



Kare Hyvämäki

Virtuaaliympäristöjen rakentaminen ja optimointi virtuaalituotantoja varten

Metropolia Ammattikorkeakoulu

Medianomi

Viestinnän tutkinto-ohjelma

Opinnäytetyö

2.5.2023

Tiivistelmä

Tekijä(t):	Kare Hyvämäki
Otsikko:	Virtuaaliympäristöjen rakentaminen ja optimointi virtuaalituotantoja varten
Sivumäärä:	52 sivua
Aika:	18.4.2023
Tutkinto:	Medianomi
Tutkinto-ohjelma:	Viestinnän tutkinto-ohjelma
Suuntautumisvaihtoehto:	Liikkuva grafiikka
Ohjaaja(t):	Lehtori Samuli Homanen

Opinnäytetyön aiheena ovat viime vuosina suuresti suosiotaan nostaneet virtuaalituotannot, ja se pyrkii tarjoamaan tarvittavan tietopohjan digitaalisten ympäristöjen rakentamisesta sekä optimoinnista virtuaalituotantoja varten. Tämän työn aihekokonaisuuksia sovelletaan toiminnallisessa osuudessa, jossa kirjoittaja rakentaa virtuaalituotantojen tarpeita vastaavan virtuaalisen ympäristön. Opinnäytetyössä mainittuja optimointikeinoja käytetään kyseisessä virtuaaliympäristössä tavoitteena vertailla niiden vaikutuksia projektissa käytetyn pelimoottorin Unreal Enginen suorituskykyyn.

Aineiston keräämisprosessin aikana käytännön esimerkkien sekä ohjeistojen löytäminen osoittautui haasteelliseksi. Opinnäytetyön aineisto koostuu pääsääntöisesti Epic Gamesin pelimoottorin Unreal Enginen virallisesta dokumentaatiosta sekä laajasta määrästä erilaisten yhteisöjen, ammattilaisten ja harrastajien jakamaa tietoa.

Johtopäätöksenä virtuaalituotannot ovat nopeasti kehittyvä ala, joka vaatii jatkuvaa oppimista sekä osaamisen päivittämistä. Opinnäytetyön aiheiden syvempi ymmärtäminen edellyttää laajaa tietopohjaa 3D grafiikasta, pelimoottoreista sekä elokuvausmenetelmistä. Virtuaaliympäristöjen rakentamiseen ja optimoimiseen on käytettävissä monia keinoja, joiden valitseminen perustuu aina tuotannon tarpeisiin.

Opinnäytetyö antaa kattavan katsauksen uuteen alaan ja toimii oppaana virtuaalituotannoista sekä pelimoottoreista kiinnostuneille. Työ sisältää hyödyllistä tietoa niin vasta-alkajille kuin aiheeseen tutustuneillekin.

Avainsanat:	Virtuaalituotanto, ICVFX, pelimoottorit, LED-volyymi, optimointi, Unreal Engine
-------------	---

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author(s):	Kare Hyvämäki
Title:	Building and optimizing virtual environments for virtual productions
Number of Pages:	52 pages
Date:	18 April 2023
Degree:	Bachelor of Culture and Arts
Degree Programme:	Media
Specialisation option:	Motion graphics
Instructor(s):	Samuli Homanen, Senior Lecturer

The subject of this thesis is virtual productions, which have become increasingly popular in recent years. The goal of this thesis is to offer a required basis of knowledge for building and optimizing virtual environments for the use of virtual productions. The information gathered for the thesis is applied in the assembly of a virtual environment by the writer. Optimization methods mentioned in the thesis are also applied in the environment to compare the effects on the performance of Unreal Engine, the game engine used in the project.

As for the study for the thesis, there appeared to be a shortage of practical examples and guidelines. The main sources of information are official documentation for the game engine Unreal Engine and a large amount of knowledge shared by the Epic Games community that consists of professionals and enthusiasts.

The results showed virtual productions to be a rapidly evolving field which therefore requires constant learning and updating of skills. A deeper understanding of these topics and line of work demands a broad knowledge base of 3D graphics, game engines and filmmaking. There is a variety of methods for building and optimizing virtual environments and the choice of method should always reflect the needs of the production.

The goal of the thesis is to provide a comprehensive introduction of this new field and act as a guide about virtual productions for people interested in building virtual environments and learning how to optimize them. The thesis will offer valuable information for people new to the subject but also for the ones already familiar to it.

Keywords:	Virtual production, ICVFX, game engines, LED volume, optimization, Unreal Engine
-----------	--

Sisällys

1	Johdanto	1
2	Virtuaalituotannot ja pelimoottorit	2
2.1	Virtuaalituotantojen historia sekä kehitys	3
2.2	Mitä ovat pelimoottorit	6
2.3	Pelimoottorit virtuaalituotannoissa	6
2.4	ICVFX ja LED-volyymi	8
3	3D-ympäristön matka luonnoksesta LED-volyymille	13
3.1	Assettien luominen	16
3.1.1	Geometrian mallintaminen	16
3.1.2	Materiaalit ja teksturointi	17
3.2	Valonlähteet	19
3.3	Volyymi ja lavastus	23
4	Optimointi	26
4.1	Alkuvaiheen optimointi ja profilointi	27
4.2	Assettien optimointi	30
4.3	Materiaalien optimointi	33
4.4	Valaistuksen optimointi	34
5	Toiminnallinen osuus	36
5.1	Virtuaaliympäristöni rakentaminen	37
5.2	Rakennetun virtuaaliympäristön esittely	39
5.3	Rakennetun virtuaaliympäristön optimointi	41
6	Yhteenveto	45
	Lähteet	48

1 Johdanto

Pyrin tässä opinnäytetyössä kertomaan tavoista, joilla fotorealistisia virtuaaliympäristöjä voidaan rakentaa Epic Gamesin pelimoottoria Unreal Engineä käyttäen. Tavoitteenani on myös selvittää optimointikeinoja, joiden avulla virtuaaliympäristöt täyttävät virtuaalituotantojen vaatimuksia.

Virtuaalituotantojen ollessa vielä verrattain harvinaisia ja tuntemattomia jopa monille alalla työskenteleville löydettävissä olevan tiedon määrä on myös rajattua. Yksi tämän opinnäytetyön tarkoituksista onkin avata ja esitellä virtuaalituotantojen käyttämää teknologiaa ja selvittää sen parissa työskentelyn haasteita ja mahdollisuuksia. Opinnäytetyö tarjoaa suomenkielistä tutkimusta aiheesta, johon liittyvä tieto löytyy vielä pääasiassa englanniksi.

Työni toiminnallisessa osuudessa rakennan virtuaaliympäristön, joka on tarkoitettu kuvitteellisen virtuaalituotannon käyttöön. Rakennan ja lopuksi optimoin virtuaaliympäristön käyttäen opinnäytetyön teoreettisessa osuudessa esitettyjä keinoja. Opinnäytetyön toiminnallinen osuus havainnollistaa optimointikeinojen vaikutuksia ja tukee teoriaa käytännön esimerkkien avulla.

Opinnäytetyön tietopohja perustuu suurelta osin verkosta löytyviin lähteisiin, kuten Epic Gamesin tarjoamaan dokumentaatioon sekä aiempien virtuaalituotantojen tarjoamiin esimerkkeihin. Verkosta löytyvät harrastajien ja ammattilaisten muodostamat yhteisöt tarjoavat myös runsaasti tukea virallisten aineistojen tulkitsemiseen. Yhteisön tarjoamaa tietoa on tässä tutkielmassa käytettykin enimmäkseen tiedon löytämisen apuna, mutta harkitusti myös suorana lähteenä.

Virtuaalituotantojen syvempi tekninen ymmärtäminen vaatii laajaa käsitystä perinteisistä elokuva- sekä televisiotuotantomenetelmistä ja niiden

yhdistymisestä uuteen pelimoottoritekнологiaan sekä kolmiulotteiseen grafiikkaan. Pyrin rajaamaan aihetta liikkuvan grafiikan opintojeni näkökulmasta ja keskityn ympäristöjen luomisen työvaiheisiin sekä niiden optimointiin pelimoottori Unreal Enginen sisällä.

Vaikka opinnäytetyössä käydään läpi perustavanlaatuisia 3D-grafiikkaan sekä mallinnukseen liittyviä käsitteitä ja työvaiheita, aiheen syvempää ymmärtämistä helpottaa aikaisempi kokemus tai tietämys 3D-mallinnusohjelmien käytöstä ja aihekokonaisuuksista kuten 3D-mallinnus, teksturointi sekä renderöinti.

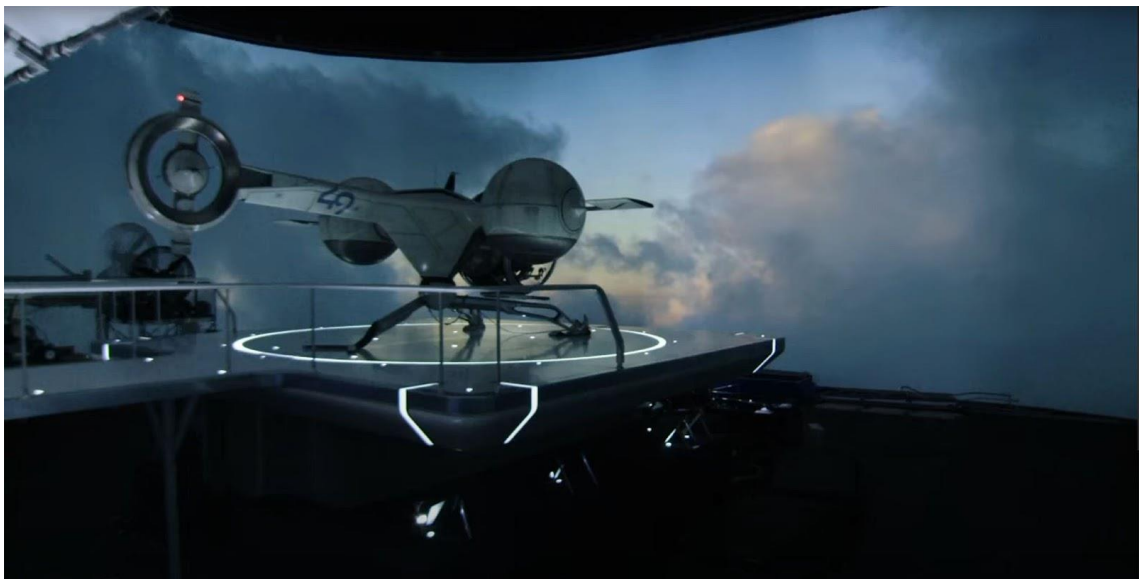
2 Virtuaalituotannot ja pelimoottorit

Virtuaalituotannot sekä niissä käytettävä teknologia on luotu kaventamaan eroa perinteisen elokuvanteon ja tietokonegrafiikkaa käyttävien tuotantojen välillä. Virtuaalituotannot antavat elokuvantekijöille tilaisuuden työskennellä virtuaalisissa kuvausympäristöissä perinteisen elokuvatuotantojen tavoin. Erikoistehosteihin sekä jälkituotantoon keskittyvä Wētā FX kuvaakin verkkosivuillaan virtuaalituotantoja paikaksi, jossa fyysinen ja digitaalinen maailma kohtaavat. Virtuaalituotannot voivat siis tarkoittaa monia asioita, mutta niitä kaikkia yhdistää perinteisten elokuvauksen keinojen yhdistäminen digitaalisiin tapoihin tuottaa tai tarkastella visuaalista materiaalia.

Opinnäytetyössä virtuaalituotannoista puhuessani tarkoitan kuitenkin vain tuotantoja, jossa LED-paneeleista koottua näyttöä käytetään kuvatun kohteen taustana fyysisen lavastuksen jatkeena halutun tilan sekä valaistuksen saavuttamiseksi. Pelimoottorien ja LED-teknologian viime vuosien kehitys on mahdollistanut LED-näyttöjen ja kameran toimimisen yhteistyössä kolmiulotteisten ympäristöjen kanssa luoden illuusion syvyydestä ja yhtenäisestä tilasta studion sekä volyymin välillä. Virtuaalituotannot ovat käyneet kuitenkin läpi monet vaiheet ennen tähän pisteeseen pääsemistä, joten tässä luvussa pyrin antamaan yleiskuvan tuosta matkasta.

2.1 Virtuaalituotantojen historia sekä kehitys

Virtuaalituotantojen käyttämiä peruseriäotteita on hyödynnetty elokuvateollisuudessa jo pitkään, vaikkei teknologia ole ollut nykyisellä tasolla. Varhaisin virtuaalituotannon muoto syntyi jo viime vuosisadalla projisoinnin muodossa, jolloin kuvattavan kohteen taakse asetettiin valkokangas, jolle heijastettiin ennalta kuvattua materiaalia. Usein projisointia käytettiin esimerkiksi ajokohtauksen kuvaamisessa, mutta projektoreiden kehittyessä sekä suurempien ja voimakkaampien projisointien mahdollistuessa tekniikkaa pystyttiin hyödyntämään monipuolisemmin. Esimerkiksi elokuvassa *Oblivion* (2013) käytettiin koko lavasteen ympäri kiertävää valkokangasta, jolle heijastettiin vuoren huipulta kuvattua materiaalia (kuva 1). Elokuvan päähenkilön futuristinen, paljon kiiltäviä pintoja sisältävä asutus oli rakennettu kokonaan tämän kankaan keskelle, mikä mahdollisti oikean sävyiset ja näköiset heijastukset rakennuksen pinnoissa. (Oseman 2022.)

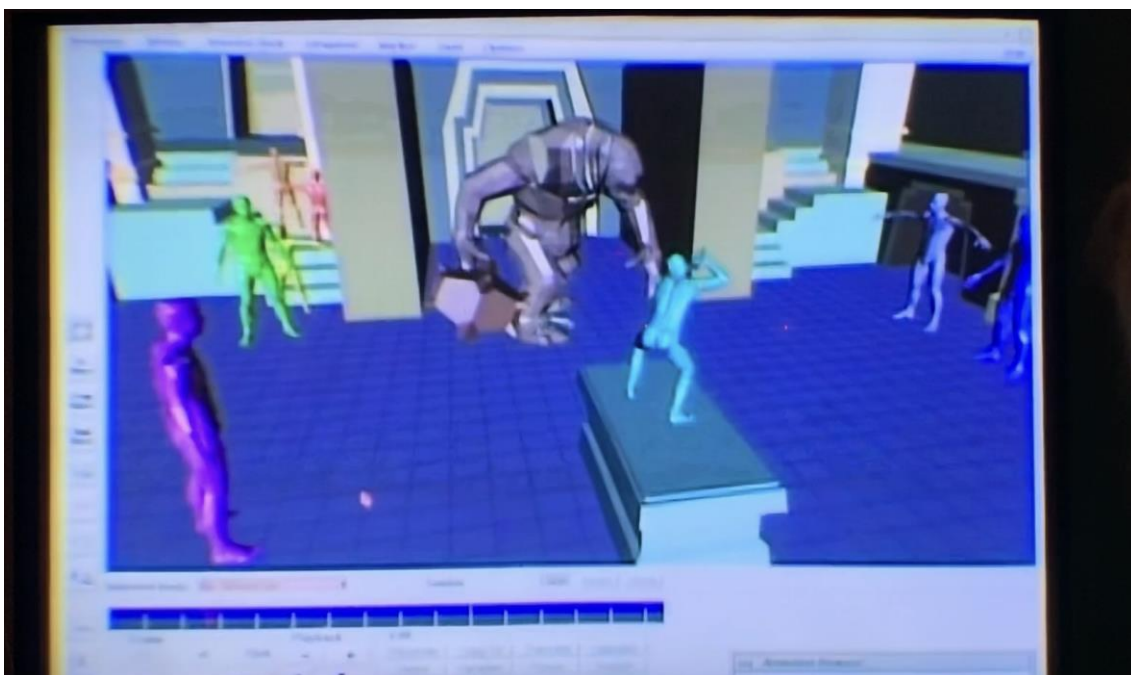


Kuva 1. Oblivion-elokuvan kuvauksissa käytettiin kankaalle projisoitua kuvamateriaalia ympäristön ja valaisun luomiseksi (Universal Pictures All-Access 2013, 0:51)

Digitaalisen virtuaalituotannon uranuurtajana toimi uusiseelantilainen erikoistehostudio Wētā FX, aiemmalta nimeltään Wētā Digital. Elokuvatrilogian

Taru Sormusten Herrasta (2001–2003) erikoistehosteista vastaava yritys kehitti Sormusten Ritarit -elokuvaa tehdessään vuonna 1999 ensimmäisen tunnetun muodon digitaalisesta virtuaalituotannosta. Yritys etsi keinoa helpottamaan kohtauksen kuvaamista, jossa digitaalinen luolapeikko oli vuorovaikutuksessa oikean lavasteen ja näyttelijöiden kanssa. Ratkaisu löydettiin hyödyntämällä lavasteesta luotua virtuaalista versiota ja liikkeenkaappausta. Liikkeenkaappaus (engl. motion capture) tarkoittaa teknologiaa, jonka avulla tallennetaan näyttelijän tai esineen liikettä, joka voidaan sitten siirtää kolmiulotteiseen tilaan digitaaliselle hahmolle. Liikedataa voidaan tallentaa eri keinoin, kuten kohteeseen kiinnitettävillä sensoreilla tai merkeillä, joiden liikettä kamera seuraa ja tallentaa.

