

# VIRTUAALISEN TILAN OPTIMOINTI

CASE: Virtuaalinen pankkisali

## Tiivistelmä

Tekijä(t) Johansson, Sami	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 31	Valmistumisaika Kevät 2019
Työn nimi <b>Virtuaalisen tilan optimointi</b> Case: Virtuaalinen pankkisali		
Tutkinto Insinööri (AMK), Mediatekniikka		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli tuottaa virtuaalitodellisuuslaseilla katseltava virtuaalinen tila Lahdessa sijaitsevasta, Aleksio10 -liikekiinteistön tyhjillään olevasta pankkisalista. Opinnäytetyön case-osuus toteutettiin osana työharjoittelua Revulon Oy:ssä ja työn tilaajana toimi Aleksio10 -liikekiinteistön hallinnoija Torikapitaali Oy. Luodussa virtuaalissa ympäristössä pystyttiin liikkumaan VR-ohjaimen ohjelmoidun toiminnollisuuden avulla.</p> <p>Rakennuksesta ennestään olemassa olevan 3D-mallin muokkaamiseen ja uusien objektien luomiseen käytettiin 3Ds Max -mallinnusohjelmistoa. Virtuaalitodellisuuden sovellus luotiin käyttämällä Unity-pelimoottoria. Laitealustana käytettiin HTC Vive -virtuaalitodellisuuslaseja. Opinnäytetyön teoriaosuudessa kerrotaan VR-sovelluskehittämisen optimoinnin toimintaperiaatteista, 3D-mallien käsittelystä, virtuaalitodellisuudessa ilmenevistä visuaalisista haitoista ja liikepahoinvoinnista sekä Unity-pelimoottorissa luodun virtuaalisen tilan koostavista objekteista. Opinnäytetyön case-osiossa käsitellään suunnitteluprosessi ja kaksivaiheinen työtoteutus. Ensimmäisessä työosuudessa luotiin virtuaalitodellisuudelle optimoitu tyhjä tila, jossa pystyi katselemaan ympärille virtuaalitodellisuuden laseilla sekä liikkumaan VR-ohjaimen avulla. Toisessa työosuudessa tilalle luotiin ja järjesteltiin toimistosisustusratkaisu.</p> <p>Opinnäytetyön tavoite saavutettiin osittain. Case-osiossa käsiteltävää toista työosuutta ei ehditty toteuttaa loppuun ennen opinnäytetyön viimeistelyä, jonka seurauksena toisen työosuuden dokumentointi jäi keskeneräiseksi. Toinen työosuus suoritetaan loppuun toukokuun 2019 aikana.</p>		
Asiasanat virtuaalitodellisuus, 3D, 3Ds Max, Unity, VR, HTC Vive		

## Abstract

Author(s) Johansson, Sami	Type of publication Bachelor's thesis	Published Spring 2019
	Number of pages 31	
Title of publication <b>Optimizing a virtual space</b> Case: Virtual bank hall		
Name of Degree Bachelor of Engineering, Media Technology		
Abstract <p>The objective of this thesis was to create a virtual reality application where the environment was based on a real-world bank hall space in Aleksio10 business property located in Lahti, Finland. The case part of this thesis was done during an internship at Revulon Ltd. and the client was Torikapitali Ltd., which is the administrator of the Aleksio10 business property. Moving in the VR environment was enabled by programming teleportation function into a VR controller.</p> <p>The VR environment was made by optimizing a pre-existing 3D model of the property and modeling new parts. The 3Ds Max modeling software was used for 3D modeling and texture mapping. The application was built using the Unity game engine and made to run with the HTC Vive VR device. The theory part of the thesis presents information about optimization in VR development, visually induced motion sickness, 3D model production, and components and objects that are crucial for VR development in Unity. The practical part tells about the planning and execution of the two-phased project. The first phase consists of creating an empty VR bank hall environment with a moving functionality and the second phase consists of visualizing the bank hall space as an office environment.</p> <p>The main goal of the thesis was partly achieved. Virtual reality space was created and functional but visualizing the office environment could not be finished during the timeframe of the thesis. The second work phase will be carried out afterwards during May 2019.</p>		
Keywords virtual reality, 3D, 3Ds Max, Unity, VR, HTC Vive		

