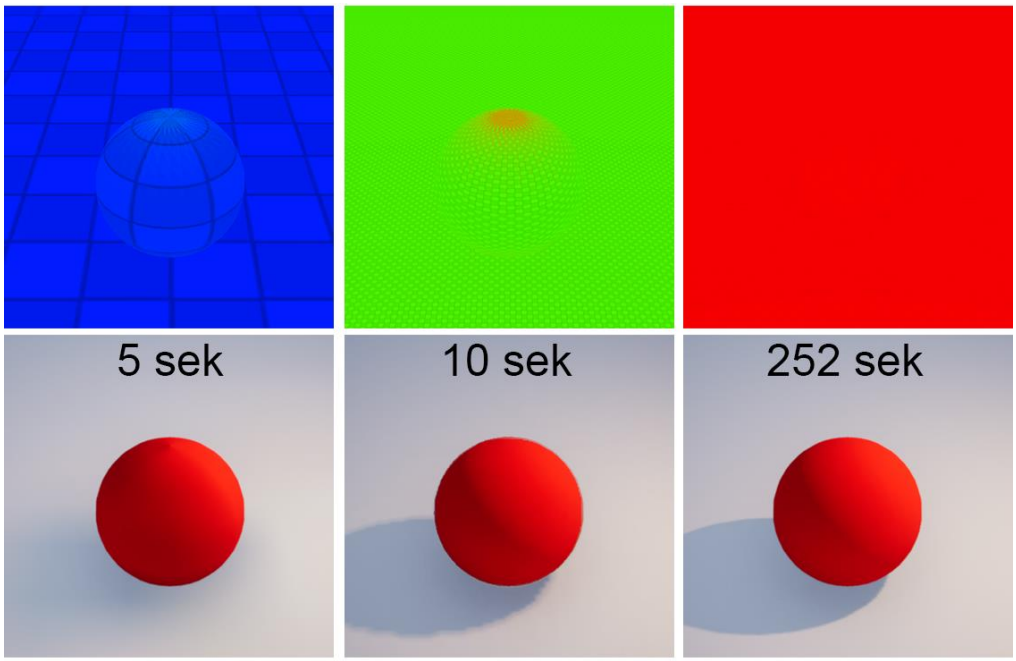
Kuvio 4. Vasemmalla esimerkkitapaus tekstuuri-UV-koordinaateista ja oikealla esimerkkitapaus valaistusta varten luodusta UV-koordinaattikartasta.

Valokarttojen tekstuurikoko vaikuttaa valaistuksen laatuun sekä valaistuksen laskuun tarvittavaan aikaan, joten on tärkeää löytää jokaiselle valokartalle sopiva resoluutio. Tässä työvaiheessa hyödyllinen kuvamoodi on Unreal Studion ”Lightmap Density”.

Tämä työkalu visualisoi ruudulla, kuinka tarkasti valaistus esitetään staattisten esineiden pinnoissa. (Doyle 2018d. osa 7) Suuri valokartan resoluutio saattaa pidentää valaistuksen laskentaan vaadittavaa aikaa, mutta laajoille pinnoille se voi olla välttämätöntä. Esimerkiksi lattiaelementille saatetaan asettaa suositeltua suurempi resoluutio. (Doyle 2018d osa 14.)

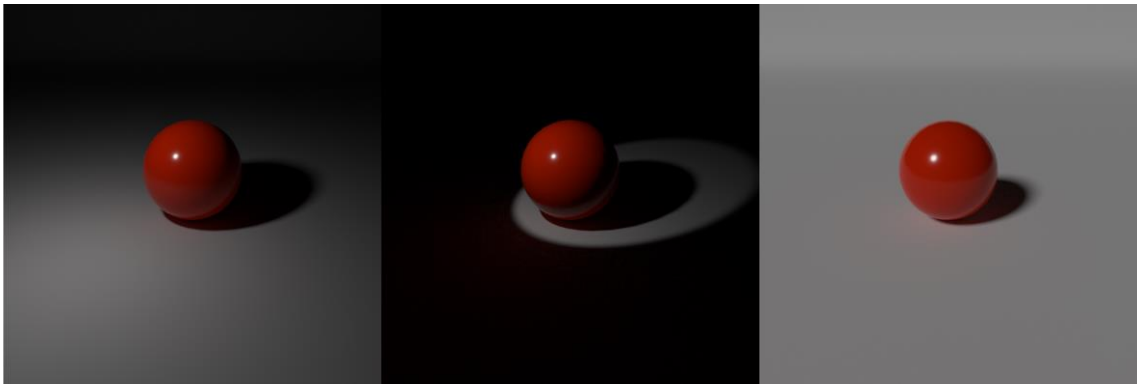


Kuvio 5. Esimerkkitapaus ”Lightmap Density” -kuvamoodista. Punainen väri tarkoittaa, että valokartta on liian tiheä, vihreä väri tarkoittaa optimaalista tiheyttä ja sininen liian harvaa tiheyttä.