Jokaisen luolapeikkokohtauksessa esiintyvän hahmon karkea liike sekä sijainti siirrettiin lavasteesta luotuun virtuaaliseen versioon (kuva 2), jolloin kohtauksen koreografia saatiin jäljiteltä. Vaikka liikkeenkaappausta oli käytetty aiemminkin realistisen liikkeen siirtämiseksi digitaaliselle hahmolle, Wētā FX oli ensimmäinen studio, joka käytti liikkeenkaappausta myöskin virtuaalisen kameran luomiseksi. Ohjaaja Peter Jackson piteli käsissään keppiä, jonka päähän kiinnitetyn kameraa jäljittelevän esineen liike tallennettiin ja siirrettiin reaaliajassa virtuaaliseen ympäristöön. Peter Jackson kykeni samaan aikaan virtuaalilasien avulla näkemään virtuaalikameran näkymän, jonka liikettä hän hallitsi. Tämä mahdollisti kuvauksen suunnittelun virtuaalimaailmassa, mitä voitiin hyödyntää oikeissa kuvauksissa. Peter Jackson (2001) kuvaili, kuinka tämä mahdollisti digitaalisen hirviöhahmon kuvaamisen samalla tavalla kuin konkreettisesti lavastuksessa oikealla kameralla.



Kuva 2. Taru Sormusten Herrasta - Sormusten Ritarit elokuvassa kehitettiin ensimmäistä kertaa virtuaaliympäristö, jossa ohjaaja pystyi virtuaalikameran avulla suunnittelemaan kohtauksen kuvaamista. Ruutukaappaus elokuvan Sormusten Ritarit lisämateriaalista: Part Two - From Vision To Reality (2001).

Wētā FX jatkoi liikkeenkaappausteknologian lisäksi virtuaalikameran kehittämistä James Cameronin ohjaaman *Avatarin* (2009) kuvauksissa. Elokuvaa varten kehitettiin teknologiaa, jota alettiin kutsua nimellä *performance capture*. Motion capturesta eroten näyttelijöiden liikkeen lisäksi heidän ilmeensä tallennettiin, mikä mahdollisti roolisuorituksen kokonaisvaltaisemman hyödyntämisen jälkituotannossa. Tämän lisäksi kuvauksia varten luotu teknologia mahdollisti liikedatan reaaliaikaisen käyttämisen, ja tuotannossa käytettävän virtuaalimonitorin ansiosta ohjaaja pystyi jo kuvauksien aikana näkemään karkean version näyttelijöiden liikkeen ohjaamista hahmoista rakennetussa virtuaaliympäristössä. Monet nykypäivän visualisointia ja digitaalisia hahmoja sisältävät tuotannot käyttävät pitkälti samanlaista teknologiaa kuin Sormusten Ritarit - ja Avatar-elokuvat, mutta laitteiston kehittyminen ja reaaliaikaisen renderöintitehon kasvaminen pelimoottorien käytön myötä on tuonut visuaalisen laadun lähemmäs tavoiteltua lopputulosta.

2.2 Mitä ovat pelimoottorit

Pelimoottorit ovat pelien luomiseen tarkoitettuja alustoja, jotka koostuvat pelien eri toiminnallisuuden mahdollistavista ohjelmistoista. Nämä ohjelmistot eli komponentit on ohjelmoitu esimerkiksi renderöimään kattavia määriä kolmiulotteisia elementtejä, analysoimaan pelaajan päätöksentekoa tekoälyn avulla tai simuloimaan realistista fysiikkaa. Pelimoottoreiden sisältämien valmiiden komponenttien avulla uusia pelejä on mahdollista luoda yksinkertaisemmin ja nopeammin. Pelimoottorit rakentuvat usein esimerkiksi komentokielestä, renderöintimoottorista, fysiikkamoottorista, äänistä, visuaalisista efekteistä sekä tekoälystä, jotka kaikki vaikuttavat pelaajan pelikokemukseen. (Gregory 2014.) Juurikin monikäyttöisyytensä ja reaaliaikaisen renderöinnin mahdollistavan renderöintimoottorinsa ansiosta pelimoottorit ovat löytäneet paikkansa myös elokuvateollisuudessa.

Pelimoottoreita on monia, joista nykypäivän tunnetuimpina esimerkkeinä toimivat Cryengine, Godot, Unity ja Epic Gamesin Unreal Engine. Viimeiseksi mainittu on pelialan lisäksi vakiinnuttanut asemansa johtavana virtuaalituotannoissa käytettävänä pelimoottorina, ja tästä syystä keskityn opinnäytetyössäni ympäristöjen optimoimiseen juurikin Unreal Enginessä. Vuonna 1991 perustettu Epic Games, Inc. aloitti matkansa peliteollisuudessa julkaisemalla pelinsä Unreal vuonna 1998 ensimmäisen pelimoottorinsa yhteydessä (Gregory 2014, 27). Viime vuosina yhtiön kehittämä pelimoottori Unreal Engine on nostanut suosiotaan alalla tarjoamiensa työkalujen tehokkuuden, kattavuuden sekä helppokäyttöisyyden ansiosta, ja esimerkiksi vuoden 2022 pelatuin peli Fortnite mukaanlukeutuu Epic Gamesin tuotantoon (Perforce.com 2023).

2.3 Pelimoottorit virtuaalituotannoissa

Pelimoottoreita alettiin hyödyntää virtuaalituotannoissa niiden renderöintitehon vuoksi. Ennen visuaalisen laadun ja realistisuuden parantumista nykyiselle

tasolle pelimoottoreiden reaaliaikainen renderöinti tarjosi nopean ja interaktiivisen tavan tuotantojen suunnitteluun. Pelien luomiseen tarkoitettut pelimoottorit tarjosivat myös valmiit työkalut immersiiivisyyden tuomiseen digitaalisen elokuvanteon maailmaan, jota perinteisistä taustoista tulevat pystyivät hyödyntämään. Esimerkiksi esivisualisoinnissa kaikki nuo pelimoottorien vahvuudet pääsivät loistamaan.

Esivisualistointi eli previs, joka juontuu sanasta previsualization, tarkoittaa kohtausten tai jopa koko elokuvan suunnittelua ja visualisointia ennen varsinaisten kuvausten alkamista. Pelimoottorit tarjosivat esivisualisoinnissa vapauden tarkastella karkeasti rakennettuja virtuaaliympäristöjä sekä animoituja kohtauksia halutusta suunnasta, niin kuin myös mahdollisuuden suunnitella kameran sijainteja sekä linssien käyttöä reaaliajassa. Unreal Engineä käyttivät esivisualisointiin esimerkiksi yritykset Halon Entertainment sekä THE THIRD FLOOR tuotannoissa kuten *The War for the Planet of the Apes* (2017) ja *Game Of Thrones* (2011-2019). Kuvassa 3 havainnollistuu THE THIRD FLOORin käyttämän virtuaaliympäristön tarjoama apu skaalan hahmottamisessa ja rajoituksen suunnittelussa, jolloin jälkituotannossa lisättävät erikoistehosteet tai hahmot voidaan ottaa huomioon jo kuvauksien aikana.



Kuva 3. Game Of Thrones -sarjan esivisualisoinnissa käytettiin virtuaalilaseja ja 3D-ympäristöjä otoksien suunnitteluun. (Unreal Engine 2020, 1:41)

Virtuaalitekniikkaa tutkinut sekä eteenpäin ajanut ohjaaja Jon Favreau kertoo BBC Clickin haastattelussaan vuodelta 2019 The Lion King elokuvan teknisistä harppauksista pelimoottoreiden hyödyntämisen suhteen. Elokuvan tekemistä valmisteltiin kuten mitä tahansa nykyaikaisia animaatioelokuvia suunnittelemalla hahmot sekä mallintamalla ympäristöt kolmiulotteisina, mutta virtuaalitodellisuus tuotiin pelimoottorin avulla ohjaajan ja elokuvauksen tueksi. Pelimoottori Unity sekä VR-teknologia mahdollistivat elokuvaajan Caleb Deschanelin liikkumisen korkealaatuisissa kolmiulotteisissa ympäristöissä, jotka muistuttivat laadultaan jo läheisesti tavoiteltua lopputulosta. Pelimoottoritekniologia mahdollisti sommitelmien ja valojen muokkaamisen reaaliajassa ohjaajan ja kuvaajan halujen mukaiseksi. Esivalmistelujen jälkeen elokuvan kuvaaminenkin suoritettiin studiossa kameran liikkeen tallentamiseksi ja tuomiseksi virtuaaliseen maailmaan. Ohjaaja Jon Favreau selittää haastattelussaan, että kaikki tuo uusi tekniologia mahdollistaa inhimillisen elementin lisäämisen tuotantoihin, jotka tapahtuvat muuten digitaalisessa ympäristössä. 3D-grafiikan suhteen kokemattomien elokuva-alan tekijöiden siirtyminen perinteisistä tuotannoista virtuaalituotantoihin on helpompi ja intuitiivisempi niiden samankaltaisuuden ansiosta. Favreaun mukaan tämä inhimillisen elementin lisääminen muutoin täysin digitaaliseen tuotantoon onkin arvokkain virtuaalituotantojen mahdollistama edistys. (Favreau 2019.)

2.4 ICVFX ja LED-volyymi

ICVFX, joka on lyhenne termistä *In-Camera Visual Effects*, tarkoittaa tehosteita, jotka pystytään vangitsemaan kuvatessa ilman jälkikäsitteilyä. ICVFX:llä voidaan tarkoittaa perinteisiä tehosteita, joissa esimerkiksi peilien, maalattujen lasilevyjen tai pakotettujen perspektiivien (engl. forced perspective) avulla muokataan katsojan näkemää todellisuutta. Nykypäivänä termillä kuitenkin usein viitataan virtuaalituotantoihin, joissa pelimoottoria käytetään toistamaan reaaliaikaisesti fotorealistisia kolmiulotteisia ympäristöjä, jotka toistetaan yhden tai useamman prosessorin avulla lavasteiden taakse asetetulla LED-volyymilla. (Perforce.com 2023.) Kuvauksissa käytettävä kamera synkronoidaan sitten

pelimoottorissa olevan virtuaalikameran kanssa, jolloin kameran liike ohjaa taustalla olevan ympäristön liikettä luoden illuusion tilan syvyydestä (L'Italien 2022).

LED-volyymi (engl. LED volume) on nimitys LED-seinän tai seinien sekä lavasteen rajaamalle tilalle, jota käytetään ICVFX-virtuaalituotannoissa. Kuvassa 4 on nähtävissä The Mandalorian -sarjan (2019) lavasteet ja niitä ympäröivät LED-seinät, joilla näkyvä horisontti sekä taivas toistetaan. LED-seinät koostuvat toisiinsa lukittuvista LED-paneeleista, jotka yhdessä laitteiston ja ohjelmiston kanssa muodostavat virtuaalinäytön (Kadner 2021).



Kuva 4. Esimerkkejä The Mandalorian -sarjassa käytetystä LED-volyymistä. (Industrial Light & Magic 2019)

LED-paneeleita on käytetty elokuvien tekemisessä jo ennen volyymin kehittämistä. Taustojen sijasta niitä käytettiin kuitenkin dynaamisen valaisun

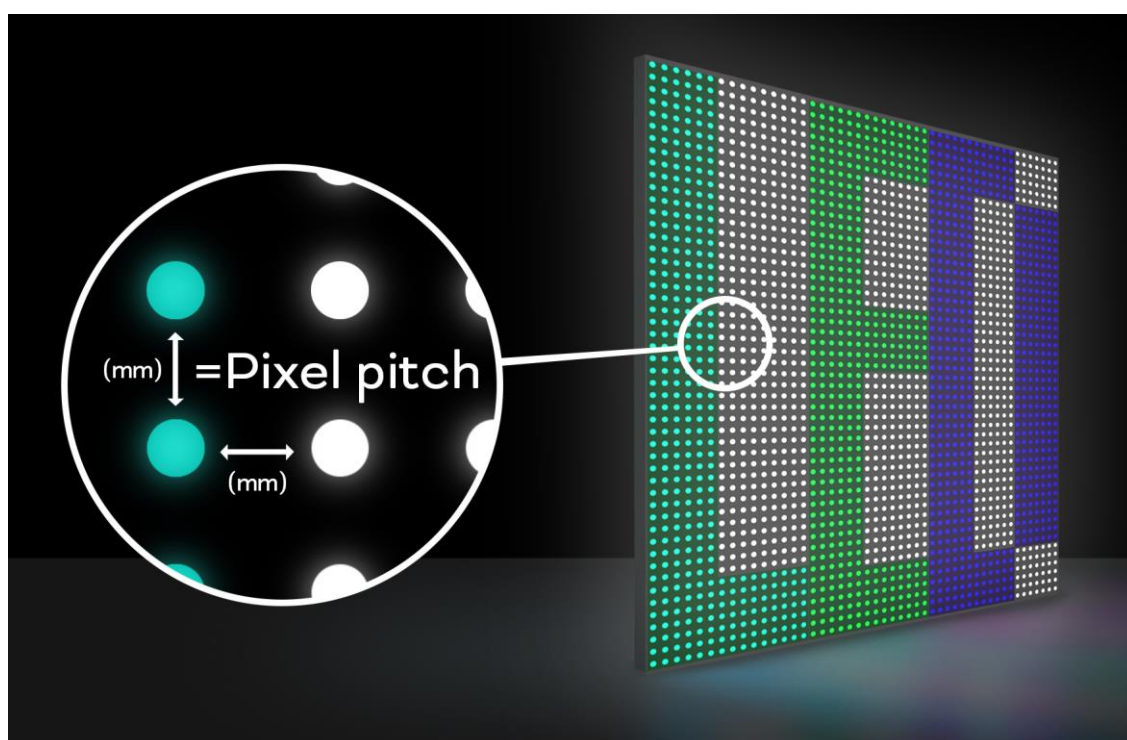
luomiseen. LED-paneelien välkkymistä hallitsemalla pystyttiin jäljentämään liikkuvaa ympäristöä, ja LED-valojen värien käyttö mahdollisti täsmällisemmän valaisun. Avaruuteen sijoittuvassa elokuvassa *Gravity* (2013) näyttelijät laitettiin LED-paneeleista rakennettuun kuutioon (Kuva 5), jolloin LED-valot mahdollistivat avaruudessa kieppuvien astronauttien valaisun jäljittelyn. Teknologia ei vielä silloin kuitenkaan mahdollistanut tarpeeksi tiheään asetettuja LED-valoja, jotta niitä olisi voitu käyttää kuvattaessa myös valmiina ympäristönä. Näin ollen ne jouduttiin korvaamaan jälkikäsittelyssä. (Oseman 2022.)



Kuva 5. Gravityn kuvauksissa valaisun luomiseen käytettiin LED-paneeleista koottua kuutiota (Brooks 2019)

LED-paneelien valojen tiheys, josta käytetään termiä *pixel pitch*, on ollut ja on vieläkin yksi LED-volyymien tärkeimmistä teknisistä ominaisuuksista, joka vaikuttaa sen käyttömahdollisuuksiin. Pixel pitch kertoo millimetreissä, kuinka

kaukana yksittäiset LED-valot ovat toisistaan paneelissa (kuva 6) (Kadner 2021). Mitä tiheämpi väli on, sitä yhtenäisemmältä seinä näyttää kameran läpi katsottuna ja sitä tarkempaa volyymia voi kuvata kameralla, ilman että se paljastuu näytöksi. Rogue One: A Star Wars Story -elokuvassa vuodelta 2016 käytettiin 9 millimetrin pixel pitchin omaavia LED-paneeleita luomaan hyperajoeffekti ja jo silloin tausta voitiin pitää kuvassa ilman jälkikäsitteilyä, kunhan syväterävyys oli tarpeeksi suppea eikä tausta ollut tarkennettuna. (Oseman 2022.)



Kuva 6. Pixel Pitch kuvaa millimetreissä LED-paneelissa olevien valojen etäisyyttä toisistaan.

Nykypäivänä saatavilla olevien LED-paneelien pixel pitchit vaihtelevat jopa 1,5 ja 3 millimetrin välillä. LED-valojen tiheyden parannuttua viime vuosikymmenen lopulla niiden edessä pystyttiin alkamaan kuvaamaan näyttelijöitä ja muita kohteita, kunhan kamera tarkennettiin kohteeseen eikä näyttöön. Kehityksestä huolimatta LED-paneelien pixel pitchit eivät vielä mahdollista niihin tarkentamista kameralla siitä aiheutuvan moiré-vaikutuksen takia. Moirélla

tarkoitetaan virtuaalinäyttöä kuvatessa ilmentyvää optista häiriötä, joka johtuu näytön alhaisemmasta resoluutiosta verrattuna kameran sensoriin (Kadner 2021). LED-paneelien muodostaman näytön resoluutio vastaa siinä olevien LED-valojen määrää.