## SISÄLLYS

1	JOHDANTO .....	1
2	TOIMINTAPERIAATTEET VIRTUAALISEN TILAN OPTIMOINNISSA .....	2
2.1	Suorituskyvyn maksimointi.....	2
2.2	Visuaaliset haitat ja liikepahoinvointi .....	4
3	VIRTUAALISEN YMPÄRISTÖN KEHITTÄMINEN .....	7
3.1	Tekstuurikartoittaminen .....	7
3.2	3D-mallien vienti ja tuonti.....	8
3.3	Renderöinti Unity-pelimoottorissa .....	10
3.4	Materiaalit Unity-pelimoottorissa .....	13
3.5	Valaistus Unity-pelimoottorissa .....	16
3.6	SteamVR- ja Virtual Reality Toolkit -liitännäiset .....	19
4	CASE: VIRTUAALINEN PANKKISALI .....	21
4.1	Pankkisalin 3D-mallin optimointi ja tekstuurikartoitus .....	21
4.2	Pankkisalin 3D-mallin tuonti ja vienti .....	23
4.3	Materiaalien ja valaistuksen luonti .....	24
4.4	Toiminnollisuuksien toteuttaminen .....	25
4.5	Ensimmäinen koonti .....	26
4.6	Pankkisalin sisustuksen 3D-mallintaminen ja tekstuurikartoitus .....	27
4.7	Sisustuksen asettelu .....	28
4.8	Sisustuksen 3D-mallien tuonti ja vienti.....	29
4.9	Työn keskeneräisyys ja jatkosuunnitelmat .....	29
5	YHTEENVETO .....	31
	LÄHTEET .....	32

## 1 JOHDANTO

Ensimmäiset kuluttajakäytölle merkittävät virtuaalitodellisuuslasit ilmestyivät yhdeksänkymmentäluvulla. Alkeellinen virtuaalitodellisuusteknologia ei kuitenkaan kerännyt suurta asiakaskuntaa, kunnes 2010-luvulla virtuaalitodellisuuden kehittäminen ja sitä tukevat laitteet alkoivat yleistyä kovaa vauhtia. Vuoden 2016 aikana useampi eri valmistaja julkaisi oman kilpailevan versionsa virtuaalitodellisuuslaitteista. Kilpailun ja aiheen kasvun myötä virtuaalitodellisuuden sovellusten ja teknologioiden käyttö on lisääntynyt ja omaksuttu erilaisiin käyttötarkoituksiin, kuten viihde-, koulutus-, markkinointi- ja tuotekehityskäyttöön.

Opinnäytetyössä käsitellään virtuaalisen tilan luontiin liittyvää optimointia. Tavoitteena on luoda visuaalisesti inspiroiva virtuaalinen tila reaali maailman pankkialasta. Teknisiin tavoitteisiin kuuluu tarvittavan kuvataajuuden ylläpito sovelluksen ollessa käynnissä. Opinnäytetyön ongelmat liittyvät suorituskyvyn maksimointiin. Ongelmaa pyritään ratkaisemaan tasapainottamalla sovelluksen visuaalisuutta suorituskykyyn nähden. Suorituskyky ongelmaan pyritään vaikuttamaan 3D-mallien ja erilaisten kuvan piirtoon vaikuttavien teknologioiden optimoinnilla. Toteutuksessa käytetään 3DS Max -mallinnusohjelmistoa ja Unity-pelimootoria.

Teoriaosuudessa kerrotaan opinnäytetyön toteutukseen liittyvistä toimintaperiaatteista ja sovelluskehittämiseen liittyvistä teknologioista. Case-osiossa kerrotaan työn suunnittelemisesta ja työn vaiheista. Ensimmäisessä työosuudessa käydään läpi virtuaalisen tilan tuottaminen ja toisessa työosuudessa käydään läpi virtuaaliseen tilaan lisättävä toimistosisustusratkaisu. Yhteenvedossa reflektoidaan opinnäytetyön teon aikana opittua tietoutta ja pohditaan työn tulosta.