Kuvio 6. Esimerkki valokartan resoluution vaikutuksesta esilaskettuun valaistukseen.

Yleisimmät valotyypit ovat pistevalo, kohdevalo ja suora kohdevalo. Pistevalo simuloi tavallisen hehkulampun valaistusta. Kohdevalo on kuin pistevalo, mutta valo säteilee kartion muodossa. Suora kohdevalo on valonlähde, joka simuloi valon säteilyä äärettömän kaukaa. Usein tätä valoa käytetään auringon valona. (Unreal Engine n.d. b.)



Kuvio 7. Pistevalo, kohdevalo ja suora kohdevalo

Kaikki valotyypit jakautuvat vielä kolmeen alakategoriaan static, stationary ja movable. Static light eli staattinen valonlähde toimii vain ennalta lasketussa valaistuksessa, eikä se varjosta liikkuvia esineitä. Staattinen valo on myös tietokoneelle kevyin toistaa, koska valaistus on etukäteen ratkottu, eikä sitä tarvitse muuttaa reaaliajassa. Stationary light eli kiinteä valo on valonlähde, joka valaisee sekä liikkuvia, että staattisia esineitä, mutta ei itse voi liikkua. Kiinteät valonlähteet ovat kuitenkin hyvin raskaita tietokoneelle, koska valaistusta joudutaan osittain ratkomaan reaaliajassa. Movable light eli liikkuva valo on täysin reaaliajassa laskettava valo, joka valaisee liikkuvia ja staattisia esineitä. Se ei kuitenkaan tue valonsäteiden kimpoilua, joten ympäristön realismi kärsii. Ympäristössä, joka pääosin koostuu staattisista esineistä kannattaa siis suosia staattisia valonlähteitä. (Doyle 2018d. osa 9) Jos projekti on ulkoympäristö tai sisätila, joka sisältää ikkunoita, voi valaistusta vielä parannella lisäämällä auringon ja taivaan valoa. Yllä mainittu suora kohdevalo on suositeltava valonlähde auringonvalolle ja esimerkiksi ”Sky Light” -elementillä on mahdollista simuloida taivaan valoa. Tällä elementillä voidaan sijoittaa taustalle 360 asteen kuva (HDRI) valaisemaan ympäristöä. Kun tarvittavat valaistus elementit on lisätty, on staattinen valaistus laskettava.

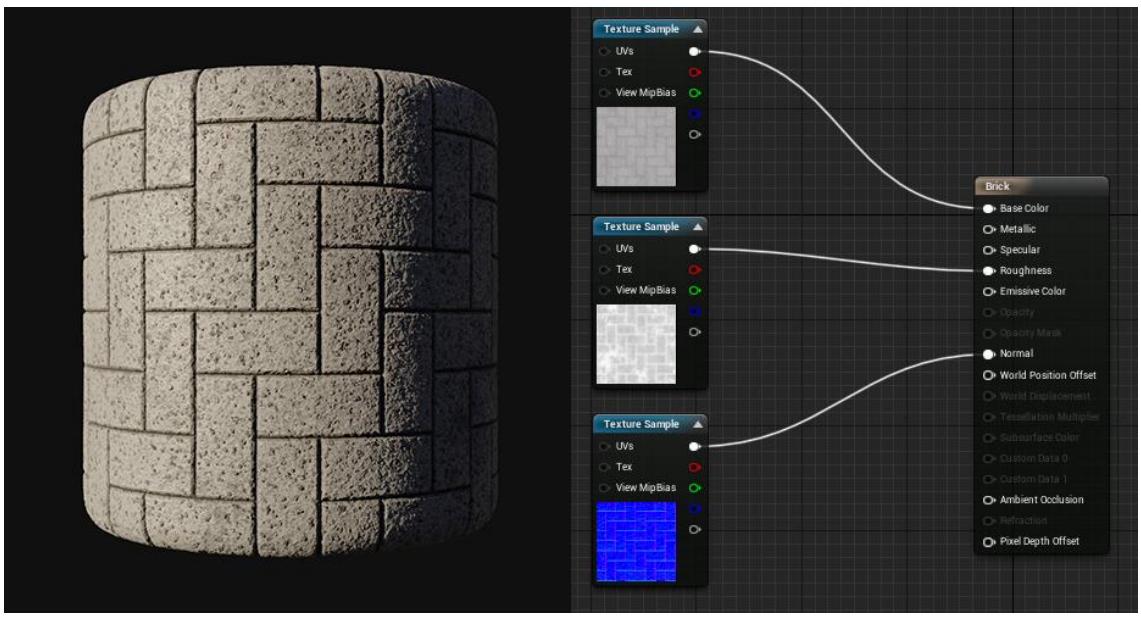
Ensimmäisillä laskentakerroilla on kannattavaa käyttää joko esikatselu tai keskiraskasta kuvalaatua, koska tätä korkeammilla laaduilla voi laskenta kestää hyvin kauan. Ensimmäisten testilaskentojen jälkeen voi tarkastella valaistuksessa tapahtuneita virheitä,

jotka voivat johtua virheellisistä valokartta UV-koordinaateista. Nämä virheet on mahdollista korjata mallinnusohjelmassa luomalla virheellisille esineille mukautetut valokartta UV-koordinaatit tai rikkomalla elementit pienempiin osiin. Testilaskentojen jälkeen on myös kannattavaa tarkastella vastaako valojen voimakkuudet visualisointiprojektissa käytettyjä voimakkuuksia. (Doyle 2018d osa 11.)