LED-teknologian kehittyessä paneelit, jotka olivat aikaisemmin tasaisia, pystyttiin suunnittelemaan lähes mihin tahansa haluttuun muotoon. Tämä mahdollisti suurten, lavastetta ympäröivien LED-seinien rakentamisen, mikä auttoi luomaan aidomman interaktiivisen valaistusympäristön heijastuksineen. Vuoden 2019 kesällä järjestetyssä SIGGRAPH-tietokonegrafiikkakonferenssissa nähtiin lukuisia esityksiä koskien Unreal Engine -pelimoottorin käyttömahdollisuuksia virtuaalituotannossa. Pelimoottorien ja LED-volyymien yhteistyön kannalta kuitenkin konferenssin tärkein hetki oli Epic Gamesin Kim Librerin esittelemä video, jossa Epic Games yhteistyössä elokuva- ja 3D-grafiikka-alan yritysten kanssa esitteli tulevan Unreal Engine -version tarjoamia LED-volyymien käyttöön tarkoitettuja työkaluja. Esimerkiksi Epic Gamesin kehittämä nDisplay -lisäosa mahdollisti Unreal Enginen kolmiulotteisen materiaalin toiston eri muotoisilla sekä kokoisilla näytöillä. Jo 2018 ilmestynyt, pääosin visuaalisia installaatioita varten kehitetty nDisplay päivitettiin vastaamaan LED-volyymien kehityksen aiheuttamiin tarpeisiin ja siitä aiheutuviin ongelmiin. LED-paneeleilla, kuin myös projektoreilla työskennellessä toimiva nDisplay-lisäosa skaalaa saumattomasti sekä synkronoi sisällön halutulle päätteelle ottaen huomioon myös sen muodon vääristymien välttämiseksi. (Dalkian 2021.)

The Mandalorian -sarjan virtuaalituotannosta vastaava Industrial Light & Magic (ILM) suunnitteli ja toteutti sarjaa varten yhden ensimmäisistä lähes 360 astetta kattavista LED-volyymeistä yhdessä Epic Gamesin, Fuse Technical Groupin sekä Profile Studiosin kanssa. Kehitettyä virtuaalituotantoteknologiaa kutsuttiin nimellä StageCraft, ja ILM on sittemmin kehittänyt ja hyödyntänyt sitä esimerkiksi Star Wars -universumiin sijoittuvien sarjojen kuten The Book Of Boba Fettin ja Obi-Wan Kenobin, sekä elokuvan Thor: Love and Thunder

(2022) tuotannoissa. (ILM i.a.) The Mandalorian -sarjaa varten luotu teknologia inspiroi tuotantoja ympäri maailmaa etsimään uusia tapoja sen hyödyntämiseen. Esimerkiksi Netflixin sarja 1899 (2022) rakensi LED-volyymiin pyöritettävän lattian, joka mahdollisti virtuaaliympäristön lisäksi myös konkreettisen lavastuksen kääntämisen rajatun tilan aiheuttamien ongelmien ratkaisemiseksi (Coldewey 2022).

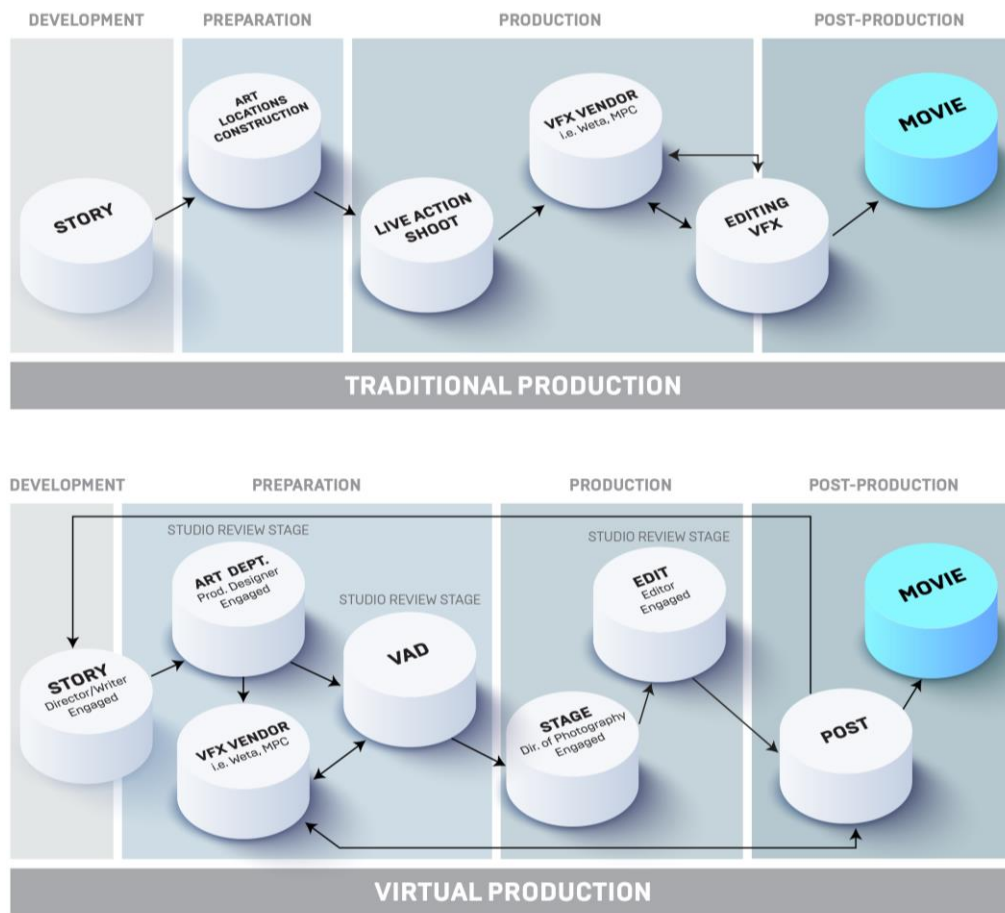
LED-paneelit ovat kuitenkin vain yksi osa teknologiaa, jonka edistyminen mahdollisti LED-volyymien käytön ICVFX -tuotannoissa. Muita LED-volyymien laatuun ja monikäyttöisyyteen vaikuttavia teknisiä tekijöitä ovat esimerkiksi prosessorit, näytönohjaimet, päivitystaajuus, värisyvyys, LED-valojen teho ja jopa tiedonsiirtoon käytetyt kaapelit. Nykyään markkinoilla on useita virtuaalituotantoon tarkoitettujen LED-paneelien valmistajia kuten esimerkiksi alalla johtavat ROE, Megapixel sekä Brompton Technology. LED-volyymeja valmistetaan useisiin käyttötarkoituksiin vaihtelevin teknisin ominaisuuksin ja studioiden valitsemat paneelit valitaan aina tapauskohtaisesta vastaamaan niiden tarpeita. Tässä opinnäytetyössä en kuitenkaan syvenny tätä enempää itse paneelien toimimiseen vaikuttavan välineistön teknologiaan.

3 3D-ympäristön matka luonnoksesta LED-volyymille

Virtuaalituotantoa varten luotavaa ympäristöä koottaessa käydään läpi samat vaiheet kuin mitä tahansa kolmiulotteista ympäristöä rakennettaessa. Jason Gregory (2014, 444) tiivistää nämä vaiheet kirjassaan Game Engine Architecture seuraavasti: kolmiulotteisten elementtien luominen ja sijoittaminen, virtuaalisen kameran asettaminen näkymän rajaamiseksi, valonlähteiden määrittäminen sekä pintojen ominaisuuksien eli materiaalien asettaminen. Näiden vaiheiden lopputuloksena on peleissä kuin myös LED-volyymillä nähtävä maailma.

Ennen varsinaista kokoamista ympäristö käy kuitenkin läpi monivaiheisen prosessin täyttääkseen kaikkien virtuaalituotantojen osapuolten vaatimukset.

Virtuaalituotannoissa sisällön tuottaminen tapahtuu esituotantovaiheessa toisin kuin perinteisissä tuotannoissa, joissa se on osa jälkituotantoa. Kuva 7 havainnollistaa tätä eroa ja korostaa samalla virtuaalituotantojen mahdollistamaa kollaboratiivisuutta eri osapuolten välillä sisällöntuotantoprosessissa. Myös Helsingissä virtuaalituotantoa tekevän Fireframe Studiosin virtuaalituotannon valvoja (VP Supervisor) Markus Helminen (2023) nostaa studion toiminnassa tärkeään rooliin suunnitteluvaiheen tiiviin yhteistyön tuotannon muiden osapuolten kanssa. Helmisen mukaan keskinäinen yhteisymmärrys käytössä olevan ohjelmiston ja teknologian käyttömahdollisuuksista auttaa optimaaliseen resurssien käyttöön sekä oikeanlaisten tavoitteiden asettamiseen jokaisessa virtuaaliympäristössä. (Helminen 2023.) Ympäristön luomisen prosessi vaihtelee tuotannoista riippuen, mutta esimerkkinä käy The Mandalorian -sarjan (2019) ensimmäinen tuotantokausi, jonka virtuaaliympäristöjen tuotantovaiheet olivat tiivistettynä seuraavat: ympäristöt hahmotteleminen konseptitaiteen, kuvakertomusten sekä kolmiulotteisten luonnosten avulla, esivisualisoinnin käyttäminen kuvasuuntien valitsemiseen virtuaaliympäristöissä, fotorealististen asettien sekä konkreettisen lavasteen rakentaminen ja virtuaaliympäristön kokoaminen ja optimoiminen. Viimeisenä konkreettinen lavastus kootaan volyymin sisälle vastaamaan kohtauksessa käytettyä ympäristöä, kuten kuvissa 4 ja 11. (Jones 2020.)



Traditional vs. Virtual Production for Film

Visual development by Fox VFX Lab

Kuva 7. Taulukko havainnollistaa perinteisten tuotantojen ja virtuaalituotantojen vaiheiden eroa. (Kadner i.a.)

Ympäristön digitaalisten elementtien luomisesta vastaa virtuaalilavasteosasto (engl. Virtual Art Department, lyhennettynä VAD). Hyvin tehdyssä virtuaaliympäristössä ohjaaja pystyy siirtämään tai pyörittämään elementtejä tai vaihtamaan päivän aikaa tai auringon suuntaa, ilman että ympäristön illuusio aitoudesta murtuu. Nämä tekijät ovat hyvä pitää mielessä seuraavia kappaleita luettaessa, joissa käyn yleisesti läpi virtuaaliympäristön luomisen vaiheet ja kokoamisen pelimoottori Unreal Engine:ssä.

3.1 Assettien luominen

Jokainen Unreal Enginen sisällä rakennettava ympäristö koostuu asseista (engl. asset). Assetit voivat koostua esimerkiksi 3D-malleista, kuvista ja tekstuureista (Epic Games 2021). Assettien luomisessa monipuolinen ohjelmisto-osaaminen auttaa löytämään toimivia ratkaisuja ympäristön rakentamiseen. Esimerkiksi The Mandalorian -sarjan virtuaalituotannon parissa työskennellyt Charmaine Chan mainitsee assettien luomiseen käytetyiksi työkaluiksi ohjelmat kuten Nuke, 3ds Max, Maya sekä Photoshop (Kadner 2021). Virtuaalituotantoja varten luotuja asetteja voidaan kutsua virtuaaliasseiksi (engl. virtual asset) ja niiden erona peliasetteihin on, että ne ovat optimoitu ICVFX -tuotantojen tarpeet, fotorealistisuus ja reaaliaikaisuus, huomioon ottaen.

3.1.1 Geometrian mallintaminen

Suurin osa virtuaalituotannossa käytettävistä ympäristöistä muodostuu kolmiulotteisesta geometriasta, joka puolestaan koostuu pisteistä ja niiden välillä muodostuvista tasoista. Näitä pisteitä kutsutaan nimellä verteksi (engl. vertex) ja tasoja nimellä polygoni (engl. polygon). Geometriaa virtuaalituotantoa varten voidaan luoda esimerkiksi mallintamalla tai 3D-skannaamalla, joista jälkimmäinen on ollut suosituin tapa esimerkiksi aiemmin mainitussa The Mandalorian -sarjassa. 3D-skannaamisella tarkoitetaan joko fotogrammetriaa tai Lidar-teknologiaa. Fotogrammetriassa kohde kuvataan kattavasti jokaisesta suunnasta, jonka jälkeen tietokoneohjelma muodostaa kuvien perusteella kohteesta kolmiulotteisen version. Lidar-teknologia toimii samankaltaisesti, mutta kuvien sijasta kohteen pintaan lähetetään lasersäteitä. Kohteen muoto rakennetaan näiden säteiden laskemien etäisyyksien avulla, jonka jälkeen se muutetaan geometriaksi. (Vü Technologies i.a.) Fotorealisimia tavoiteltaessa skannausteknologiat ovat suosittuja niiden taltioidessa elävän elämän orgaanisuuden sekä pienimmätkin yksityiskohdat sekä pintojen sävyt. Esimerkiksi The Mandalorian -sarjassa mallintamisen sijasta asetteja luotiin

rakentamalla ensin pienoismalli osasta lavastetta, joka sitten skannattiin ja tuotiin virtuaaliympäristöön (Jones 2020). Myös Kadner (2021) kehottaa fotogrammetrian käyttöön epäuskottavan lopputuloksen välttämiseksi.

Kolmiulotteisen geometrian mallinnukseen tarkoitettua ohjelmia on monia, joista Blender, Cinema 4D sekä edellä mainitut Maya ja 3ds Max ovat tunnetuimpia esimerkkejä. Unreal Engine tarjoaa mahdollisuuden käyttää ympäristöissä yleisimpiä 3D-mallien tiedostomuotoja kuten FBX sekä OBJ, mikä mahdollistaa asettien luomisen sekä tuomisen virtuaaliympäristöön lähes kaikilla 3D-mallinnukseen tarkoitetuilla ohjelmilla.

3.1.2 Materiaalit ja teksturointi

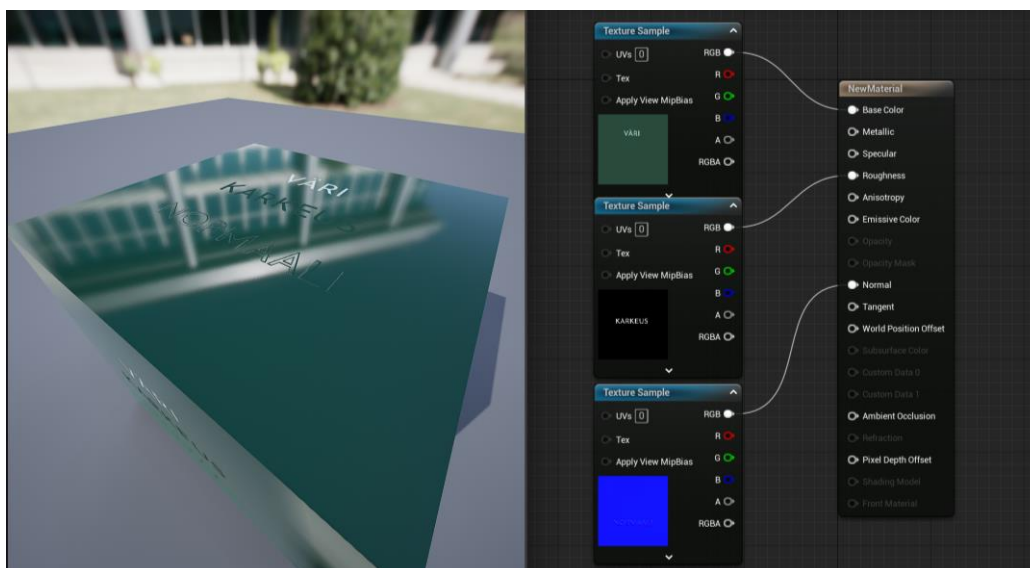
Geometrian mallintamisen luomisen jälkeen sille asetetaan materiaalit, jotka määrittelevät pintojen visuaaliset ominaisuudet. Unreal Engine käyttää monien muiden 3D-ohjelmien tavoin fysiikkaperusteista renderöintiä, josta käytetään lyhennettä PBR (physically based rendering). Fysiikkaperusteinen renderöinti tarkoittaa varjostus- ja renderöintimallia, joka kehitettiin tarjoamaan täsmällisempi representaatio valon ja pintojen vuorovaikutuksesta (McDermott 2018). Tämän ansiosta pinnat käyttäytyvät myös samalla tavalla valaistuksesta riippumatta, tarjoten materiaaleille visuaalisen johdonmukaisuuden.

Fysiikkaperusteista renderöintiä hyödyntävät ohjelmat käyttävät fysiikkaperusteisia materiaaleja. Renderöintiin ja pelimoottoreihin erikoistuva Yan Chernicov (2022) kuvaili fysiikkaperusteisten materiaalien tarjoavan standardoidut parametrit, jotka mahdollistavat realistisen, elävän elämän pintojen kuvaamisen.

Unreal Enginen materiaalit ovat optimoitu suorituskyky, käyttäjäystävällisyys sekä monipuolisuus huomioon ottaen (Karis 2013 1–2). Unreal Enginessä eri varjostusmallit (engl. shading model) määrittelevät käytettävissä olevat materiaaliparametrit, joista aseteissa käytetyissä materiaaleissa yleisimpiä ovat pohjaväri (base color), metallisuus (metallic), karkeus (roughness) sekä normaali (normal). Näiden lisäksi valittu varjostusmalli voi tarjota

mahdollisuuden asettaa materiaalille ominaisuuksia kuten esimerkiksi läpikuultavuus, läpinäkyvyys tai hajautus.