## 2 TOIMINTAPERIAATTEET VIRTUAALISEN TILAN OPTIMOINNISSA

### 2.1 Suorituskyvyn maksimointi

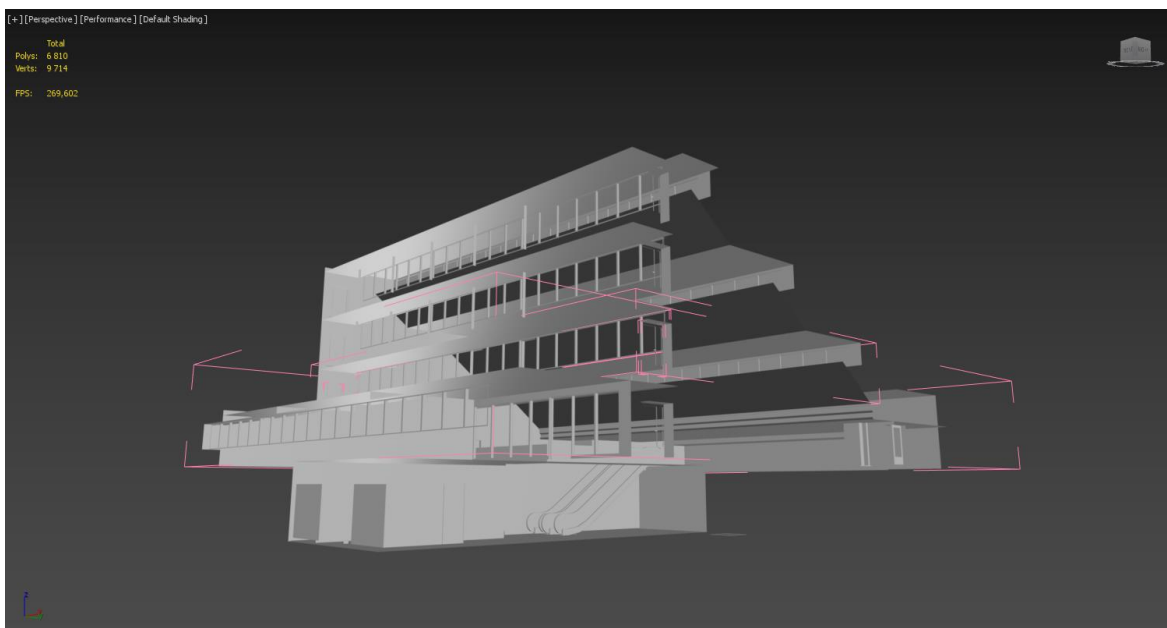
Virtuaalitodellisuuden sovellusten optimoinnissa noudatetaan suorituskyvyn maksimoinnin toimintaperiaatetta tuotannon eri osa-alueilla (Unity Technologies 2019f). Optimoinnin ensisijaisena tavoitteena on tasapainottaa sovelluksen suoritettavuuteen vaikuttavaa visuaalista rasiitetta, mutta myös ohjelmoinnin vaikutus suorituskykyyn korreloittuu sovelluksen laajuuteen nähden. Optimoimatta jättäminen kuormittaa tietokoneen kykyä suorittaa ensisijaisia toimintoja, koska tietokoneen resursseja menetetään toiminnoille, joita ei tarvitse suorittaa. Tämä on yksi kuvataajuuteen vaikuttavista tekijöistä suoritettaessa virtuaalitodellisuuden sovellusta. Apua suorituskykyyn ja kuvataajuuden ylläpitämiseen saa tehokkaammista tietokonekomponenteista, mutta noudattamalla optimoinnin toimintaperiaatteita, saadaan luotua sovellus, jota pystytään käyttämään halvemmän hintatason komponenteilla. Halvempien komponenttien käyttömahdollisuus parantaa saatavuutta ja tekee sovelluksesta asiakas- ja kuluttajaystävällisemmän.



Kuva 1. Varjo VR-1 -virtuaalitodellisuuslasit (Varjo Technologies 2019)

Optimoinnin keskeisin tekninen tavoite on suorituskyvyn maksimointi ja sitä kautta kuvataajuuden ylläpitäminen tavoitelukemissa (Unity Technologies 2019f). Virtuaalitodellisuuden sovellusten tavoiteltu kuvataajuus on yli yhdeksänkymmentä hertsiä. Kuvataajuudella