Kun valaistus näyttää oikealta, voi sen laskea korkeammilla asetuksilla. Tämä voi viedä hyvin pitkiä aikoja, mutta Unreal Engine tukee laskennan suorittamista useammalla lähiverkkoon liitetyllä koneella samanaikaisesti. Tätä varten on Unreal Engine asennettava jokaiselle koneelle ja käynnistettävä Swarm Agent ohjelma. Useamman koneen käyttö säästää aikaa ja mahdollistaa nopeamman palautteen valaistuksesta. (Doyle 2018d osa 15)

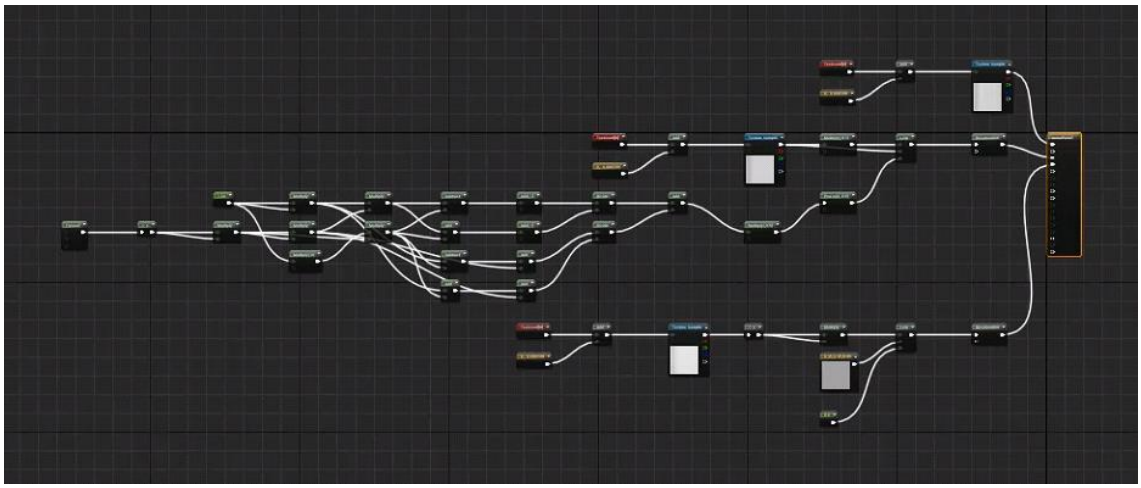
4.3.2 Materiaalit ja tekstuurit

Yksinkertaisimmillaan materiaalin voi käsittää maalina mallin pinnassa. Se on elementti, joka asetetaan mallille ja sillä kerrotaan tietokoneelle, miten valo käyttäytyy osuessaan malliin. Materiaalilla määritellään mallille muun muassa väri, heijastus ominaisuudet ja opasiteetti, joka määrittelee, kuinka paljon valoa pääsee mallin lävitse. Näitä ominaisuuksia voi ajaa numeerisilla arvoilla tai kuvatiedostoilla, jolloin on mahdollista asettaa mallin pintaan erilaisia kuvioita. (Unreal Engine n.d. c.)



Kuvio 8. Materiaaliesimerkki yksinkertaisesta materiaalista lieriön pinnalla.

Mallinnusohjelmasta Datasmith-työkalulla tuodut materiaalit muuttuvat automaattisesti Unreal Engine -materiaaleiksi. Datasmith yrittää muuttaa mallinnusohjelmassa käytetyt materiaalit mahdollisimman saman näköisiksi, jolloin materiaalin kaavio ei välttämättä ole yhtä yksinkertainen kuin yllä esitetyssä kuvassa. Datasmith tukee 3ds Max -natiivi-materiaalien lisäksi muun muassa V-Ray, Corona, Arnold ja Mental Ray materiaaleja. Materiaalien mukana siirtyvät myös tekstuurikartat, joista Datasmith tukee kaikkia yleisimpiä kuvatiedostomuotoja. Mallinnusohjelmasta tapahtuneen viennin yhteydessä, Datasmith ilmoittaa ei-tuetuista materiaali- ja tekstuuriformaateista.



Kuvio 9. Corona-materiaali, jonka Datasmith-työkalu on muuttanut Unreal Engine -materiaali-muotoon.