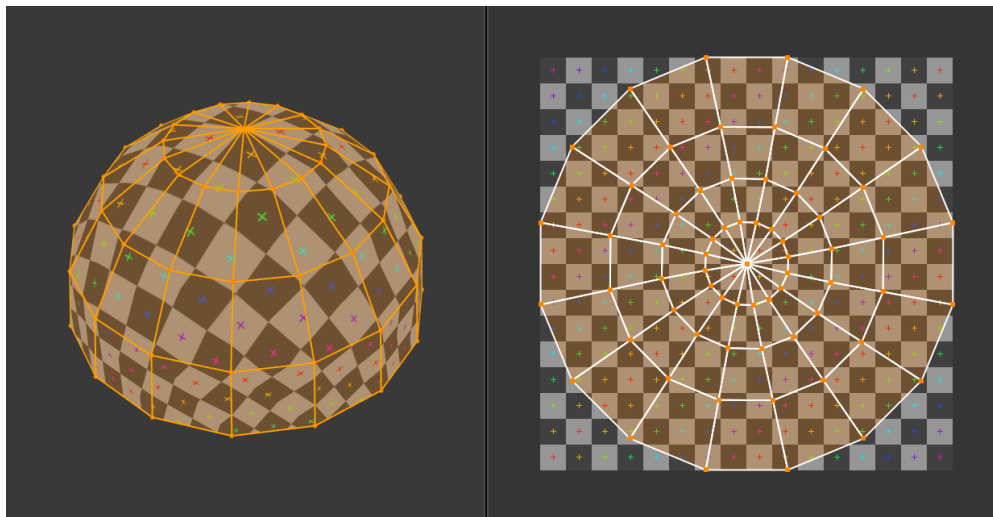
Siinä missä materiaalit määrittelevät pintojen ominaisuudet, tekstuurit määrittelevät tavan, jolla nuo ominaisuudet ilmentyvät. Tekstuurit voidaan luoda hyödyntämällä proseduraalinen teksturointia (engl. procedural texturing), tekstuurikarttoja tai näiden kahden yhdistelmää. Proseduraalinen teksturointi tarkoittaa tekstuurien luomista algoritmeja käyttäen. Tekstuurikarttoja (engl. texture map) puolestaan käytetään kertomaan materiaalille pinnan visuaaliset ominaisuudet rasterikuvan, yleensä jpg tai png, muodossa (kuva 8).



Kuva 8. Materiaalin sisältävillä tekstuurikartoilla voidaan määritellä pinnan visuaaliset ominaisuudet. Kuvassa materiaalille on asetettu pohjaväri, karkeus sekä normaali.

Osa mallintamis- ja teksturointiprosessia on myös UV kartan (engl. UV Map) luominen. UV kartta määrittelee, kuinka kaksiulotteisessa muodossa oleva tekstuurikartta ilmenee kolmiulotteisen mallin pinnalla, kuten kuvassa 9. UV kartan luomista kutsutaan nimellä *UV unwrapping* ja sen aikana määritellään, miten geometrian pinnat esitetään kaksiulotteisella tasolla. UV kartan huolellinen luominen auttaa 3D mallin teksturoimisessa ja materiaalien

toimimisessa halutulla tavalla, mutta sillä on myös suuri rooli Unreal Engine:ssä valaistuksen suhteen. Tähän palaamme seuraavassa luvussa.



Kuva 9. Vasemmalla puolella näkyy puolikkaan pallon muotoinen geometria ja oikealla sen UV kartta, joka määrittelee, kuinka ruutukuviainen teksturi esiintyy geometrian pinnassa. (Blender i.a.)

3D-mallinnusohjelmat sisältävät useimmiten omat teksturointityökalunsa.

Teksturointia varten on kuitenkin olemassa myös siihen erikseen tarkoitettuja ohjelmia kuten Adobe Substance 3D Painter sekä Mari, joista jälkimmäistä erikoistehostestudio Framestore käytti elokuvan Blade Runner 2049 (2017) futurististen kaupunkiympäristöjen realistiseen teksturointiin.

Teksturointiohjelmat tarjoavat monipuolisemman mahdollisuuden esimerkiksi valokuvien käyttämiseen ja uniikkien yksityiskohtien lisäämiseen tekstuureissa, joka on tärkeää fotorealismia tavoiteltaessa. Taiteellisena johtajana (Art Director) sekä mallintajana työskennellyt Andrew Jones kertoo, että alalla asetteja käsin mallintaessa niiden realistiset tekstuurit pohjautuvat suuresti valokuvien käyttöön (Jones i.a.).

3.2 Valonlähteet

Ympäristön valaisu ja sen muovattavuus ovat monella tapaa erittäin tärkeitä tekijöitä virtuaalituotannoissa. Visuaaliselta kannalta valaisulla on suuri rooli

fotorealistisuuden saavuttamisessa ja tuotantokannalta valaisuosuhteiden muokattavuuden säilyminen on olennaista taiteellisen vapauden säilyttämisessä. Esimerkiksi sarjan The Mandalorian elokuvaaja Greig Fraser (2021) sekä The Batman -elokuvan erikoistehostevastaava Dan Lemmon (2022) mainitsevat valaistuksen säädeltävyyden, esimerkiksi päivänajan hallinnan muodossa, olevan yksi suurista eduista LED-volyymillä työskentellessä.

Unreal Engine tarjoaa kattavan valikoiman erilaisia valonlähteitä niin luonnosta kuin myös sisätiloista löytyvän valaistuksen jäljittelemiseksi ja niitä aseteltaessa voi valita dynaamisen ja staattisen valaistuksen välillä (Epic Games i.a.). Isoimpina eroina näiden kahden välillä on muovattavuus sekä suorituskky.

Dynaamista valoa käytettäessä valonlähteet pysyvät liikutettavina ja niiden ominaisuuksia voidaan muuttaa vaikutusten päivittyessä reaaliajassa.

Dynaamisen valaistuksen Unreal Enginessä mahdollisti aikaisemmin renderöintitekniologia nimeltä rasterointi (engl. rasterization). Rasteroinnissa kolmiulotteiset elementit piirtyvät kaksiulotteiselle tasolle matemaattisten kaavojen avulla ja renderöidyn kuvan jokaisen pikselin väri määrittyy valonlähteen ja geometrian pinnan vuorovaikutuksen mukaan (Epic Games i.a.). Vaikka teknologia oli nopea, rasterointi ei kuitenkaan kyennyt imitoimaan valon vuorovaikutusta kolmiulotteisessa tilassa eri elementtien välillä. Tämä aiheutti puutteelliset heijastukset sekä epäsuoran valon vaikutukset. Ray Tracing -teknologian kehitys tarjosi ratkaisun näihin ongelmiin.

Yksinkertaistettuna Ray Tracing imitoi oikean maailman valonsäteiden käyttäytymistä, tosin käänteisellä tavalla. Teknologia toimii lähettämällä kamerasta säteen, joka kimpoilee 3D-ympäristössä geometrioiden välillä kunnes se saavuttaa valonlähteen. Pintoja kohdatessaan säde poimii ja taltioi värejä, joiden perusteella valmis kuva renderöidään (Epic Games i.a.). Toisin kuin rasteroinnissa, Ray Tracingissä valonsäteiden jatkuva vuorovaikutus eri

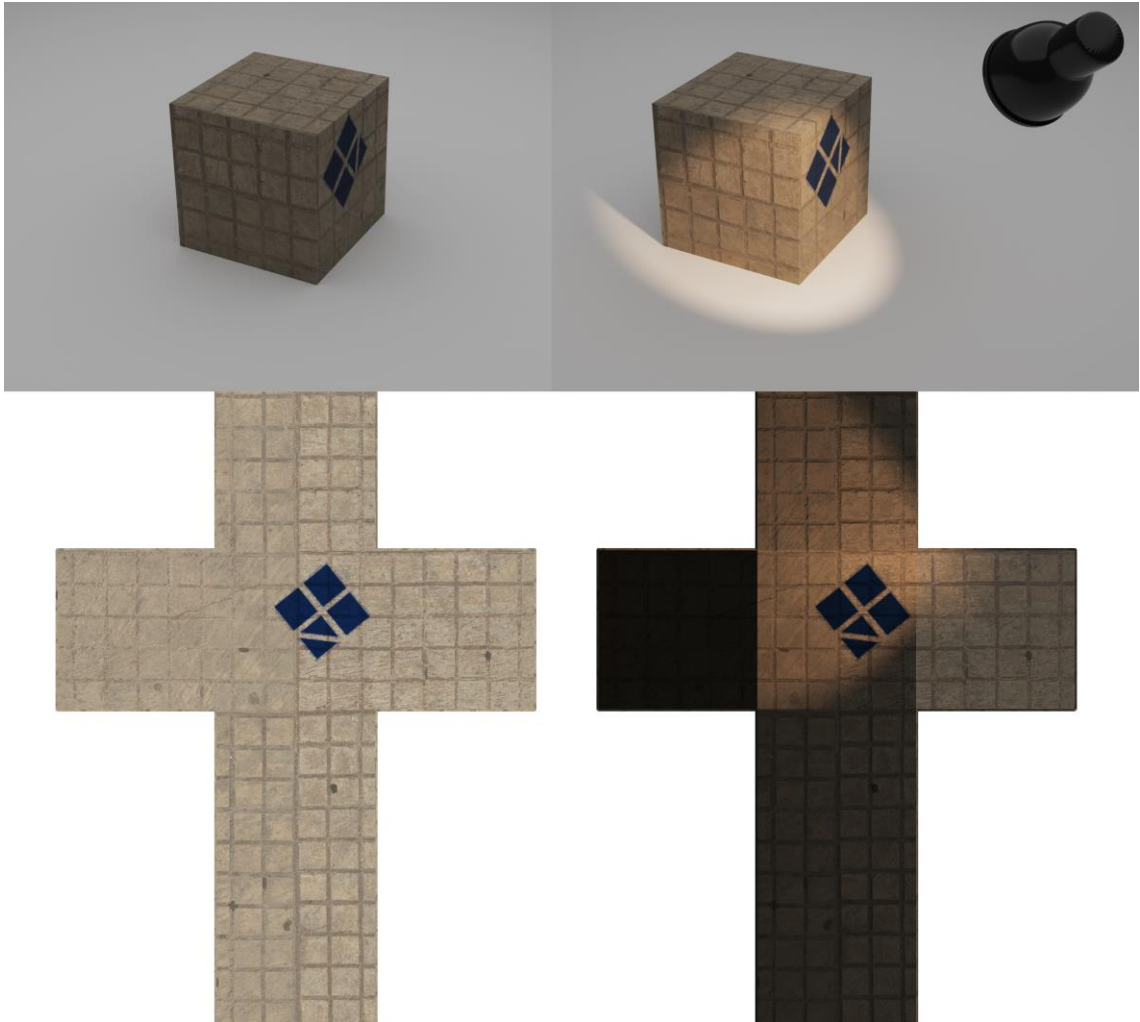
geometrioiden välillä mahdollistaa esimerkiksi realistisemmat heijastukset, varjot sekä epäsuoran valon eli GI:n (Global Illumination).

Ray tracing -teknologiaa on käytetty 3D-ohjelmissa jo pitkään, mutta sen käyttö on yleensä vaatinut pitkiä renderöintiaikoja. Vuonna 2019 Epic Games yhdisti Ray Tracing- ja rasterointitekniikkaa Unreal Enginen versiossa 4.22 mahdollistaen sen toimimisen reaaliajassa. Kyseinen teknologia on kuitenkin kuormittavaa sekä laitteistoriippuvaista vaatien siihen soveltuvan grafiikkaprosessorin. Tätä laitteistoa hyödyntävää Ray Tracing -teknologiaa kutsutaan nimellä Hardware Ray Tracing. Vaikka Hardware Ray Tracing tarjoaa Unreal Enginen vaihtoehtoisia realistisimman valaistuksen, on sen dynaaminen käyttö usein liian kuormittavaa reaaliaikaisiin käyttötarkoituksiin.

Lumen Global Illumination and Reflections on Epic Gamesin ratkaisu realistisen dynaamisen valaistuksen luomiseen tarjoten laitteiston sijasta ohjelmistopohjalta toimivan, eri Ray Tracing -teknologioita yhdistelevän valaisujärjestelmän (Epic Games i.a.). Lumen on optimoitu tavalla, joka mahdollistaa aidomman epäsuoran valon sekä heijastukset myös uusilla pelikonsoleilla ja tietokoneilla, jotka eivät kykene käyttämään Hardware Ray Tracing -teknologiaa.

Staatinen valaistus vaatii valaisuolosuhteiden tallentamista ja se tapahtuu valon beikkaamisella (engl. baking). Valon beikkaaminen tarkoittaa valojen ja varjojen vaikutusten kiinnitymistä geometrian tekstuureihin (kuva 10). Valojen beikkaaminen Unreal Enginissä onnistuu CPU Lightmass tai GPU Lightmass -työkalun avulla. Jälkimmäinen mainituista vaatii toimiakseen Hardware Ray Tracing -teknologiaa, mutta se mahdollistaa nopeamman valaistuksen laskemisen ja muutoksien tekemisen helpommin beikkaamisen aikana. Molemmat työkalut laskevat valojen vuorovaikutuksen ympäristössä ja tallentavat vaikutukset geometrioiden *UV valokarttoihin* (engl. Lightmap UV). UV valokartat voidaan luoda itse mallinnus- tai teksturointiohjelmien työkaluilla, mutta Unreal Engine luo ne myös automaattisesti geometrian UV kartan

pohjalta. Tästä syystä mallinnus- ja teksturointivaihe ovat tärkeitä, sillä beikatun valon laatu ja toimivuus perustuu suurelta osin toimiviin UV karttoihin.



Kuva 10. Vasemmalla esine ja sen pohjaväritekstuuri ilman valon beikkaamista, oikealla beikattuna

Lukuisista mahdollisuuksista huolimatta LED-volyymillä työskentely asettaa myös rajoituksia valaisuun. Virtuaaliympäristöön ei kannata muun muassa asettaa valoja, joiden vaikutusta ei pysty jäljentämään konkreettisen lavasteen valaistuksella. Greig Fraser ohjeistaa Epic Gamesin julkaisemassa haastattelussaan myös välttämään suoran auringonvalon käyttämistä ympäristöissä, sillä sen jäljentäminen studio-olosuhteissa on ongelmallista (Kadner 2021). Unreal Enginen ICVFX ohjeistuksessa ratkaisuksi esitetään

esimerkiksi virtuaalilavastuksen asettaminen varjoisaan tilaan ja yleisestikin valaistusta asetettaessa kannattaa suosia pehmeää valoa. Tämän lisäksi realistiset peiliheijastukset ovat haaste, johon Unreal Engine ei vielä tarjoa ihanteellista ratkaisua.

Heijastukset voidaan pitää dynaamisina, jolloin pelimoottori laskee ne joka ruutua varten tai niitä voidaan imitoida erilaisilla Reflection Capture -työkaluilla. Lumenin tarjoamia heijastuksia on kehitetty Unreal Enginen viimeisimmissä päivityksissä, mutta niiden laatu ei vieläkään tarjoa realistisia tuloksia. Hardware Ray Tracing mahdollistaa realistisemmat heijastukset, joissa voi kuitenkin myös ilmentyä visuaalisia häiriöitä täysin kiiltävien pintojen, kuten ikkunalasien, peilien tai kromin kanssa.

Reflection Capture -työkalut toimivat taltioimalla halutusta sijainnista kuvan ympäristöstä, jonka se sitten projisoi kiiltäviin pintoihin. Työkalu ei kuormita ympäristöä, joten se on suorituskyvyn kannalta edullinen vaihtoehto. Reflection Captures -työkalujen avulla saatavat heijastukset eivät kuitenkaan käyttäydy todenmukaisesti niiden ollessa vain imitaatio heijastuksista, joten realistinen lopputulos voi olla haastavaa saavuttaa varsinkin isompien kiiltävien pintojen kanssa.

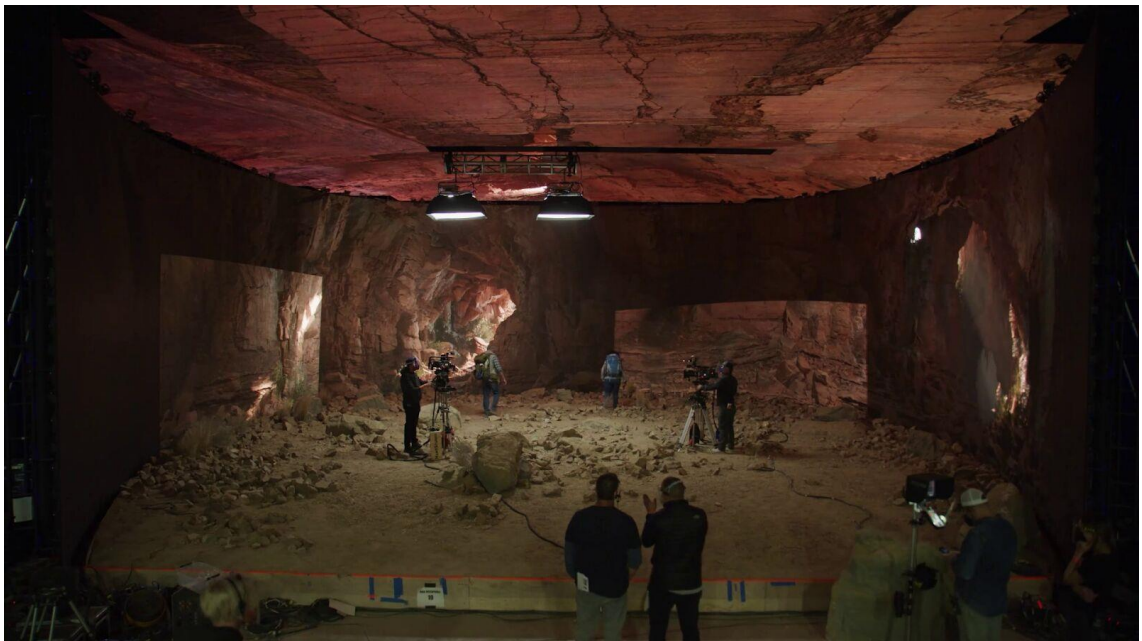
Virtuaalituotannoissa valaistuksessa tavoitellaan fotorealistsuutta, joka on kuitenkin jatkettavissa studio-olosuhteissa luotavalla valaistuksella. Valaistuksen tulee palvella tuotannon tarpeita, joten valinta edellä mainittujen työkalujen välillä on lähes aina tuotantokohtainen.