ilmaistaan kuvamäärää, jonka tietokoneen näytönohjain piirtää sekunnin aikana (Wikipedia 2018). Kuvamäärän esiintymistä näyttölaitteen näytöllä sekunnin aikana kutsutaan virkistystaajuudeksi (Wikipedia 2019b). Tämän hetkiset virtuaalitodellisuuslaitteet, kuten Oculus Rift, HTC Vive, PSVR, sekä Varjo VR-1, joka on nähtävissä kuvassa 1, toimivat yhdeksänkymmenen hertsin virkistystaajuudella. Laitteet siis kykenevät maksimissaan toistamaan yhdeksänkymmenen hertsin kuvataajuuden. Kuvataajuuden pudotessa alle virkistystaajuuden käyttäjä voi kokea haittavaikutuksia. Sovellukseen uppoutuminen, eli immersio, alkaa kärsiä ja kuvan piirrossa, eli renderöinnissä ilmenee virheitä ja käyttäjä voi kokea ristiriitaisista näköhavainnoista aiheutuvaa liikepahoinvointia.

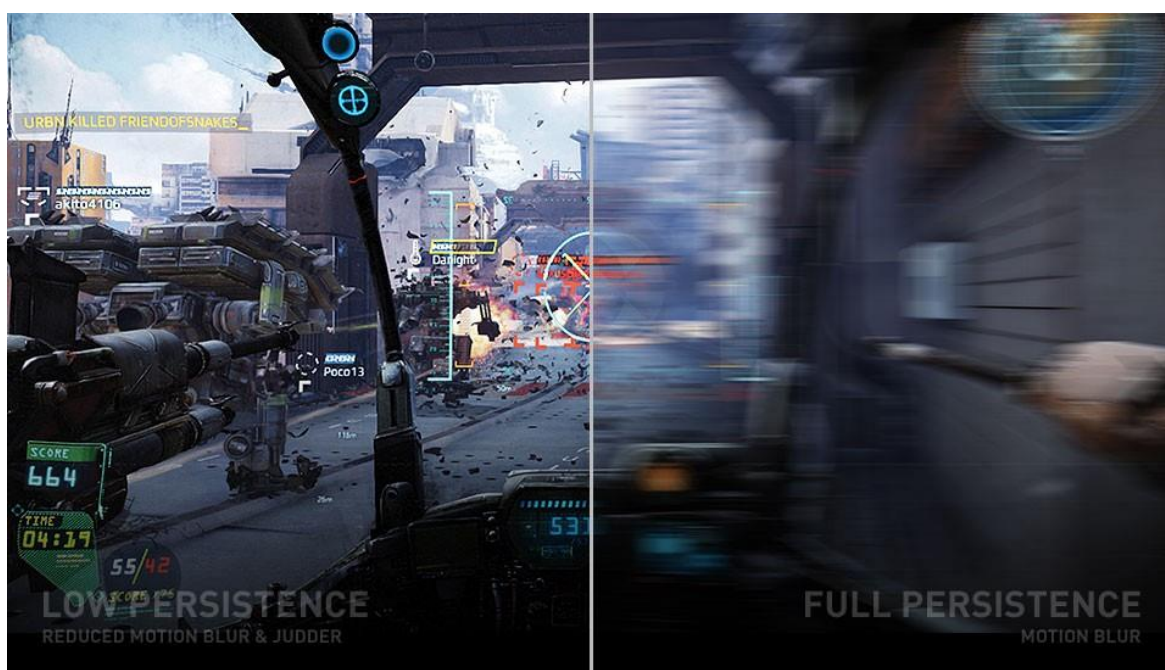


Kuva 2. Optimoitu ja yksipuolisista 3D-pinnoista koostuva 3D-malli

Optimoitaessa 3D-malleja virtuaalitodellisuudelle sopivaksi keskitytään yksinkertaisuuteen ja yksipuoleisuuteen. 3D-mallien polygonien määrällä on suora korrelaatio suorituskykyyn ajettaessa virtuaalitodellisuuden sovelluksia. Mitä monimuotoisempia ja tarkempia 3D-mallit ovat, sitä enemmän prosessoitavaa virtuaalitodellisuuslasien näkökentässä on. 3D-malleista pyritään poistamaan polygonit, joita ei tulla näkemään yleisistä katselukohdista tai peittyvät muun geometrian toimesta. Näkökenttään päin takaperin olevista yksipuoleisista 3D-pinnoista voidaan nähdä lävitse. Yksipuoleisista 3D-tasoista renderöidään vain toinen puoli, mikä vähentää renderöityjen polygonien määrää. 3D-pintojen yksipuoleisuus on nähtävissä kuvassa 2. Nähtävien osien tarkkuuteen kiinnitetään huomiota tekemällä kaarevista osista vain niin pyöreitä, kuin tarve vaatii. Yksityiskohtien luomisessa voidaan käyttää hyödyksi tekstuurikarttoja muuttamatta itse geometriaa.