Kohoumakartat, joilla luodaan illuusio pinnan epätasaisuudesta, Datasmith muuttaa automaattisesti normaalikartoiksi, jotka luovat saman illuusion, mutta ovat helppolukuisempia pelimoottoreille. Usein mallinnusohjelmissa luotujen materiaalien heijastukset pohjautuvat pinnan sileyteen, jolloin täysin valkoinen sileyskartta tarkoittaa täysin terävää heijastusta. Unreal Engine -materiaalien heijastusta ajaa sileyskartan sijaan karheuskartta. Tämän takia Datasmith kääntää sileyskartan värit päinvastaisiksi. Metalli materiaaleja Datasmith ei osaa muuntaa oikein, koska mallinnusohjelmissa pinnan metallisuutta ei ohjata metallikartalla. Nämä materiaalit on siis korjattava käsin laittamalla Unreal-materiaalin ”Metallic -syöttöön numeerinen arvo tai tarvittaessa metallikartta (engl. metallic map). (Doyle 2018c osa 2.2.)

4.3.3 Heijastukset

Unreal Engine tukee sekä ennalta että reaaliajassa laskettavia heijastuksia. Ennalta lasketut heijastukset ovat tietokoneille kevyitä, eivätkä vaadi näytönohjaimelta paljon laskentatehoa. Reaaliaikaisten heijastusten (engl. Screen Space Reflections) esittäminen vaatii näytönohjaimelta enemmän laskentaa, mutta kuluttavat vähemmän muistia. Ennalta lasketut heijastukset heijastavat vain staattista ympäristöä eikä liikkuvia esineitä kuten pelihahmoja, mutta reaaliaikaiset heijastukset peilaavat myös liikkuvia kappaleita. Näitä kahta heijastus menetelmää on mahdollista käyttää samanaikaisesti ja näin pääsee myös realistisimpiin lopputuloksiin. (Unreal Engine n.d. d.)

Jotta ennalta laskettuja heijastuksia voi hyödyntää, on ympäristöön lisättävä elementtejä, jotka määrittelevät heijastuspisteiden kohdat. Heijastus elementtejä on kaksi erilaista pallo (engl. Sphere Reflection Capture) ja särmiö (engl. Box Reflection Capture). Pallon muotoinen heijastuspiste sieppaa kuvan staattisesta ympäristöstä pallon muodossa ja särmiö särmiön muodossa. Kaapatut kuvat voidaan sitten projisoida heijastavien mallien pintaan luoden illuusion valon peilautumisesta. Jotta heijastukset vastaisivat mahdollisimman hyvin kameran havaitsevaa ympäristöä, peilaava esine käyttää aina lähintä heijastus elementtiä. Heijastus elementtejä kannattaa siis olla lukuisia tilaa kohden, jotta peilaukset vastaisivat mahdollisimman hyvin havaittua ympäristöä. Unreal Engine asettaa oletuksena heijastukset 128x128 pikselin kokoon, mutta realismia voi lisätä kasvattamalla heijastusten resoluutioasetusta. (Doyle 2018d osa 12.)

Kuvio 10. Heijastuselementtien vaikutusta on mahdollista tarkastella heijastuskuvamoodilla (engl. Reflections View Mode).

4.3.4 Jälkikäsittely ja erikoistehosteet

Jälkikäsittelyllä tarkoitetaan kuvan päälle luotavia erikoistehosteita. Nämä erikoistehosteet luodaan kuvan viimeisessä laskentavaiheessa ja mahdollistavat kuvan muokkauksen samaan tapaan, kuin valokuvan muokkaamisen kuvankäsittelyohjelmassa. Erikoistehosteilla voi esimerkiksi korjata tilan värimaailmaa, asettaa syväterävyysalueita (engl. Depth of Field) tai simuloida valon hehkumista. Unreal Engine ohjelmassa erikoistehosteita voi lisätä, joko yksittäisille kameroille tai luomalla ympäristöön jälkikäsittely objekteja (engl. Post Process Volume). Jälkikäsittely objektit ovat käytännöllisiä, koska niitä voi ympäristössä olla monta samanaikaisesti. On esimerkiksi mahdollista luoda jokaiselle huoneelle oma jälkikäsittelynhallinta. (Doyle 2018a osa 5.1.)

Kameran linssiin voi myös lisätä erilaisia tehosteita, joiden avulla on mahdollista luoda illuusio oikean kameran käyttäytymisestä. Näitä efektejä on esimerkiksi automaattinen valotus (engl. Auto Exposure), linssin heijastus (engl. Lens Flare), linssin väriväristys (engl. Chromatic Aberration) ja syvyysepätarkkuus (engl. Depth of Field). Usein linsseihin kohdistuvat tehosteet ovat hyvin raskaita, joten niiden liiallinen käyttö voi vaikuttaa kuvataajuuteen. Jos päätelaitteena on virtuaalilasit, ei kannata käyttää kameran linssi erikoistehosteita, koska virtuaalilaseilla on tarkoitus simuloida silmän käyttäytymistä. Kun päätelaitteena käytetään tavallista ruutua, voi kameran linssiefekteillä kuitenkin kasvattaa projektin lopullista realismia. (Doyle 2018a osa 5.2.)