3.3 Volyymi ja lavastus

Rakennettu virtuaaliympäristö heijastetaan kuvauspäivänä LED-volyymille, jonka edustalle konkreettinen lavastus on rakennettu (kuva 11). Volyymillä nähtävä ympäristö jakautuu sisäiseen sekä ulkoiseen *frustumiin* (engl. Frustum). Ulkoinen frustum näyttää staattista ympäristöä valitusta sijainnista sekä kulmasta. Sisäinen frustum puolestaan näyttää oikean kameran liikkeitä

seuraavan virtuaalikameran näkymän. Kuvassa 11 sisäiset frustumit näkyvät kahtena pienempänä laatikkona eri puolilla volyymiä.

Virtuaalinen versio oikeasta volyymistä sijoitetaan ympäristöön aiemmin mainitun nDisplay-lisäosan avulla. Virtuaalisen volyymin sijainti suhteessa oikean kameras kanssa kalibroituun virtuaalikameraan määrittelee, mikä osa ympäristöstä näkyy oikealla volyymilla. NDisplay liikuttaa koko virtuaalivolyymin sekä virtuaalikameran muodostamaa kokonaisuutta ja mahdollistaa näin halutun näkymän valitsemisen. Halutun sijainnin ja suunnan löytyessä tulee kuitenkin varmistaa, ettei ympäristössä ole asetteja studion lavasteen kokoa vastaavalla alueella, niiden rikkoessa illuusion yhtenäisyydestä.



Kuva 11. Unreal Enginen ICVFX testissä LED-volyymin kanssa käytettiin kahta kameraa sekä lavasteet rakennettiin yhdenmukaisiksi virtuaaliassettien kanssa. (Epic Games i.a.)

Aiemmissa kappaleissa mainittujen vaiheiden tavoitteena on rakentaa virtuaaliympäristö, joka on integroitavissa konkreettiseen lavasteeseen. Toimiva integroiminen vaatii virtuaaliympäristöltä kuitenkin muokattavuutta. Epic Gamesin Juan Gomez (2021) mainitsee konkreettisen lavasteen ja virtuaaliympäristön integroimistavoiksi esimerkiksi ympäristön visuaalisten

ominaisuuksien, kuten kirkkauden ja sävyn, täsmäämisen sekä konkreettisten lavasteiden toistumisen virtuaaliympäristössä. Jälkimmäinen onnistuu luvussa 3 selitetyillä tekniikoilla, etenkin 3D skannausmenetelmillä. Mahdollisuus visuaalisten ominaisuuksien muokkaaminen voidaan ulottaa esimerkiksi yksittäisiin materiaaleihin, ympäristön alueisiin, tai koko ympäristöön. Isommissa tuotannoissa monien työskennellessä samojen ympäristöjen parissa, tulee muokkaamismahdollisuudet tehdä helposti löydettäviksi sekä hallittaviksi myös rakennusprosessin ulkopuolisille. Tämä onnistuu muun muassa instanssoiduilla materiaaleilla (engl. instanced materials), keräämällä lavasteessa toistuvat assetit omiin kokoelmiinsa ja asettamalla ympäristölle PostProcess Volume. Instanssoitu materiaali tarkoittaa materiaalista luotua kopiota, jossa halutut ominaisuudet ovat muokattavissa parametrien avulla. Tämän avulla materiaaliominaisuuksiin kuten väri tai karkeus voidaan päästä reaaliajassa käsiksi ilman varsinaisen materiaalirakenteen avaamista ja muokkaamista. PostProcess Volume on ympäristöön asetettava volyyymi, joka tarjoaa mahdollisuuden ympäristön visuaalisten ominaisuuksien, kuten värien, sävyjen sekä valojen, kokonaisvaltaiseen säätelyyn (Epic Games i.a.).

Integroiminen tapahtuu myös konkreettista lavastetta muuttamalla, esimerkiksi savun tai studiovalaistuksen yhteensovittamisella virtuaaliympäristön kanssa. Valojen säätö voidaan hoitaa myös pelimoottorin sisällä, sillä Unreal Enginen sisältämä Digital Multiplex (DMX) lisäosa mahdollistaa virtuaaliympäristön sekä lavasteen valaistuksen yhteensovittamisen sekä säätämisen reaaliajassa. DMX tarkoittaa vakiintunutta tiedonsiirtoprotokollaa, jota käytetään valaistuksen hallitsemiseen.

Yhteenvedona integroimisen toimiminen varmistetaan virtuaaliympäristön muovattavuudella. Kustomoitujen asetusten helppokäyttöisyys kannattaa varmistaa jo ympäristön rakentamisen aikaisissa vaiheissa. Muuttujat, kuten kameran käyttämä väriavaruus, lavasteiden todelliset sävyt tai LED-volyymin korkeampi kirkkaus verrattuna monitoriin, tekevät lopputuloksen ennakoimisesta vaikeaa (Gomez 2021).

4 Optimointi

Optimointiin perehtyessäni kohtasin saman neuvon suuressa osassa materiaalia: optimoi vain vaadittaessa. Virtuaaliympäristö ei edellytä optimointia, jos se täyttää virtuaalituotannon vaatimukset, sillä optimointi lähes aina tarkoittaa tinkimistä visuaalisesta laadusta. Visuaalisten efektien tuottaja (visual effects producer) Ryan Stafford (i.a.) kertoo prosessointitehon olevan yksi suurimmista virtuaalituotantojen haasteista, joka selätetään tekemällä kompromissi visuaalisen näytävyyden sekä suorituskyvyn välillä. Samasta syystä liian aikainen optimointi voi estää korkeimman mahdollisen laadun saavuttamisen, minkä takia Arnbjörnsson (2021) ei suosittele ennen aikaista optimointia, vaan kehottaa olemaan suorituskykytietoinen työskentelyprosessin aikana. Myös Helmisen (2023) mukaan optimointi on läsnä koko tuotannon ajan ja se toteutetaan aina tuotannon vaatimukset huomioon ottaen.

Kolmiulotteisen ympäristön vaatimukset sekä optimointikeinot ovat samankaltaiset peleissä ja virtuaalituotannoissa, onhan pelimoottori tehty alun perin pelien luomiseen. Isoin ero näiden välillä on, että virtuaalituotannoissa ympäristöissä tavoitellaan fotorealismia korkeimmalla mahdollisella laadulla, kun taas pelien ympäristöissä voidaan tinkiä visuaalisista yksityiskohdista tarinankerronnan ja pelattavuuden sujuvuuden varmistamiseksi. Tärkeimpiä tavoitteita ympäristöjen optimoimisessa virtuaalituotantoja varten onkin ympäristön laadun ja yksityiskohtien säilyminen tavalla, joka mahdollistaa ympäristön toiston vaaditulla ruudunpäivitysnopeudella. Ruudunpäivitysnopeus eli FPS (engl. frames per second) kertoo, kuinka monta kertaa kuva päivittyy näytöllä sekunnin aikana ja ruudunpäivitysnopeuden minimivaatimus LED-volyymillä on 24 ruutua sekunnissa. Tämä johtuu televisio- ja elokuvatuotantojen vakiintuneesta ruudunpäivitysnopeudesta. Vaadittu ruudunpäivitysnopeus voi kuitenkin vaihdella tuotannon mukaan.

Luodun ympäristön ja projektitiedoston asetuksia voi joutua muuttamaan tapauskohtaisesti LED-näyttöjen ja niihin liittyvän tekniikan vaihdellessa. Unreal Enginellä rakennetuille projekteille voi kuitenkin asettaa tavoitteita omalla koneella työskennellessä, jotka auttavat varmistamaan sen pyörimisen sulavasti LED-volyymin kanssa. Unreal Enginen ohjeistuksessa korkearesoluutioista monitoria käytettäessä ympäristölle kehoitetaan asettamaan vaatimukseksi 2–3 kertaa tavoiteltua korkeampi ruudunpäivitysnopeus. Tällöin virtuaalituotannossa käytettävien työkalujen kuten nDisplayn kuormittavuus on otettu huomioon. (Epic Games i.a.)

Vaikka ruudunpäivitysnopeus on kokonaisuutta tarkasteltaessa tärkeä indikaattori ympäristön suorituskyvylisistä vaatimuksista, optimoinnin vaikutuksia tutkitaan ruudun renderöimiseen kulumien millisekuntien avulla. Ruudunpäivitysnopeudessa tapahtuvat muutokset ovat eksponentiaalisia, eivätkä ne aina anna selkeää kuvaa optimoinnin vaikutuksista, toisin kuin millisekunnit, joiden muutokset ovat lineaarisia (Epic Games 2022).

Arnbjörnsson (2022) tiivistää optimoinnin vaiheet kysymyksiin mitä, miksi ja miten, eli mikä hidastaa peliä, miksi se hidastaa peliä ja miten se korjataan. Tässä luvussa käydään läpi keinot noihin kysymyksiin vastaamiseen virtuaalituotannon kontekstissa, sekä muitakin ympäristön rakentamisessa huomioon otettavia asioita.

4.1 Alkuvaiheen optimointi ja profilointi

Vaihetta, jossa selvitetään vastaus kysymykseen ”mikä hidastaa peliä”, tai tässä tapauksessa virtuaaliympäristöä, kutsutaan profiloinniksi. Profilointi tarkoittaa pelimoottorin suorittamien toimintojen ja niiden viemän ajan tarkastelua, mihin Unreal Engine tarjoaa useita apuvälineitä.

Reaaliaikaiseen suorituskyvyn visualisointiin voidaan käyttää erilaisia stat-komentoja (engl. stat commands). Stat-komennot ovat erityisesti pelistatistiikan näyttämiseen tarkoitettuja konsolikomentoja (Epic Games i.a.), joita käyttämällä

saadaan näkyville muun muassa ruudunpäivitysnopeus, ruudun luomisen vievät millisekunnit ja piirtokutsut. Kuvassa 12 on nähtävissä kuinka profilointiin usein käytetty Stat GPU-komento esittää laajemmin suorituskäytävien vaikuttavien tekijöiden osuutta ruudunpäivitysajasta.

GPU [STATGROUP_gpu]	Average	Max	Min
Counters			
[TOTAL]	38.60	46.81	37.71
Shadow Projection	8.74	9.04	8.66
Temporal Super Resolution	5.46	5.72	5.44
LumenScreenProbeGather	5.17	5.49	5.05
LumenSceneLighting	4.92	6.20	4.41
Shadow Depths	2.80	3.27	2.46
Slate UI	1.16	1.23	1.12
Lights	1.86	2.13	1.81
Nanite VisBuffer	1.31	1.40	1.12
Nanite BasePass	1.21	1.37	1.17
LumenReflections	1.12	1.21	1.09
RenderDeferredLighting	1.15	1.17	1.14
[unaccounted]	0.83	6.13	0.68
Postprocessing	0.43	0.47	0.39
VolumetricCloudShadow	0.42	0.43	0.41
Nanite Roadback	0.30	0.44	0.26
Basepass	0.43	3.91	0.33
SkyAtmosphereLUTs	0.25	0.25	0.24
Translucent Lighting	0.24	0.25	0.23
LumenSceneUpdate	0.21	0.39	0.14
Editor Primitives			
VolumetricCloud	0.05	0.07	0.05
BeginOcclusionTests	0.06	0.06	0.05
Composition BeforeBasePass	0.06	0.08	0.06
HZB	0.08	0.10	0.07

[20 more stats. Use the stats MaxPerGroup CVar to increase the limit]

Kuva 12. Stat GPU -komento näyttää ruudun renderöimiseen kuluvien millisekuntien lisäksi miten suuren osuuden eri toiminnot vievät tuosta ajasta.

Piirtokutsut (engl. draw calls) tarkoittavat pelimoottorin tapahtumia, joissa assetti haetaan renderöitäväksi. Piirtokutsut tapahtuvat joka renderöidyssä ruudussa, ja niiden suuri määrä on Unreal Enginen dokumentaation mukaan pelin graafiseen suorituskäytävien eniten vaikuttava tekijä silloin kun Nanite-teknologiaa ei käytetä. Piirtokutsujen summa koostuu jokaisesta ainutlaatuisesta geometriasta sekä niille asetetuista ainutlaatuisista materiaaleista.

Tarkempaan tarkasteluun Unreal Engine tarjoaa GPU Visualizer -työkalun. Se tarjoaa mahdollisuuden tarkastella ruudun renderöinnin aikana tapahtuvaa prosessia eriteltynä eri vaiheisiin, joka mahdollistaa jokaisen laskelmointivaiheen ja niiden ajallisen vaikutuksen tutkimisen erikseen. Tämä antaa tekijän nähdä esimerkiksi yksittäisten valojen vaikutusten suorituskäytävien.

Kattavin Unreal Enginen profilointityökalu on nimeltään Unreal Insights, joka avataan omana sovelluksenaan ennen varsinaisen pelimoottorin

käynnistämistä. Unreal Insights kerää, analysoi ja visualisoi pelimoottorin dataa halutun ajan (Epic Games i.a.), mikä auttaa tunnistamaan yksittäisiä ongelmakohtia virtuaaliympäristössä liikuessa.

Visualisointityökaluja on Unreal Enginessä runsaasti ja ovat tarkoitettu ympäristön ja siinä käytettyjen työkalujen toimivuuden visualisointiin. Visualisointityökaluilla voidaan tulkita esimerkiksi, kuinka hyvin Nanite- tai Lumen-teknologiat toimivat ympäristössä, mitä LODeja aseteista käytetään miltäkin etäisyydeltä tai onko tekstuurikarttojen resoluutio sopiva.

Unreal Enginen projektiasetuksiin tutustuminen on tärkeää niiden määrittäessä virtuaaliympäristön hyödyntämät työkalut ja teknologiat. Unreal Engine tarjoaa valmiita projektipohjia eri tarkoituksiin tehtäville ympäristöille, niin kuin myös virtuaalituotannolle. Tästä huolimatta Epic Games suosittelee asettamaan jokaisen uuden projektin asetukset aina kyseessä olevan tuotannon tarpeisiin sopivaksi. Epic Games kehottaa myös turhien asetusten tai lisäosien pois päältä asettamiseen, niiden alentaessa suorituskykyä taustalla pyöriessään. Näistä esimerkkinä toimii VR-pelien tukemiseen tarkoitettut lisäosat, jotka ovat oletuksena käytössä, vaikka niitä ei yleensä virtuaalituotannoissa tarvita. (Epic Games i.a.)

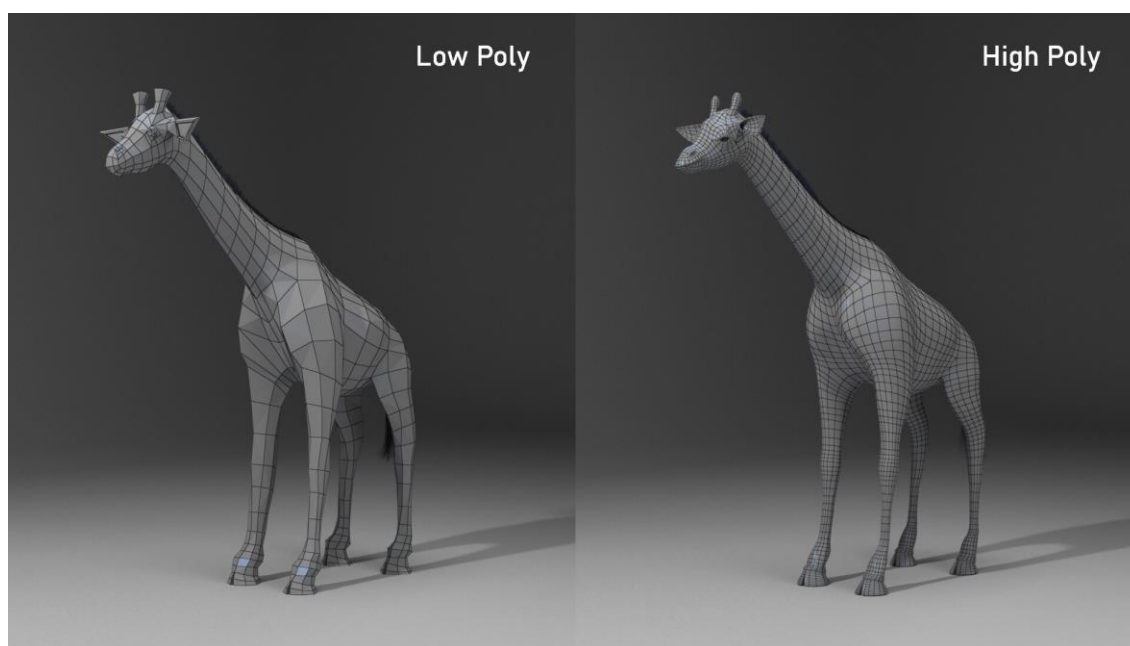
Asetusten muokattavuudesta voidaan tehdä myös helpompaa niille tarkoitetuilla työkaluilla. Isoissa virtuaaliympäristöissä esimerkiksi tiettyjä renderöintiasetuksia saatetaan haluta käyttää vain ennalta määritellyissä sijainneissa. Projektin asetusten nopeaa muuttamista varten siihen voidaan asettaa kustomoituja *konsolimuuttujia* (engl. Console Variable, lyhennettynä CVar). Näiden arvoa vaihtamalla voidaan vaikuttaa haluttuun projektin asetukseen, kuten valaistuksen laatuun, tekstuurien latautumiseen tai LODien käyttöön. CVarit antavat käyttäjälle mahdollisuuden optimoida ympäristöä nopeasti ja joustavasti tuotannonkin aikana. Yksittäisten asetusten nopeaa muuttamista varten voidaan käyttää myös komentoriville kirjoitettavia konsolikomentoja.