## 2.2 Visuaaliset haitat ja liikepahoinvointi

Virtuaalitodellisuuden yleistymisen myötä ristiriitaisista näköhavainnoista aiheutunut liikepahoinvointi on ilmentynyt virtuaalitodellisuuslasien käytöstä. Ristiriitaisista näköhavainnoista aiheutuneeseen liikepahoinvointiin, eli visually induced motion sickness (VIMS), on useita vaikuttavia tekijöitä. Ihmiskohtaisesti vaikuttavia tekijöitä ovat ikä, etnisyys, sukupuoli sekä terveydentila. Teknologisesti yleisin pahoinvointiin suorasti tai epäsuorasti vaikuttava tekijä on viive. Viiveellä tarkoitetaan toiminnallisia sekä visuaalisia viiveitä, joita virtuaalitodellisuuden sovelluksessa tapahtuu. Virtuaalitodellisuuden sovelluksissa koettavista viiveistä johtuen ihmisen aistit viestivät eri tahdissa totuttuun verrattuna, mistä seuraa pahoinvointia. Viivettä eliminoidaan panostaen suorituskyvyn maksimointiin optimoimalla sovelluskehittämisen eri osa-alueilla. (Aukstakalnis 2017, 332 - 336.)



Kuva 3. Esitys kuvan pysyvyydestä ja sitä seuraavasta kuvavahahtelusta (Facebook Technologies, LLC. 2014)

Yksi virtuaalitodellisuuslasilla ilmenevistä haittailmiöistä on kuvavahahtelu, eli Judder. Kuvavahahtelu ilmenee kuvan välkkymisenä strobovalon kaltaisesti tai töhriintymisenä jossain kohtaa näytölle piirtyvää kuvaa. Kuvavahahtelu aiheutuu tilanteissa, joissa virtuaalitodellisuuslasien virkistystaajuus ei pysty päivittämään näytölle piirtyvää uutta kuvainformaatiota tarpeeksi nopeasti. Liian nopea välitön liikkuminen, eli teleportaatio, voi aiheuttaa strobovalon kaltaista välkkymistä optimoimattomassa virtuaalisessa tilassa. Siirryttäessä nopeasti, muutos edellisen ja seuraavan renderöitävän kuvan välillä voi vaikuttaa hetkelli-



sesti virkistystaajuuteen. Uuden sijainnin kuva ei kerkeä täysin päivittyä virtuaalitodellisuuslasien näytöille vanhan sijainnin kuvan ollessa vielä näytöille. Tätä kutsutaan kuvan pysyvyydeksi, eli persistence, joka on nähtävissä kuvassa 3. Joissakin kohdissa näyttöä sekä uuden että vanhan renderöidyn kuvan pikselit ovat valaistuna päällekkäin. Päällekkäin olevien kuvien myötä näytöllä valaistujen pikseleiden valovoimakkuus kasvaa normaalia suuremmaksi. Virtuaalitodellisuuslasien näytöillä näkyvä kuvavaihtelu muistuttaa valokuvan palamista puhki reaaliajassa. Kuvavaihtelu kestää sekunnista muutamiin sekunteihin, riippuen aiheuttajan toistuvuudesta ja laitteiston tehokkuudesta. (Abrash 2013.)



Kuva 4. Haamukuvien esiintyminen pelin renderöinnissä (Ziff Davis, LLC. 2011)

Virtuaalitodellisuuden laseissa esiintyvä visuaalinen haitta on kuvaviiveen vaihtelu, eli Jitter. Kuvaviiveen vaihtelulla tarkoitetaan virtuaalitodellisuuden lasien pyörimisliikkeessä ilmenevää visuaalista haittaa. Katsellessa ympärille virtuaalitodellisuuslaseille renderöitävät kuvat tulevat viiveellä liikkeeseen ja katselukohtaan nähden. Kuvaviiveen vaihtelu ilmenee kuvataajuuden pudotessa alle tavoitellun yhdeksänkymmenen rajan. Kuvaviiveen vaihtelu muistuttaa näkökentässä viiveellä liikkuvia haamukuvia, mikä on nähtävissä kuvassa 4. Toinen kuvaus on verrattavissa peleissä yleisesti käytettyyn liikesumennus-, eli motion blur -erikoistehosteeseen. (Wikipedia, 2019a.)