Kuvio 11. Tila ennen jälkikäsittelyä ja sen jälkeen.

4.3.5 Toiminnallisuus

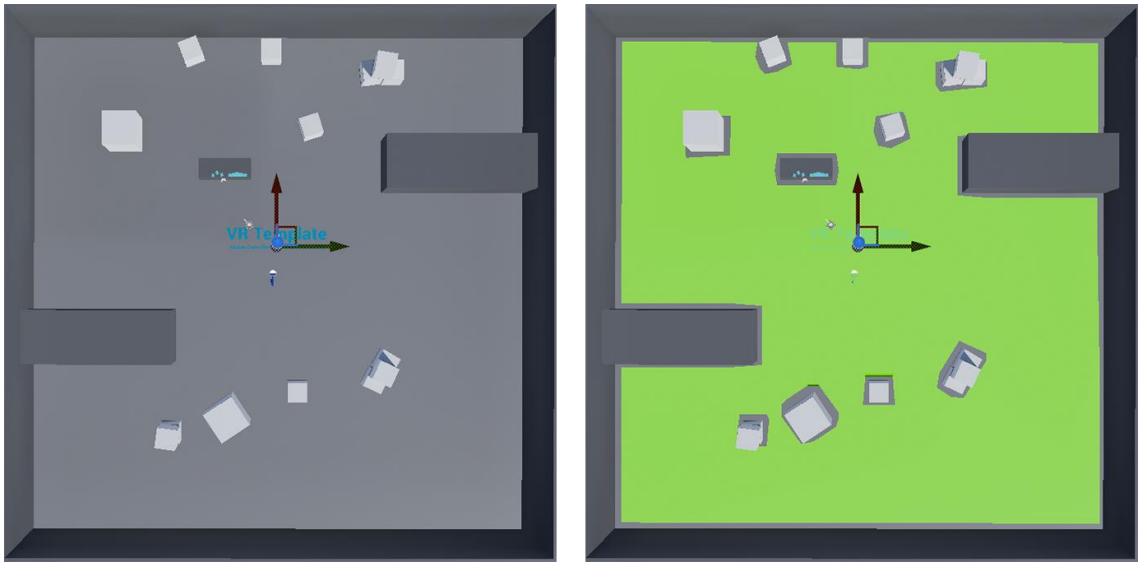
Projektin päätelaitteesta riippumatta täytyy ympäristölle määritellä törmäyksenhallinta, ettei havaitsija joudu ympäristön ulkopuolelle tai kävele seinien läpi. Törmäyksenhallintaa ei tarvitse luoda jokaiselle yksityiskohdalle, vaan pelkästään suurimmille ja olennaisille kappaleille tilassa. Näitä elementtejä ovat esimerkiksi lattia, seinät, katto ja suurimmat huonekalut kuten tuolit, pöydät ja sängyt. Törmäyksenhallinnassa on mahdollista käyttää mallien omaa geometriaa, mutta tämä voi olla tietokoneelle hyvin raskasta. Tämän takia on suositeltavaa tehdä objekteille erilliset törmäysobjektit, jotka ovat kevennettyjä versioita alkuperäisistä malleista. Kevennettyjä törmäysobjekteja voi tehdä, joko mallinnusohjelmassa tai Unreal Engine -ohjelmassa. (Doyle 2018d osa 19.)

Kuvio 12. Erilaisia vaihtoehtoja Unreal Engine -ohjelman luomista automaattisista törmäyksentunnistus elementeistä.

Jos ympäristöä on tarkoitus tarkastella tietokoneen ruudulta, helpoin tapa toteuttaa liikkumisominaisuudet on First Person -mallipohjan käyttäminen. Tämä pohja sisältää malliesimerkin yksinkertaisesta ensimmäisen persoonan ampumispelistä, mutta soveltuu hyvin myös tilaesittelyihin. Kun pelihahmo on siirretty haluttuun aloituspisteeseen, on tilaa jo mahdollista tarkastella, mutta ampumispelissä mukana tulevat ampumis- ja hyppimisominaisuudet on kuitenkin järkevää poistaa, että tilaesittely tuntuu ammattimaiselta. (Doyle 2018d osa 20.)