Edellä mainittujen työkalujen hyödyntäminen ja säännöllinen tarkkaileminen mahdollistaa isompien suorituskyvylisten ongelmakohtien tunnistamisen välittömästi (Oztalay 2020). Optimointiin voi keskittyä myös työskentelyn loppuvaiheessa, mutta suurten muutoksien tekeminen jälkeinpäin voi viedä olla runsaasti aikaa siihen verrattuna, että työskentelytapoja mukauttaa ympäristön rakentamisen varrella löytyneiden ongelmien ratkaisemiseksi.

4.2 Assettien optimointi

Kuten kappaleesta 3.1 ilmenee, yksityiskohtaisten mallien luominen sekä teksturointi virtuaalituotantoa varten on aikaa vievä prosessi ja usein assetteja hankitaan valmiina tai niiden luominen ulkoistetaan niihin erikoistuville yrityksille. Epic Gamesin Virtual Production Field Guidessa virtuaalituotantoon opitmoitujen assetin hankkimiseen mainitaan alustoiksi esimerkiksi Epic Games Marketplace, Quixel Megascan sekä Sketchfab (Kadner i.a.). Valmiiden assettien hankinta voi olla ajan säästämisen lisäksi tapa varmistaa niiden sujuva toiminta virtuaalituotantoympäristössä.

Polygonimäärien minimoiminen oli pelien optimoinnissa pitkään yksi olennaisimmista toimista suorituskyvyn parantamiseksi, sillä mitä enemmän polygoneja assetissa on, sitä raskaampaa se on pelimoottorille renderöidä. Peliassetien mallinnusprosessissa suositellaan yhä työskentelytapaa, jossa samasta geometriasta tehdään yksityiskohtaisempi, korkeapolygoninen eli niin sanottu high poly -versio sekä yksinkertaisempi, vähemmän polygoneja sisältävä low poly -versio. Kuva 13 havainnollistaa, kuinka korkeampi polygonimäärä mahdollistaa 3D-mallissa enemmän yksityiskohtia sekä luonnollisemman muodon. Mallinnuksen high poly -versiota avuksi käyttäen yksityiskohdat pystytään siirtämään low poly-versiolle tekstuuri- ja normaalikarttojen avulla. Tämä auttaa antamaan yksinkertaisemmalle ja kevyelle geometrialle illuusion yksityiskohtaisuudesta. (Hridja 2022.)



Kuva 13. Vasemmalla yksinkertaisempi 3D-mallin low poly -versio ja oikealla enemmän yksityiskohtia sisältä high poly -versio. (Eggen i.a.)

Yksityiskohtaisten, fotorealististen asettien mahdollistamat teknologiat kuten fotogrammetria sekä LiDAR tuottavat kuitenkin erittäin tiheää geometriaa.

Orgaanisia muotoja mallintaessakin realisuus vaatii korkeita polygonimääriä realistisuuden saavuttamiseksi. Epic Gamesin 2022 julkaiseman Unreal Engine version 5 sisältämä teknologia, jota kutsutaan nimellä Nanite, esitteli uuden tavan käsitellä tällaisia valtavia polygonimääriä sisältäviä geometrioita.

Yksinkertaistettuna Nanite-teknologia jakaa tiheää geometriaa ryhmiin kameran näkymän pohjalta hyödyllistäen pintojen suuntaa, syvyyttä sekä niiden viemää pikselimäärää näytöltä tavalla, joka mahdollistaa yksityiskohtien sekä suorituskyvyn säilymisen. Naniten tapa renderöidä ja jättää renderöimättä ympäristöä kameran näkymän mukaan poistaa myös aiemmin mainittujen piirtokutsujen tarpeen, mikä tekee siitä erittäin nopean. (Epic Games i.a.)

Monissa ohjeistuksissa, kuten Epic Gamesin vuoden 2021 optimointiohjeessa, kehoitetaan Naniten hyödyntämiseen. Sen käyttö on kuitenkin suorituskykyä verottavaa ja vaatii tehokkaita näytönohjaimia sekä grafiikkakortteja, eikä se niidenkään kanssa ole aina optimaalisin tapa käsitellä geometriaa. Unreal

Engine 5:n dokumentaatioissa mainitaan esimerkiksi rikkinäinen geometria sekä runsas päällekkäisyys tapauksiksi, joiden kanssa Nanite ei kykene hyödyntämään teknologiaa, johon sen optimointi perustuu. Rikkinäisellä geometrialla tarkoitetaan monia pieniä, erillään olevia geometrioita, jotka etäisyydessä muodostavat volyymin, kuten esimerkiksi lehdet, ruohikko tai hiukset. Runsa päällekkäisyys tarkoittaa taas lomittain tai päällekkäin olevia geometrioita, joiden pinnat ovat hyvin lähellä toisiaan. Naniten suorituskyvyn kuormittavuus kasvaa myös suhteessa näytön resoluutioon, mikä vaikuttaa sen käyttöön virtuaalituotannossa. Eri profilointi- ja visualisointityökalujen lisäksi Unreal Engine tarjoaa Nanite auditing tool -työkalun sen tulkitsemiseen, tulisiko geometrian käyttää Nanite-teknologiaa vai ei.

Ilman Nanite-teknologian käyttöä korkeapolygonisen geometrian tuoma kuormittavuus voidaan ratkaista teknologialla, jota kutsutaan nimellä Level Of Detail (LOD). Se tarkoittaa pitkään käytössä ollutta tekniikkaa, jossa samasta 3D-mallista on olemassa eri yksityiskohtaisuuden tasoja eli LODEja, joita pelimoottori näyttää riippuen elementin koosta näytöllä. Lähellä kameraa, suuremman osan näytöstä viedessään pelimoottori renderöi laadukkaimman ja yksityiskohtaisimman version assetista, kun taas sen viedessä pienen määrän tilaa se muuttuu yksinkertaisemmaksi. Unreal Engine tarjoaa mahdollisuudet luoda LODit automaattisesti tai manuaalisesti. Manuaalinen LODien luominen vaatii assetista itse mallinnettua, alhaisemman polygonimäärän omaavaa versiota tai versioita. Eri laatutasoisia geometrioita käytettäessä virtuaalituotannoissa tulee kuitenkin huomioida niiden vaihtumiseen vaikuttavat asetukset, sillä assetin laadun vaihtuminen ympäristöä kuvatessa on helposti huomattavissa. Ongelman pystyy kuitenkin välttämään määrittelemällä itse, kuinka paljon tilaa assetti voi viedä näytöltä ennen kuin LOD vaihtuu.

Mitä enemmän asetteja sekä materiaaleja ympäristössä käytetään, sitä enemmän ne vaikuttavat suorituskyykyyn, varsinkin kun ympäristö ei hyödynnä Nanite-teknologiaa. Unreal Enginessä on monia tapoja vähentää monien erilaisten asettien tarvetta. Esimerkiksi ympäristössä ohjeistetaan suosimaan

instanssoituja geometrioita (engl. instanced static mesh) (Epic Games 2022). Tämä tarkoittaa assetin monistamista tavalla, jolla pelimoottori tulkitsee sen yhdeksi kokonaisuudeksi monen erillisen assetin sijasta. Unreal Engine tarjoaa assettien yhdistämiseen eri vaihtoehtoja, joiden erot koskevat syntyvän assetin muokattavuutta, laatua ja materiaaleja.

Edellä mainituista teknologioista huolimatta polygonimäärä on hyvä pitää niin alhaisena kuin mahdollista, yksityiskohdista kuitenkin tinkimättä. Alhaisempi polygonimäärä tekee geometriasta kevyempää säilyttää sekä siirrellä ja sitä on myös usein helpompaa muokata jälkeenpäin tarpeen vaatiessa. Vaikka Nanite keventää korkeapolygonisen geometrian tuomaa taakkaa, voidaan tuotannoissa kohdata tilanteita, joissa kyseistä teknologiaa ei käytetä. Tällaisissa tilanteissa järkevästi mallinnettu, kevyempi geometria on tärkeä tekijä hyvän suorituskyvyn säilyttämisessä.

4.3 Materiaalien optimointi

Unreal Enginen proseduraalinen teksturointi tarjoaa mahdollisuuden tekstuurikarttojen monipuolisempaan käyttöön, joka auttaa vähentämään tekstuurikarttojen vaativaa tallennustilaa sekä kuormittavuutta. Epic Games suosittelee geneeristen pintojen käyttöön materiaaleja, joiden sisältämiä tekstuureita pystytään saumattomasti monistamaan laajempien pintojen peittämiseksi. (Epic Games i.a.)

Materiaalien suhteen Epic Games suosittelee käyttämään vain yhtä materiaalia assettia kohden piirtokutsujen ja vaaditun laskemisen vähentämiseksi. Materiaaleissa kannattaa myös hyödyntää proseduraalista teksturointia esimerkiksi tapauksissa, joissa usein toistuviin tekstuurikarttoihin saadaan visuaalista vaihtelua. (Epic Games i.a.) Tämä vähentää monien uniikkien tekstuurikarttojen tarvetta, joka säästää muistia ja parantaa suorituskykyä. Hyvin monimutkaista proseduraalista teksturointia käyttävät materiaalit voivat kuitenkin vaatia runsaasti laskentatehoa niiden rakentuessa algoritmeista, jotka

ohjelma joutuu laskemaan jokaista ruutua varten. Tällaisissa tapauksissa Arnbjörnsson (2022) suosittelee tekstuurikarttojen luomista materiaalin kuormittavuuden vähentämiseksi.

Unreal Engine luo automaattisesti tuoduista tekstuureista eri resoluutioisia versioita, joita se näyttää etäisyyden mukaan. Tätä kutsutaan nimellä Mip-map generation, ja se on mahdollista vain, jos tekstuurit ovat resoluutioltaan kahden potenssi, kuten yleiset tekstuuriresoluutiot 1024x1024, 2048x2048 ja 4096x4096. Mip-pohjaisen teksturoinnin haittana on, että vaikka ympäristössä assetista näkyisi vain pieni osa, pelimoottori lataa kaikki sen käyttämät tekstuurit. (Epic Games i.a.) Unreal Engine -projektiin pystytään tuomaan tekstuurikarttoja, joiden resoluutio on jopa 8192 kertaa 8192 pikseliä, mutta niin suuria tekstuureita kuitenkin harvoin tarvitaan ja niitä tulisi välttää muistin säästämiseksi (Epic Games 2022). Jokaisen tekstuurin suhteen tulisi arvioida tavoiteltuun laatuun riittävä resoluutio, mitä varten Unreal Engine tarjoaa oman visualisointityökalunsa. Unreal Engine tarjoaa myös mahdollisuuden asettaa yksittäisille tekstuureille enimmäisresoluutioita ympäristön keventämiseksi ja varsinkin muistin säästämiseksi. Tätä toimintoa voi käyttää vähentämään tekstuurien tuomaa taakkaa esimerkiksi suurten tekstuurien kanssa, joiden tiedetään jäävän taka-alalle.

Materiaalien sekä tekstuurien kanssa olennaista on löytää tasapaino niiden tuoman kuormittavuuden ja hyödyn välillä. Oikein käytettynä materiaalit ovat hyödyllinen keino lisätä realistisuutta sekä yksityiskohtia virtuaaliympäristöön, mikä mahdollistaa kevyemmän geometrian käytön. Monimutkaisten materiaalien sekä korkearesoluutioisten tekstuurien suuri määrä voi kuitenkin lisätä ympäristön kuormittavuutta ja heikentää suorituskyykyä.

4.4 Valaistuksen optimointi

Kuten aiemmin mainittiin, virtuaaliympäristön valaistuksen voi asettaa dynaamiseksi tai staattiseksi. Valaistuksen ollessa yksi suurimmista

kuormittavuuteen vaikuttavista tekijöistä, staattisen valon käyttäminen tarjoaa useimmissa tapauksissa parhaan suorituskyvyn. Esimerkiksi Epic Gamesin julkaisema ICVFX-ohjeisto sekä ympäristöartisti (Environment artist) Pedro Mameluque (2022) kehottavat suosimaan staattista valoa suorituskyvyn parantamiseksi. Staattisen valon beikkaaminen mahdollistaa valaistuksen, joka ei laskemisen jälkeen ei verota suorituskykyä reaaliajassa toistettuna. GPU Lightmass -työkalun ansiosta Hardware Ray Tracing -teknologiaa käyttävä ympäristö saadaan toistumaan korkealla ruudunpäivitysnopeudella. GPU Lightmass ei kuitenkaan kykene luomaan kiiltäviä pintoja tai heijastuksia, joiden kanssa tulee turvautua esimerkiksi aiemmin mainittuihin Reflection Capture -työkaluihin tai kuormittavaan Hardware Ray Tracing -teknologiaan.

Ympäristön vaatiessa dynaamista valaistusta Lumen Global Illumination and Reflections tarjoaa reaaliajassa muutettavan realistisen valaistuksen. Lumen on erittäin optimoitu valaisujärjestelmä, joka hyödyntää erilaisia ray tracing teknologioita käyttäen raskaampia menetelmiä vain tarvittaessa. Lumenin voi halutessaan myös asettaa hyödyntämään Hardware Ray Tracing -teknologiaa paremman lopputuloksen saamiseksi suorituskyvyn kustannuksella. Hardware Ray Tracing tarjoaa ympäristössä todenmukaisemmat pehmeät varjot ja heijastukset, mutta sen käyttäminen huonontaa suorituskykyä huomattavasti. (Epic Games i.a.)

Lumenia käytettäessä ympäristön suurista polygonimääristä koostuvien assettien tulisi Unreal Enginen dokumentaation mukaan hyödyntää Nanite-teknologiaa, kun mahdollista, muuten Lumenilla on suorituskykyyn heikentävä vaikutus. Jos Naniten käyttäminen ei ole mahdollista, assettien tulee hyödyntää LOD-teknologiaa, mutta senkin kanssa Lumen voi vaikeuttaa haluttuun ruudunpäivitysnopeuteen pääsemistä.

Virtuaaliympäristö pystyy hyödyntämään dynaamista ja staattista valaistusta myös samaan aikaan. Suorituskykyä voidaan parantaa asettamalla paikallaan pysyvät valot staattiseksi, mutta pitämällä muokattava valonlähde dynaamisena.

Molempien, dynaamisten sekä staattisten, valotyyppien optimoimiseksi kannattaa huolehtia myös niiden vaikutussäteestä (attenuation radius) sekä intensiteetistä. Valon ympärille piirtyvä säde määrittää alueen, jossa pelimoottori laskee valon vaikutuksen. Säde kannattaa asettaa sisältämään vain pinnat, joihin valon on tarkoitus vaikuttaa. Suurempi säde vaatii pelimoottorilta enemmän laskentaa, vaikkei ympäristö muuttuisi visuaalisesti. (Arnbjörnsson 2022). Intensiteetin suhteen tulisi huomioida, että erittäin suuri intensiteetti voi aiheuttaa Lumenin kanssa visuaalisia häiriöitä ja nolla-arvon intensiteetin omaavakin valo vaatii pelimoottorilta laskentatehoja, joten näitä tulisi välttää.

Valaistuksen suunnitteleminen ja rakentaminen virtuaaliympäristöön on saavutettavan realismin sekä suorituskvyn kannalta merkittävä vaihe. Kuten optimoiminenkin, valaiseminen vaatii lähes aina kompromisseja muokattavuuden, suorituskvyn sekä realistisuuden välillä. Toimivan valaistuksen luominen vaatii ymmärrystä ja kokemusta eri valaisukeinojen sekä -teknologioiden vaatimuksista sekä tuloksista, mikä mahdollistaa optimaalisten keinojen valitsemisen.

5 Toiminnallinen osuus

Opinnäytetyöni projektiosuudessa rakensin virtuaalituotantoa varten luodun 3D-ympäristön pelimoottori Unreal Enginellä. Toiminnallisen osuuden tarkoituksena on tukea opinnäytetyön tarjoamaa tietoa käytännön esimerkein. Tulevissa kappaleissa esittelen rakentamani virtuaaliympäristön ja kerron sen luomiseen ja optimoimiseen käytetyistä menetelmistä. Otin projektissani tärkeimmiksi tavoitteiksi vertailla tapoja geometrioiden sekä valaistuksen käsittelemiseen optimaalisimmilla tavoilla, koska niillä on suurimmat vaikutukset suorituskvyn. Toiminnallisessa osuudessa avaan myös opinnäytetyössä mainittujen uusien teknologioiden sekä työkalujen vaikutusta ympäristöihin ja vertailen niitä vaihtoehtoihin menetelmiin.