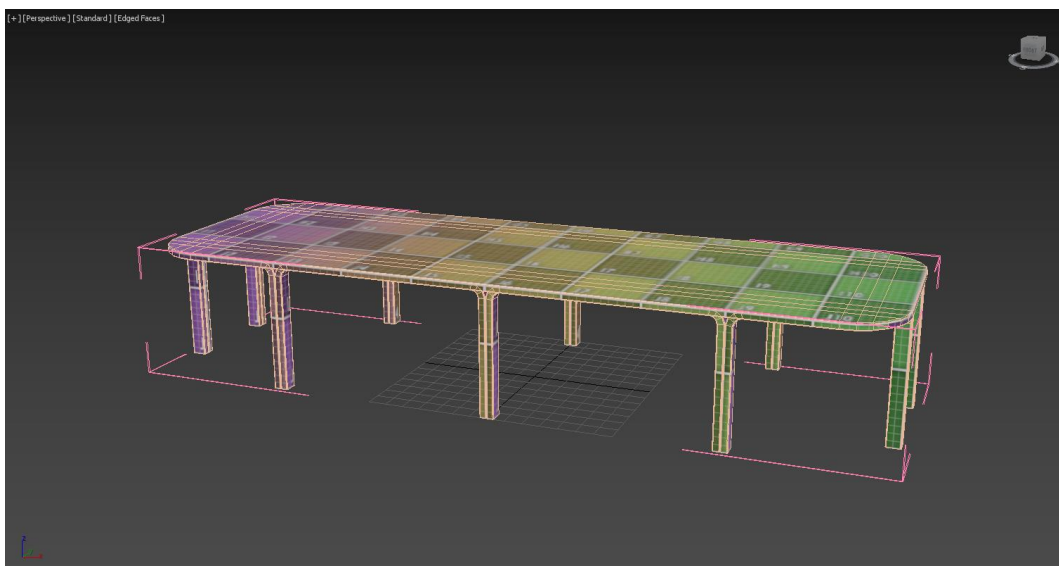
Virtuaalitodellisuudesta aiheutunut pahoinvointi ei aina johdu suoranaisesti visuaalisista haitoista. Virtuaalisessa ympäristössä on otettava huomioon myös käyttäjän liikkuminen paikasta toiseen. Virtuaalitodellisuuden sovelluksissa yleisiä liikkumistapoja ovat kävely,

juokseminen, teleportaatio tai liikkuminen jonkinlaisen kuljettimen kyydillä. Liikkeen nopeus ja kuvataajuus vaikuttavat ensisijaisesti kokemukseen. Reaalimaailmaa vastaavilla liikenopeuksilla tuetaan ihmisen normaalia aistikokemusta ja samalla parannetaan immersiota. Teleporttisiirtymistä voidaan pitää poikkeuksellisena liikkumistapana, koska siirtymä on lähes välitön. Käytettäessä teleportaatiota virtuaaliodellisuuden sovelluksessa siirtymässä voi olla siirtymätehoste liikepahoinvoinnin oireiden lievittämiseksi. Siirtymätehoste voi olla esimerkiksi lievä häivytyksen mustaan ja takaisin, jolloin liian nopeaa siirtymistä ei ehditä nähdä ja siitä seuraavaa liikepahoinvointia ei tule. Yhdessä nopeat siirtymät, tietynlainen liike suhteessa katselukohtaan ja visuaaliset haitat voivat aiheuttaa, henkilöstä riippuen muun muassa kuvotusta, uneliaisuutta, pyörryttämistä, päänsärkyä, keskittymisvaikeuksia, sumentunutta näköä ja silloin tällöin jopa oksentelua (Aukstakalnis 2017, 333.)