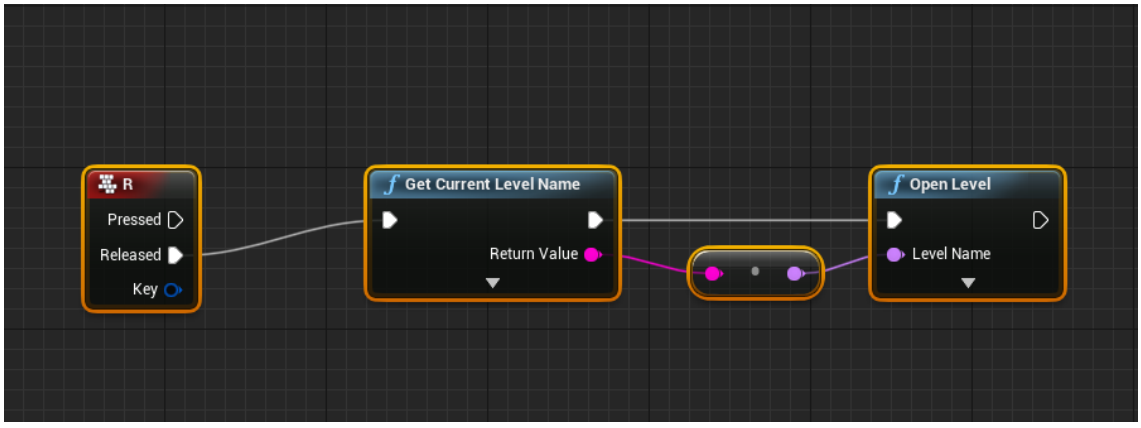
Helpoin tapa käyttää HTC Vive- tai Oculus Rift -virtuaalilaseja päätelaitteena tilaesittelyssä, on tehdä projekti Virtual Reality -mallipohjaan. Mallipohjassa on kummallekin laitteelle omat pelikentät, jotka sisältävät yksinkertaiset ominaisuudet liikkumiseen ja esineistä tartuntaan. Pelihahmon aloituspiste on jälleen siirrettävä haluttuun kohtaan, mutta tällä kertaa kameraa ei kannata laittaa silmän korkeudelle, vaan samalle tasolle kuin lattia. Tämä johtuu siitä, että virtuaalilaseja käytettäessä kamera nousee automaattisesti käyttäjän todelliseen korkeuteen. Unreal Engine -ohjelma tarvitsee vielä tiedon mahdollisista alueista, joilla käyttäjä voi liikkua. Virtual Reality -mallipohjassa tähän käytetään Unreal Engine -ohjelman navigaatio dataa. Navigaatio data generoituu automaattisesti törmäysobjekteista erillisen ”Nav Mesh Bounds Volume” -elementin sisäpuolelle, joten

asettamalla tämän elementin ympäröimään tilaa mahdollistaa virtuaalilaseilla liikkumisen. (Learn Arch Viz 2017b.)

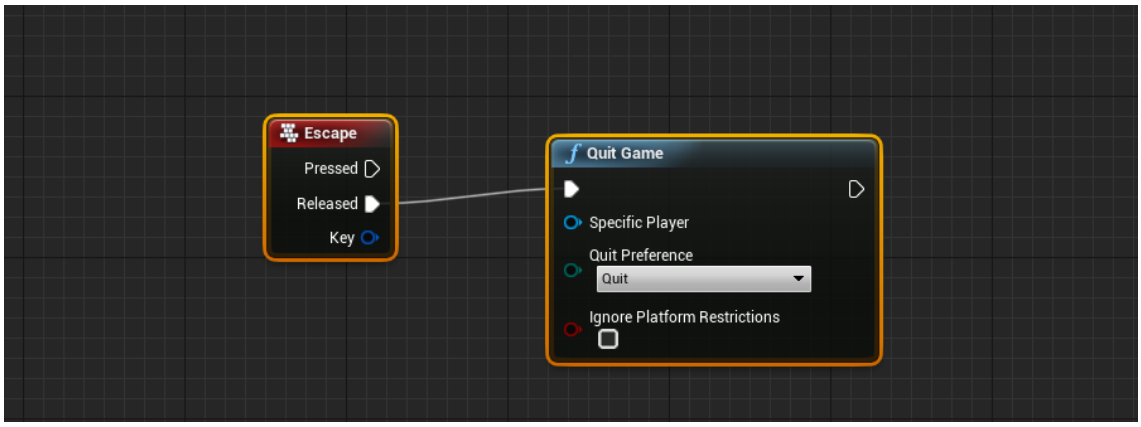


Kuvio 13. Vasemmalla pelikenttä ja oikealla siitä generoitu navigointi verkko.

Tilaesittelyyn on vielä kannattavaa lisättävä näppäin poistumista ja uudelleen käynnistämistä varten. Tason uudelleen käynnistäminen on hyödyllinen ominaisuus, jos esimerkiksi pelihahmo jää jumiin tai tapahtuu jotain muuta odottamatonta.



Kuvio 14. Yksinkertainen skripti, joka aloittaa tilaesittelyn uudestaan R-näppäintä painettaessa.



Kuvio 15. Yksinkertainen skripti, joka sulkee ohjelman ESC-näppäintä painettaessa.