5.1 Virtuaaliympäristöni rakentaminen

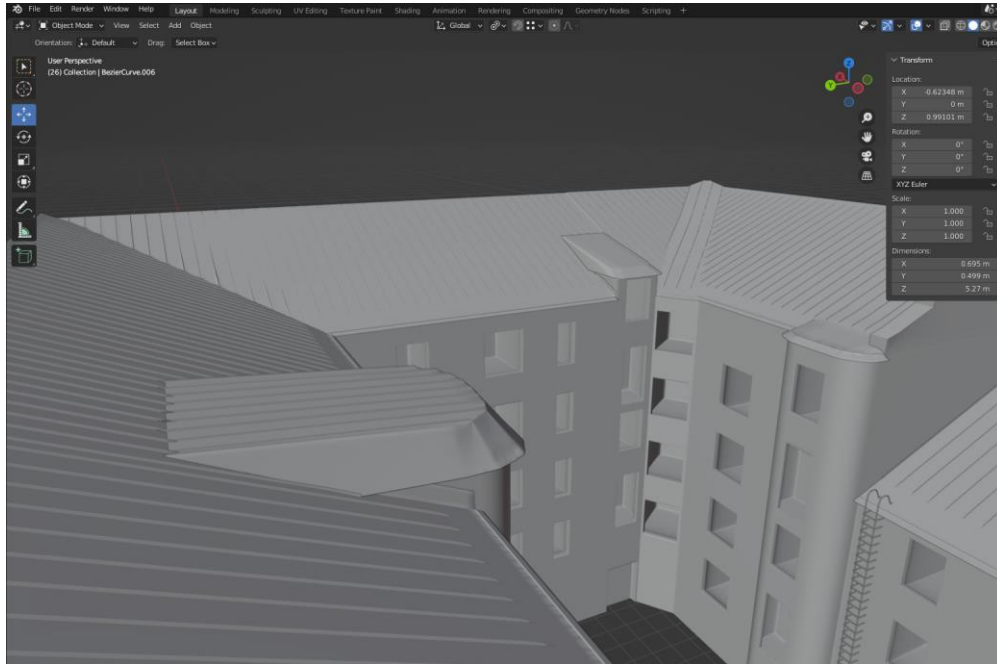
Ympäristöni teemana toimi futuristinen Helsinki. Projektin ympäristö kuvaa kaukaisen tulevaisuuden Helsinkiä, jossa jugend-tyylisen sisäpihan taakse horisonttiin on rakennettu korkeuksiin kohoava futuristin pilvenpiirtäjäkeskittymä kuvassa 14 näkyvän konseptitaidteeni tapaisesti.



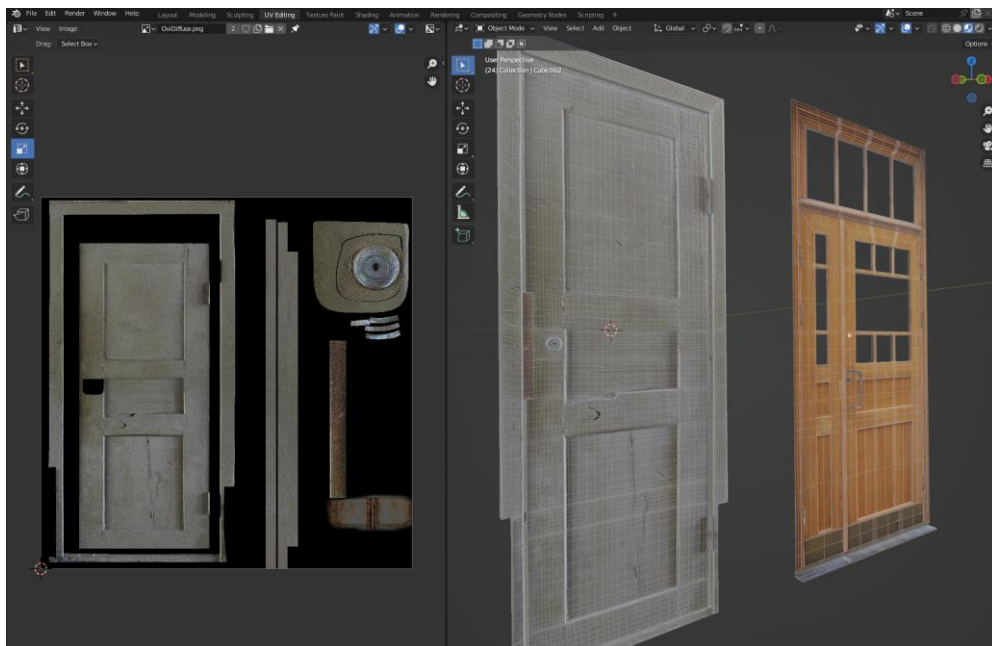
Kuva 14. Ympäristöni suunnitteluvaiheen konseptitaidetta

Asettien mallinnukseen ja hankintaan käytin 3D-mallinnusohjelmaa Blenderiä, verkosta löytyviä, julkiseen käyttöön tarkoitettuja 3D-malleja tarjoavia sivustoja, kuten TurboSquid ja CGTrader sekä nykyään Epic Gamesiin kuuluvaa Quixel-kirjastoa, joka tarjoaa laajan valikoiman materiaaleja sekä fotogrammetrialla luotuja 3D-asetteja. Kuvat 15 ja 16 havainnollistaa mallintamis- ja teksturointiprosessiani, jonka tein Blenderissä ennen 3D-mallien tuomista Unreal Engineen. 3D-mallien lisäksi käytin ympäristössä fotogrammetriaa ja valokuvia, joista jälkimmäiset olivat tärkeässä roolissa teksturoinnissa. Kokoon

elementit pelimoottori Unreal Enginen sisällä (versio 5.1.1) ja olen lisännyt käyttämäni tietokonen ominaisuudet kuvan 17 muodossa laitteiston vaikuttaessa olennaisesti suorituskykyyn. Käytössäni ollut laitteisto mahdollisti esimerkiksi Hardware Ray Tracing -teknologian käytön.



Kuva 15. Mallinsin sisäpihan geometrian 3D-mallinnusohjelmalla Blenderillä.



Kuva 16. Projisoin ottamiani valokuvia geometriaan realististen tekstuureiden saavuttamiseksi.

System	
Manufacturer:	Metropolia Helpdesk
Processor:	12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz
Installed memory (RAM):	32,0 GB (31,7 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	Pen Support

Kuva 17. Käyttämäni tietokoneen ohjelmisto- sekä laitteisto-ominaisuudet

LED-volyymin käytön kuormittavuuden huomioon ottamiseksi asetin Unreal Enginen näkymän korkealaatuisimpaan Cinematic-asentoon. Unreal Enginen ohjeistuksen mukaisesti varmistin myös tietokoneen järjestelmäasetuksista, että monitorini käytti korkeinta mahdollista resoluutiota, eli tässä tapauksessa 2560 kertaa 1440 pikseliä.

5.2 Rakennetun virtuaaliympäristön esittely

Valitsin ympäristöstäni kolme näkymää (Kuvat 18, 19 ja 20), joihin asetin liikkumattomat kamerat. Tämä mahdollisti ympäristön ja siihen tehtyjen muutosten selkeän tarkastelun ja vertailun, sillä kameran liikkumisella ja näkymän muuttumisella voi olla suorituskyykyyn suurikin vaikutus. Valitsin näkymät sen perusteella, että ne toimisivat virtuaalituotannoissa. Tämä tarkoittaa sen varmistamista, ettei kameroiden läheisyydessä etualalla ole virtuaaliassetteja, jotka olisivat päällekkäin konkreettisen lavasteen kanssa.



Kuva 18. Ensimmäinen virtuaaliympäristöni näkymä



Kuva 19. Toinen virtuaaliympäristöni näkymä



Kuva 20. Kolmas virtuaaliympäristöni näkymä

Ensimmäinen versio ympäristöstä on rakennettu käyttäen Unreal Enginen vakioasetuksia. Se hyödyntää kaikessa geometriassa Nanite-teknologiaa ja valaistuksessa käytetään dynaamista Lumen-teknologiaa. Unreal Enginen vakioasetukset tarjosivat hyvän suorituskyvyn ja visuaalisen laadun ympäristölleni jo ennen optimointia, vaikka olin yrittänyt rakentaa siitä mahdollisimman raskaan. Ensimmäisen näkymän vaatima renderöintiaika oli 17,88 millisekuntia, toisen 21,02 millisekuntia ja kolmannen 22,87 millisekuntia. Tämä tarkoitti ympäristöni ruudunpäivitysnopeuden olevan noin 42–52 ruutua sekunnissa, riippuen kameran sijainnista.

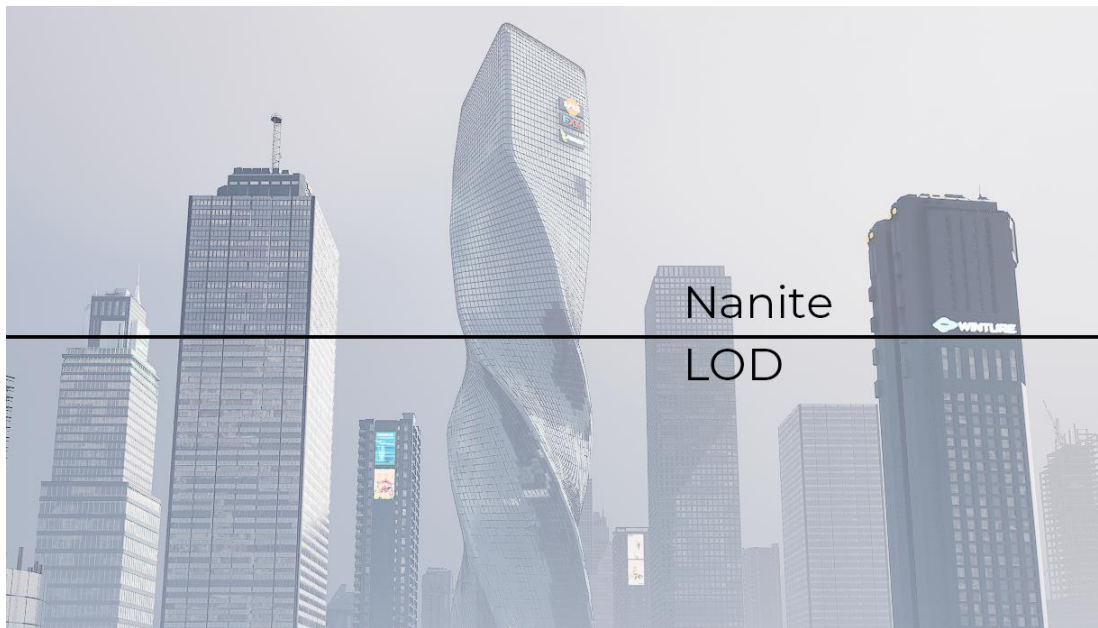
5.3 Rakennetun virtuaaliympäristön optimointi

Ympäristöni valmistuttua pyrin ensimmäisenä suorittamaan yksinkertaisimmat optimointivaiheet. Suljin esimerkiksi lisäosat, joita en projektissani tarvinnut. Näistä olennaisimmat olivat VR-tarkoituksiin suunnitellut lisäosat, joiden poistaminen käytöstä laski ruudun renderöintiaikaa reilun millisekunnin verran. Tämän jälkeen, kuten kuva 21 näyttää, pienensin sisäpihan valonlähteiden vaikutussädettä, huolehtien kuitenkin, että niiden visuaalinen vaikutus ei muuttunut. Säteen asettaminen sopivan kokoiseksi vähensi valojen laskemiseen vaadittua aikaa virtuaaliympäristössäni reilun millisekunnin. Yhdistin myös ensimmäisessä näkymässä esiintyviä rakennuksia ja kattoja, jotka alun perin koostuivat monista eri aseteista, vähentääkseni yksittäisten asettien määrää. Geometrioiden yhdistämisellä ei lopulta ollut suurta vaikutusta suorituskykyyn, mikä johtui suurelta osin kameran sijainnista. Vaikka sisäpihan taakse oli asetettu monia asetteja, kameran sijainnin takia niistä ei näkynyt kuin pieni osa. Tämä on osoitus pelimoottoreiden kyvystä tunnistaa ja renderöidä vain asetteja, jotka ovat nähtävissä ruudulla.



Kuva 21. Kuvassa on nähtävissä sisäpihalle asetettujen valojen vaikutusalueet, jotka oikealla puolella ovat asetettu sopivampaan kokoon kuormittavuuden vähentämiseksi.

Seuraavaksi otin tavoitteekseni tutkia Lumenin käytön vaikutuksia ilman Nanite-tekniikan hyödyntämistä. Loin ympäristöni aseteille LODit käyttämällä Unreal Enginen automaattista toimintaa ja kytkin Naniten pois päältä. Asetin LODit niin, että taustalla näkyvät rakennukset olivat vähemmän yksityiskohtaisia ja etualan assetit olivat täysilaatuisia. Lumenin käyttö ilman Nanitea aiheutti kuitenkin suurta kuormittavuutta, mikä näkyi tässä ympäristössä ruudunpäivitysnopeuden hidastumisella yli kolmellakymmenellä millisekunnilla. Automaattisesti luodut LODit aiheuttivat myös tietyissä geometrioissa yksityiskohtien laadun heikkenemistä, kuten kuva 22 osoittaa. Polygonimäärien vähentäminen geometrioissa sekä kustomoitujen LODien tekeminen vähentäisi assettien ja dynaamisen valon kuormittavuutta sekä auttaisi säilyttämään tarvittavat yksityiskohdat.



Kuva 22. Unreal Engineissä automaattisesti luodut LODit voivat heikentää yksityiskohtien laatua

Mallinnus- ja teksturointivaiheen tärkeys tuli geometrian kuormittavuuden lisäksi ilmi beikatun valon toimivuudessa. Joidenkin käyttämieni 3D mallien UV kartoissa ilmeni pintojen päällekkäisyyttä tai muita virheitä, jotka johtivat staattisen valon toimimattomuuteen. Virheelliset UV valokartat ilmenivät ympäristössäni beikkauksen aiheuttamina tummina alueina assettien pinnoissa. Ongelman ratkaiseminen vaati UV karttojen korjaamista mallinnus- tai teksturointiohjelmassa.

Beikkaamisen laatuun vaikuttavia muita tekijöitä on esimerkiksi GPU Lightmass -työkalun asetukset sekä valokarttatekstuurien resoluutio. Beikatun valaistuksen laatua voidaan parantaa edellä mainittuja ominaisuuksia säätämällä halutun lopputuloksen saavuttamiseksi, mutta korkeampi laatu vaatii aina pidemmän ajan valojen laskemiseen.

Ympäristöni sumuinen taivas ja sisäpihan laajat pyöreät valot edellyttivät pehmeitä varjoja realismin edistämiseksi. Laadukkaat pehmeät varjot puolestaan vaativat Hardware Ray Tracing -teknologian käyttöä, jota voi hyödyntää Lumenin tai GPU Lightmass -työkalun kanssa. Kuva 23

havainnollistaa valonlähteiden langettamien varjojen eroa Lumenin hyödyntäessä ja ollessa hyödyntämättä Hardware Ray Tracing -teknologiaa. Kyseisen teknologian kuormittavuus kuitenkin laskee ruudunpäivitysnopeutta suuresti, mikä johti beikattuun valoon turvautumiseen sisäpihan näkymissä.



Kuva 23. Vasemmanpuoleisessa kuvassa valaistuksessa käytetty Hardware Ray Tracing mahdollistaa pehmeämmät varjot suorituskyvyn kustannuksella, kun taas oikeanpuoleisessa kuvassa valaistukseen käytetty Lumen langettaa terävämmät varjot.

GPU Lightmass -työkalun käyttö mahdollisti pehmeiden varjojen saavuttamisen ilman suorituskyvyn madaltumista ja valojen beikkaamisen jälkeen ruudun renderöiminen kesti vain noin 10 millisekuntia.

Heijastukset aiheuttivat lopulta isoimman haasteen suorituskyvyn sekä visuaalisuuden kannalta. Heijastusten rooli on ympäristössäni suuri sisäpihan ollessa täynnä ikkunoita ja katonkin materiaalin ollessa osittain heijastava. Vaikka GPU Lightmass mahdollistaa korkealaatuisen valaistuksen tallentamisen, se ei kykene tallentamaan heijastuksia niiden muuttuessa kameran liikkeen mukaan. Heijastuksiin voidaan käyttää Hardware Ray Tracing -teknologiaa beikatun valaistuksen kanssa, mutta se siltikin laskee suorituskykyä huomattavasti. Ruudunpäivitysnopeuden säilymisen mahdollistavien Reflection Capture -työkalujen luomat heijastukset toimivat

melko uskottavasti ympäristöni sisäpihan näkymässä, mutta katon sekä pilvenpiirtäjien materiaalien kanssa ne eivät toimineet halutulla tavalla. Tästä johtuen totesin Lumen valaistuksen tarjoavan realistisemman jäljen ensimmäisessä näkymässä ja sen käyttö ilman Hardware Ray Tracing -teknologiaa mahdollisti ruudun renderöimisen 15 millisekunnissa.

GPU Lightmass -työkalun käyttö mahdollisti parhaan suorituskyvyn virtuaaliympäristössäni ja koen sen käytön edulliseksi varsinkin ympäristöissä, joissa kiiltäviä tai heijastavia pintoja ei ole runsaasti. Realistimman lopputuloksen saavutin käyttämällä Hardware Ray Tracing -teknologiaa, jolloin en kuitenkaan päässyt tavoittelemiini ruudunpäivitysnopeuksiin. Tapauksissa, joissa valaistusta haluttaisiin pystyä muuttamaan reaaliajassa, löysin toimivimmaksi vaihtoehdoksi staattisen ja dynaamisen valaisun yhdistelemisen. Asetin sisäpihan lamput staattisiksi ja beikkasin ne käyttämällä GPU Lightmass -työkalua, mutta pidin auringonvalon dynaamisena, mikä nopeutti valojen laskemisen vievää aikaa.