### 3 VIRTUAALISEN YMPÄRISTÖN KEHITTÄMINEN

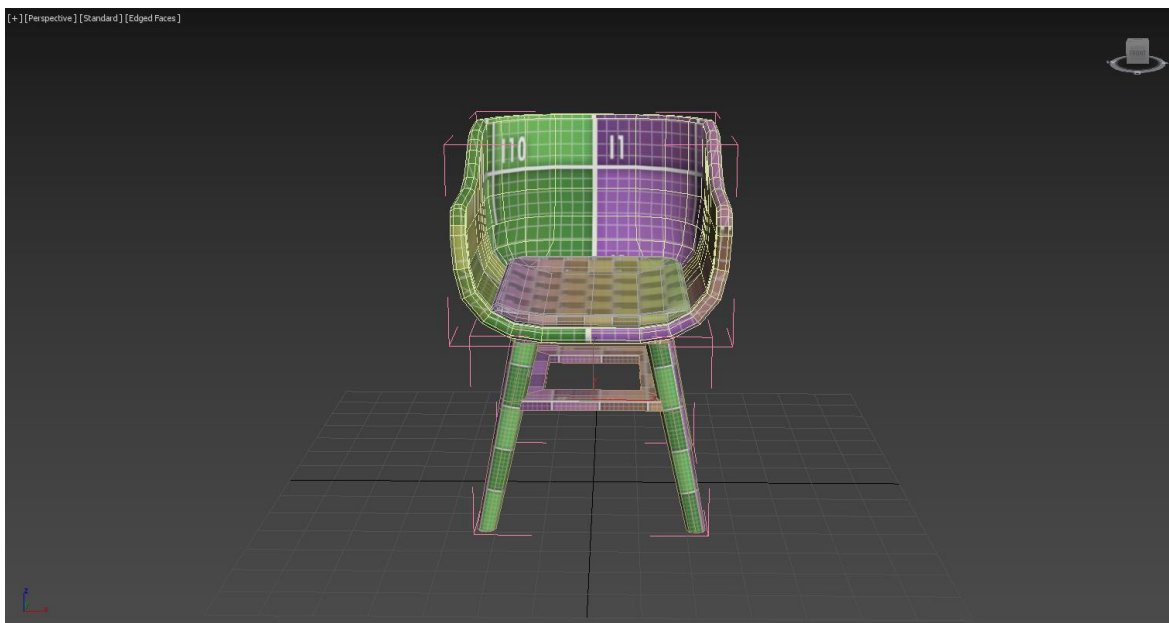
#### 3.1 Tekstuurikartoittaminen

3D-grafiikassa 3D-mallien pintojen ulkonäkö luodaan tekstuurikarttojen avulla. Tekstuurikartoilla pystytään määrittelemään pintojen yksityiskohtia, heijastumista ja väriä. Useiden eri tarkoituksiin erikoistuneiden tekstuurikarttojen yhdistelmällä voidaan saavuttaa lähelle fotorealistisia tuloksia. Jotta tekstuurikartat asettuisivat pinnoille oikealla tavalla, 3D-kappaleille täytyy suorittaa tekstuurikartoitus. Tekstuurikartoituksessa kappaleen pinnalle, eli meshille, määritellään tekstuurikoordinaatisto. Tekstuurikoordinaatiston avulla tekstuurikartat, eli tietyntyyppiset bittikarttakuvat asettuvat kappaleen pinnalle. Tekstuurikoordinaatistojen määrittelemiseen ja tekstuurikarttojen luomiseen tarkoitettuja ohjelmistoja on useita ja suurin osa 3D-mallinnusohjelmistoista sisältää sisäänrakennetun tekstuurikarttaeditorin. (Wikipedia 2019c.)



Kuva 5. UVW Map -modifikaattorilla tekstuurikartoitettu konferenssipöytä

3Ds Max -mallinnusohjelmiston sisältämällä UVW Map -modifikaattorilla voidaan määrittellä 3D-kappaleen tekstuurikoordinaatisto. Modifikaattorilla määritellään tekstuurikohtaiset U-, V- ja W-koordinaatit, joita voidaan ajatella yleisemmin kappaleen pinnan XYZ-koordinaatteina. UVW Map -modifikaattori sisältää seitsemän esiasetusta, jotka ovat eri tapoja määrittää kappaleen pinnan tekstuurikoordinaatiston muoto. Esiasetuksia ovat tason mukainen, sylinterin muotoinen, pallon muotoinen, pinnan mukainen kutistekalvo, laatikon muotoinen, kappaleen pinnan mukainen ja mallinnusympäristön koordinaatiston mukainen tekstuurikartoitus. Riippuen tekstuurikartoitettavan kappaleen muodosta, esiasetuksista etsitään kappaletta vastaava muoto tai sopiva tapa luoda tekstuurikartoitus. Esiasetuksien avulla luotu tekstuurikartoitus nähtävissä kuvassa 5. (Autodesk Inc. 2017b.)