5 Pohdinta

Aiemmin suuressa käytössä ollut tapa siirtää projekti pelimoottooriin oli käyttämällä FBX-formaattia. Tähän verrattuna Datasmith-formaatti on selkeästi parempi, koska Datasmith osaa tuoda Vray- ja Corona-pohjaisia materiaaleja ja valoja. FBX-formaatti tukee pelkästään 3ds Max -ohjelman Standard-materiaalityyppiä. Tämän lisäksi Datasmith tukee lukuisia eri CAD-formaatteja, mikä mahdollistaa projektin tuonnin suoraan arkkitehtien tuottamista suunnitelmista. Tulevaisuudessa Unreal Studio -ohjelman on tarkoitus tukea lukuisia muita formaatteja, kuten Autodesk Mayaa, Autodesk Revitiä, Cinema 4D:tä ja Sketchupia (Unreal Engine Documentation 2018). Käyttämällä Unreal Studio -ohjelmaa säästää huomattavasti aikaa eikä samoja asioita tarvitse tehdä kuin yhdessä ohjelmassa.

Vielä beetavaiheessa oleva Unreal Studio ei kuitenkaan ole täydellinen työkalu. Se sisältää vielä runsaasti virheitä ja ongelmia. Muun muassa valojen voimakkuus, tekstuurien koko ja metallimateriaalit eivät aina näytä siirtyvän oikein 3ds Max -ohjelmasta. Beetavaiheen loputtua uskon ongelmia olevan silti huomattavasti vähemmän, ja jo tässä vaiheessa Datasmith-työkalua käytettäessä säästyy satoja työtunteja verrattuna FBX- tai OBJ-tiedostoformaatteihin. Alla on vielä kuvia lopullisista visualisointiprojekteista, jotka on muutettu reaaliaikaisiksi ympäristöiksi.

Kuvio 16. Hotellihuone visualisointi, joka on mallinnettu 3ds Max -ohjelmalla ja renderöity Corona-moottorilla. (Visualisointitoimisto 3D Taito Oy n.d. a.)

Kuvio 17. Kuvakaappaus Unreal Engine -ohjelmalla luodusta hotellihuoneympäristöstä reaaliajassa.

Kuvio 18. Hotellihuone visualisointi, joka on mallinnettu 3ds Max -ohjelmalla ja renderöity Corona-moottorilla. (Visualisointitoimisto 3D Taito Oy n.d. b.)

Kuvio 19. Kuvakaappaus Unreal Engine -ohjelmalla luodusta hotellihuoneympäristöstä reaaliajassa.

Lähteet

Autodesk FBX 2017. FBX Node Attributes [Verkkodokumentti] <https://help.autodesk.com/view/FBX/2017/ENU/?guid=__files_GUID_FCD71AFF_F753_4D58_B96E_453ED84BF5C4_htm> Viitattu 20.1.2019

Chakravorty, Dibya 2019. 8 Most Common 3D File Formats in 2019 [Verkkodokumentti] <https://all3dp.com/3d-file-format-3d-files-3d-printer-3d-cad-vrml-stl-obj/> Viitattu 7.3.2019

Doyle, Matt 2018a. ARCHITECTURAL EXTERIOR REAL TIME WORKFLOWS. Katsottavissa osoitteesta <https://academy.unrealengine.com/Class-Viewer/Arch_Exterior> Viitattu 23.2.2019.

Doyle, Matt 2018b. PREPARING DESIGN DATA FOR OPTIMAL PERFORMANCE. Katsottavissa osoitteessa <https://academy.unrealengine.com/Class-Viewer/DataPrep_Arch> Viitattu 23.2.2019.

Doyle, Matt 2018c. DATASMITH PIPELINE Katsottavissa osoitteessa <https://academy.unrealengine.com/Class-Viewer/datasmith-pipeline> Viitattu 4.3.2019.

Doyle, Matt, 2018d. EPIC ARCHITECTURAL INTERIOR REAL TIME PROJECTS Katsottavissa osoitteessa <https://academy.unrealengine.com/Class-Viewer/arch_interiors> Viitattu 4.3.2019.

Dukes, Robert 2016. 360 Architectural 3D Tours by Robert Dukes [Verkkodokumentti] <https://www.ronenbekerman.com/360-architectural-3d-tours-robert-dukes/> Viitattu 7.3.2019

Epic Games 2018. Assets [Verkkodokumentti] <https://www.unrealengine.com/en-US/studio/features/assets> Viitattu 6.5.2018.