6 Yhteenveto

Opinnäytetyön yhtenä tärkeimmistä tavoitteista oli pelimoottori Unreal Engineä käyttäviin virtuaalituotantoihin sovellettavien optimointikeinojen löytäminen. Vaikka Unreal Enginen kehittäjä Epic Games tarjoaa runsaasti dokumentaatioita ympäristöjen rakentamisesta sekä optimoimisesta, niiden käyttötarkoitus koskee pääsääntöisesti pelejä. Haasteeksi osoittautuikin tunnistaa virtuaalituotantoihin sovellettava tieto. Virtuaalituotannoista saatavilla oleva aineisto koostuu pääosin ohjelmistojen tarjoajien tuottamista ohjeistoista ja sisällöistä, joita niiden vastaanottajien muodostamat yhteisöt soveltavat ja muotoilevat helpommin saavutettavaksi. Yritysten sekä niiden palvelua käyttävien yhteisöjen vuorovaikutus näkyy vahvasti myös Epic Gamesin toiminnassa. He tarjoavat dokumentaatioiden lisäksi arvokasta tietoa ja vastauksia kysymyksiin esimerkiksi Dev Community -fooruminsa ja youtube-kanavansa kautta.

Yhteisöjen hyödyntämisen tärkeys tuli vahvasti esille opinnäytetyöni tekemisen yhteydessä. Vaikka suurin osa aineistostani koostuu virallisesta dokumentaatiosta, toimi yhteisön tarjoama apu vähintään yhtä olennaisessa roolissa. Harrastajien sekä ammattilaisten henkilökohtaisesti laatimat tutoriaalit sekä tiivistelmät toimivat tärkeinä käytännönesimerkkeinä teoreettisen dokumentaation tukena.

Teknologian ja pelimoottorien kehitys näkyy myös optimointikeinoissa, joiden tarve vähenee koko ajan. Pelkästään viimeisen vuoden sisällä monet Unreal Enginen optimointia koskevat ohjeistukset ovat saaneet rinnalleen uusia, tarjoten vaihtoehtoisia menetelmiä ja teknologioita käytettäväksi. Virtuaalituotannot ovat ala, jossa toivottuun lopputulokseen voi päästä monella tavalla. Tärkeimpiä ominaisuuksia alalla työskenneltäessä vaikuttaakin olevan monipuolisuus ja kyky tunnistaa tuotantokohtaisesti hyödyllisimpiä työtapoja ja teknologioita.

Monipuolisuuden tärkeys tuli ilmi vahvasti opinnäytetyön toiminnallisessa osuudessa. Se osoitti ettei virtuaaliympäristöjen optimointiin ole yhtä oikeaa menetelmää, jota voidaan käyttää tilanteesta riippumatta. Virtuaaliympäristön optimointi alkaa jo suunnitteluvaiheessa ja on läsnä koko prosessin ajan. Tällöin rakennettu ympäristö pystytään optimoimaan tavalla, joka ottaa huomioon tuotannon sille asettamat vaatimukset. Optimoinnin huomioiminen loppuvaiheessa vähentää vaihtoehtoja ja pakottaa tuotannon muovautumaan ympäristön mukaan, eikä toisinpäin. Toiminnallisessa osuudessa optimointi olisi tullut ottaa huomioon jo asettien mallinnusvaiheessa ja taiteellisissa päätöksissä, mikä olisi mahdollistanut optimointikeinojen monipuolisemman käytön ja paremman suorituskyvyn. Projektini osoitti kuitenkin, että teknologisten kehitysten kuten Naniten ja Lumenin tarjoaman joustavuuden myötä on paremman suorituskyvyn lisäksi koko ympäristönrakentamisprosessi muuttunut saavutettavammaksi kokemuksesta tai resursseista riippumatta.

LED-volyymiä käyttävien virtuaalituotantojen tekninen kehitys sekä yleistyminen vaikuttaa jatkuvan nopeaan tahtiin ympäri maailman. Alan uutuus näkyy vakiintuneiden ohjeistusten sekä aineistojen vähyydessä ja saatavilla olevan aineistonkin tulkitseminen vaatii kattavaa pohjatietoa ja soveltamiskykyä. Ohjelmien päivitysten tuomat jatkuvat muutokset vaativat jatkuvaa ajan tasalla pysymistä, joten taito löytää ja päästä käsiksi ajankohtaisimpaan tietoon nousee erittäin tärkeään rooliin. Jatkuva muutos tarjoaa kuitenkin myös mahdollisuuden olla mukana tulevaisuuden alan kehityksessä, jota kukin voi edesauttaa omalla panoksellaan.

Lähteet

Adobe i.a. a. Frame Rate. Verkkosivu.

<https://www.adobe.com/creativecloud/video/discover/frame-rate.html> (viitattu 4.4.2023)

Adobe i.a. b. What is motion capture and how does it work? Verkkosivu.

<https://www.adobe.com/uk/creativecloud/animation/discover/motion-capture.html> (viitattu 27.3.2023)

Adobe i.a. c. 3D texturing and Adobe Substance 3D. Verkkosivu.

<https://www.adobe.com/mt/products/substance3d/discover/3d-texturing.html> (viitattu 2.4.2023)

Autodesk i.a. Textures. Verkkosivu.

<https://help.autodesk.com/view/MAYAUL/2022/ENU/?guid=GUID-1C24534D-8B91-45FF-971A-B3AD07E1067F> (viitattu 2.4.2023)

BBC Click 2019. Jon Favreau On The Lion King, The Mandalorian And Virtual Production - BBC Click. Verkkovideo 27.7.2019. Youtube. 17:47.

https://www.youtube.com/watch?v=0aiAP_zjubE (viitattu 17.4.2023)

Blender i.a. UV Editor. Verkkosivu.

<https://docs.blender.org/manual/en/latest/editors/uv/introduction.html> (viitattu 13.4.2023)

Brooks, Michael 2019. VFX in movies: from weightlessness to curly hair.

Verkkosivu. <https://physicsworld.com/a/vfx-in-movies-from-weightlessness-to%E2%80%AFcurly-hair/> (viitattu 30.3.2023)

Dalkian, Sevan i.a. nDisplay Technology. Unreal Engine.

https://cdn2.unrealengine.com/Unreal+Engine%2FnDisplay-whitepaper-final-updates%2FnDisplay_Whitepaper_FINAL-f87f7ae569861e42d965e4bffd1ee412ab49b238.pdf (viitattu 2.4.2023)

Das, Siddhartha 2022. James Cameron's 'Avatar' Is An Innovation Of Cinematic Visuals. Verkkosivu.

<https://fugitives.com/avatar-cinematic-visuals-explained-2022-fantasy-action-james-cameron/> (viitattu 2.4.2023)

Eggen, Eric i.a. Low to High Poly Animal – Giraffe. Verkkosivu.

https://eric_eggen.artstation.com/projects/JIR16D (viitattu 10.4.2023)

Epic Games 2022a. Optimizations. Verkkosivu.

<https://dev.epicgames.com/community/learning/courses/eER/unreal-engine-technical-guide-to-linear-content-creation-production/4G4r/unreal-engine-optimizations> (viitattu 2.4.2023)

Epic Games 2022b. Optimizing the Medieval Game Environment. Verkkosivu. 2.6.2022. <https://dev.epicgames.com/community/learning/talks-and-demos/585Y/optimizing-the-medieval-game-environment> (viitattu 31.3.2023)

Epic Games 2022c. VFX vs Real-Time Best Practice for Asset Preparation. Verkkosivu. <https://dev.epicgames.com/community/learning/tutorials/8kRj/unreal-engine-vfx-vs-real-time-best-practice-for-asset-preparation> (viitattu 2.4.2023)

Epic Games i.a. a. GPU Lightmass Global Illumination. Verkkosivu. <https://docs.unrealengine.com/5.1/en-US/gpu-lightmass-global-illumination-in-unreal-engine/> (viitattu 27.3.2023)

Epic Games i.a. b. Guidelines for Optimizing Rendering for Real-Time. Verkkosivu. <https://docs.unrealengine.com/5.1/en-US/guidelines-for-optimizing-rendering-for-real-time-in-unreal-engine/> (viitattu 27.3.2023)

Epic Games i.a. c. In-Camera VFX Best Practices. Verkkosivu. <https://docs.unrealengine.com/5.1/en-US/in-camera-vfx-best-practices-in-unreal-engine/> (viitattu 2.4.2023)

Epic Games i.a. d. In-Camera VFX Overview. Verkkosivu. <https://docs.unrealengine.com/4.26/en-US/WorkingWithMedia/InCameraVFX/InCameraVFXOverview/> (viitattu 31.3.2023)

Epic Games i.a. e. Light Types and Their Mobility. Verkkosivu. <https://docs.unrealengine.com/5.0/en-US/light-types-and-their-mobility-in-unreal-engine/> (viitattu 21.3.2023)

Epic Games i.a. f. nDisplay Technology. Verkkosivu <https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/#supportedfeaturesofnanite> (viitattu 2.4.2023)

Epic Games i.a. g. Physically Based Materials. Verkkosivu. <https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/Materials/PhysicallyBased/> (viitattu 24.3.2023)

Epic Games i.a. h. Post Process Effects. Verkkosivu. <https://docs.unrealengine.com/5.1/en-US/post-process-effects-in-unreal-engine/> (viitattu 29.3.2023)

Epic Games i.a. i. Ray Tracing Performance Guide. Verkkosivu. <https://docs.unrealengine.com/5.1/en-US/ray-tracing-performance-guide-in-unreal-engine/> (viitattu 27.3.2023)

Epic Games i.a. j. Real Shading in Unreal Engine 4. Verkkosivu. <https://docs.unrealengine.com/5.1/en-US/optimizing-and-debugging-projects-for-real-time-rendering-in-unreal-engine/> (viitattu 2.4.2023)

Epic Games i.a. k. Stat commands. Verkkosivu.

<https://docs.unrealengine.com/5.1/en-US/stat-commands-in-unreal-engine/> (viitattu 31.3.2023)

Epic Games i.a. l. Static Meshes. Verkkosivu.

<https://docs.unrealengine.com/5.0/en-US/static-meshes/> (viitattu 27.3.2023)

Epic Games i.a. m. Streaming Virtual Texturing. Verkkosivu.

<https://docs.unrealengine.com/5.1/en-US/streaming-virtual-texturing-in-unreal-engine/> (viitattu 2.4.2023)

Epic Games i.a. n. Texture Properties. Verkkosivu.

<https://docs.unrealengine.com/4.26/en-US/RenderingAndGraphics/Textures/Properties/> (viitattu 27.3.2023)

Epic Games i.a. o. Unreal Insights Overview. Verkkosivu.

<https://docs.unrealengine.com/4.26/en-US/TestingAndOptimization/PerformanceAndProfiling/UnrealInsights/Overview/> (viitattu 5.4.2023)

Epic Games i.a. p. What is real-time ray tracing, and why should you care?

Verkkosivu. <https://www.unrealengine.com/en-US/explainers/ray-tracing/what-is-real-time-ray-tracing> (viitattu 23.3.2023)

Evanson, Nick 2021. How 3D Game Rendering Works: Vertex Processing.

Verkkosivu. <https://www.techspot.com/article/1857-how-to-3d-rendering-vertex-processing/> (viitattu 2.4.2023)

Foundry i.a. Texturing the dystopian landscapes of Blade Runner 2049.

Verkkosivu. <https://www.foundry.com/insights/film-tv/texturing-blade-runner> (viitattu 2.4.2023)

Gregory, Jason 2014 Game Engine Architecture. Florida: CRC Press

Helminen, Markus 2023. VP Supervisor. Fireframe Studios. Haastattelu 14.4.2023.

Hridja 2022. High poly vs Low poly in 3D Modeling explained in simple terms.

Verkkosivu. <https://www.queppelin.com/high-poly-vs-low-poly-in-3d-modeling/> (viitattu 17.4.2023)

Industrial Light & Magic 2019. ILM STAGECRAFT™. Verkkosivu.

<https://www.ilm.com/ilm-stagecraft/> (viitattu 30.3.2023)

Industrial Light & Magic 2020. The Virtual Production of The Mandalorian Season One. 20.2.2020. Youtube. 4:42.

<https://www.youtube.com/watch?v=gUnxzVOs3rk> (viitattu 2.4.2023)

Industrial Light & Magic 2022. ILM STAGECRAFT™ HONORED WITH ENGINEERING, SCIENCE & TECHNOLOGY EMMY® AWARD. Verkkosivu. <https://www.ilm.com/television-academy-announces-recipients-of74th-engineering-science-technologyemmy-awards/> (viitattu 2.4.2023)

Industrial Light & Magic i.a. THE MANDALORIAN (SEASON 1). Verkkosivu. <https://www.ilm.com/vfx/the-mandalorian/> (viitattu 2.4.2023)

Kadner, Noah 2021. Virtual Production Essentials: 10 Things to Know Before You Start Shooting. 7.9.2021 Frame.io. <https://blog.frame.io/2021/09/07/10-things-to-know-about-virtual-production/> (viitattu 28.3.2023)

Kadner, Noah i.a. a. The Virtual Production Field Guide. Epic Games. <https://cdn2.unrealengine.com/Unreal+Engine%2Fvpfieldguide%2FVP-Field-Guide-V1.2.02-5d28ccec9909ff626e42c619bcbe8ed2bf83138d.pdf> (viitattu 16.4.2023)

Kadner, Noah i.a. b. The Virtual Production Field Guide Volume 2. Epic Games. <https://cdn2.unrealengine.com/Virtual+Production+Field+Guide+Volume+2+v1.0-5b06b62cbc5f.pdf> (viitattu 16.4.2023)

Karis, Brian 2013. Real Shading in Unreal Engine 4. Epic Games. 1-2. <https://de45xmedrsdbp.cloudfront.net/Resources/files/2013SiggraphPresentationsNotes-26915738.pdf> (viitattu 2.4.2023)

McDermott, Wes 2018. The PBR Guide. Verkkojulkaisu. <https://substance3d.adobe.com/tutorials/courses/the-pbr-guide-part-1> (viitattu 2.4.2023)

Oseman, Neil 2022. THE HISTORY OF VIRTUAL PRODUCTION. Verkkosivu. <https://neiloseman.com/the-history-of-virtual-production/> (viitattu 27.3.2023)

Park, Sukjun & Baek, Nakhoon 2021. A Shader-Based Ray Tracing Engine. Verkkoartikkeli. Applied Sciences. MDPI. 1. (viitattu 3.4.2023)

Patel, Maurice i.a. The New Art of Virtual Moviemaking http://images.autodesk.com/apac_grtrchina_main/files/the_new_art_of_virtual_moviemaking_-_autodesk_whitepaper1.pdf (viitattu 29.3.2023)

Perforce 2023. In-Camera VFX — How Industry Experts are Using ICVFX. Verkkosivu. <https://www.perforce.com/blog/vcs/in-camera-VFX> (viitattu 28.3.2023)

Perforce 2023. What Are the Best Game Engines? Verkkosivu. <https://www.perforce.com/blog/vcs/most-popular-game-engines> (viitattu 7.4.2023)

Tan, Cherie 2022. Textures vs. Materials in 3D Modeling: What's the Difference?. Verkkosivu. <https://www.makeuseof.com/textures-vs-materials-3d-modeling/> (viitattu 2.4)

Taru Sormusten Herrasta: Sormusten ritarit (The Lord of the Rings: The Fellowship of the Ring). Uusi Seelanti 2001. Ohjaus Peter Jackson. 178 min. DVD-Koelma.

The Chernobyl 2022. Materials and Physically Based Rendering // Ray Tracing series. 31.10.2022. Youtube. 27:17
https://www.youtube.com/watch?v=7_iy4Rov2Ck&t=454s (viitattu 2.4.2023)

Universal Pictures All-Access 2013. Oblivion - Sky Tower. 25.6.2013. Youtube. 02:12. <https://www.youtube.com/watch?v=sx1U2VsCnLg> (viitattu 27.2.2023)

Unreal Engine 2020a. 'Game of Thrones': Virtual Production | Spotlight | Unreal Engine. Verkkovideo 18.3.2020 Youtube. 03:07.
<https://www.youtube.com/watch?v=3tSxWTBdd54&t=101s> (viitattu 30.3.2023)

Unreal Engine 2020b. How to Get Started With Real-Time Ray Tracing | Get to Know. 25.6.2020. Youtube. <https://www.youtube.com/watch?v=Bk5HOtRwgd8> (viitattu 2.4.2023)

Unreal Engine 2021a. Lumen | Inside Unreal. 10.6.2021. Youtube. 2:34:51.
https://www.youtube.com/watch?v=QdV_e-U7_pQ&t=1892s (viitattu 2.4.2023)

Unreal Engine 2021b. Unreal Engine 4.27 In-Camera VFX Tutorials | 7: Integrating CG & Practical Set. 15.11.2021. Youtube. 47:57.
https://www.youtube.com/watch?v=9S-B0ZnVDoQ&list=PLZlv_N0_O1gaXvxPtn8_THYN_Awx-VYeu&index=7 (viitattu 17.4.2023)

Vu Technologies i.a. The Ins and Outs of 3D Scanning. Verkkosivu.
<https://www.vu.network/blog/photogrammetry-and-lidar> (viitattu 29.3.2023)