Franczak, Michal 2017. Preparing Unity for arch-viz work [Verkkodokumentti] <https://evermotion.org/tutorials/show/10810/preparing-unity-for-arch-viz-work> Viitattu 22.1.2019

Higgwe, Samuel n.d. Twinmotion 2018 vs Lumion 8.0 [Verkkodokumentti] <https://lets-designstudios.com/twinmotion-2018-vs-lumion-8-0/> Viitattu 28.1.2019

Learn Arch Viz 2017a. Learn Arch Viz: What It Is, and How Do I Do it? [Verkkodokumentti] <https://www.learnarchviz.com/single-post/2017/03/11/Learn-Arch-Viz-What-Is-It-and-How-Do-I-Do-It Viitattu> 3.2.2019

Learn Arch Viz 2017b. VR tutorial: Introduction To VR Development in UE4 using The VR Template. [Verkkodokumentti] <https://www.youtube.com/watch?v=htwW_Xf0hDM> Viitattu 6.4.2019

LNG Studios, 2016. The 5 Main Benefits of Using VR for Architecture and Design Today [Verkkodokumentti] <http://www.lngstudios.com/blog/the-5-main-benefits-of-using-vr-for-architecture-and-design-today> Viitattu 16.1.2019.

Maheut, Pierre 2017. OBVIOOS: IMMERSIVE EXPERIENCES FOR REAL-TIME ARCHVIZ [Verkkodokumentti] <https://www.allegorithmic.com/blog/obvioos-immersive-experiences-real-time-archviz> Viitattu 16.1.2019.

Mottle, Jeff 2016. 2016 Architectural Visualization Rendering Engine Survey – RESULTS [Verkkodokumentti] <http://www.cgarchitect.com/2016/11/2016-architectural-visualization-rendering-engine-survey---results> Viitattu 9.2.2019

Mottle, Jeff 2018. 2018 Architectural Visualization Rendering Engine Survey – RESULTS [Verkkodokumentti] <http://www.cgarchitect.com/2018/02/2018-architectural-visualization-rendering-engine-survey> Viitattu 9.2.2019

Nichols, Christopher 2017. HOW TO MOVE V-RAY SCENES BETWEEN APPLICATIONS [Verkkodokumentti <https://www.chaosgroup.com/blog/how-to-move-v-ray-scenes-between-applications> Viitattu 20.1.2019

Parkin, 2018. The Benefits of Virtual Reality in Architecture [Verkkodokumentti] < http://www.parkin.ca/blog/the-benefits-of-virtual-reality-in-architecture/>

Pimetel, Ken 2018, CGarchitect survey shows shift to real-time rendering [Verkkodokumentti] <https://www.unrealengine.com/en-US/blog/cgarchitect-survey-shows-shift-to-real-time-rendering> Viitattu 9.2.2019

Ronai, Andras 2017. The Pros and Cons of Using the Unreal Engine for Archviz Animations [Verkkodokumentti] <https://andrasronai.com/2017/06/the-pros-and-cons-of-using-ue4-for-archviz-animations/> Viitattu 16.1.2019.

Simmons, Thomas n.d. Which Format is Better – FBX or OBJ? [Verkkodokumentti] < http://aecobjects.com/2014/10/which_format_is_better/Z> Viitattu 20.1.2019.

Unity n.d.a Build once, deploy anywhere. [Verkkodokumentti] <https://unity3d.com/unity/features/multiplatform> Viitattu 22.1.2019

Unity n.d.b Coding in C# in Unity for beginners. [Verkkodokumentti] <https://unity3d.com/learning-c-sharp-in-unity-for-beginners> Viitattu 30.3.2019

Unreal Engine 2019a. Setting Up Automatic LOD Generation [Verkkodokumentti] < https://docs.unrealengine.com/en-us/Engine/Content/Types/StaticMeshes/HowTo/AutomaticLODGeneration> Viitattu 9.2.2019

Unreal Engine Documentation 2018. Datasmith Supported Software and File Types [Verkkodokumentti] <https://docs.unrealengine.com/en-us/Studio/Unreal-Datasmith/Datasmith-Supported-Software-and-File-Types> Viitattu 6.5.2018.

Unreal Engine n.d.a FREQUENTLY ASKED QUESTIONS (FAQ) [Verkkodokumentti] <https://www.unrealengine.com/en-US/faq> Viitattu 4.3.2019

Unreal Engine n.d.b Types of Lights [Verkkodokumentti] <https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/LightTypes> Viitattu 5.3.2019

Unreal Engine n.d.c Materials [Verkkodokumentti] <https://docs.unrealengine.com/en-US/Engine/Rendering/Materials> Viitattu 6.3.2019

Unreal Engine n.d.d Reflections [Verkkodokumentti] <https://docs.unrealengine.com/en-us/Resources/Showcases/Reflections> Viitattu 6.3.2019

Unreal Studio 2018, Epic Games. Verkkolinkki: <https://www.unrealengine.com/en-US/studio> Viitattu 6.5.2018.

Visualisointitoimisto 3D Taito Oy n.d. a. Lapland Hotels Bulevardi, Helsinki. Kuvamateriaali yrityksen sisäisestä kirjastosta, käyttö sallittu.

Visualisointitoimisto 3D Taito Oy n.d. b. Hotel Amerikanlinjen, Oslo. Kuvamateriaali yrityksen sisäisestä kirjastosta, käyttö sallittu.

Kuvia lopullisista projekteista


PUROPLAN
www.mallihuone.fi