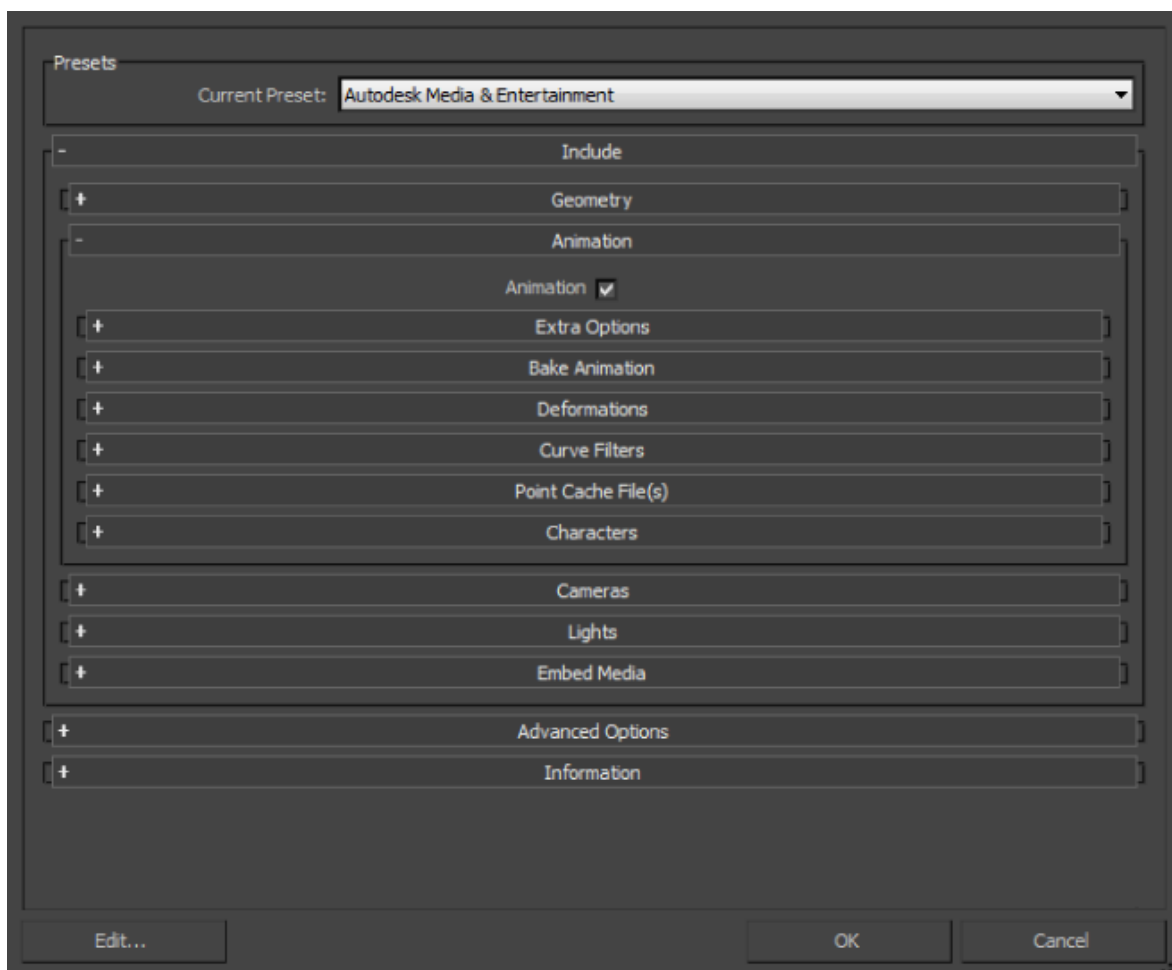
Kuva 6. Unwrap UVW -modifikaattorilla tekstuurikartoitettu tuoli

Monimuotoisemmissa 3D-kappaleissa, UVW Map -modifikaattorilla ei välttämättä pystytä saavuttamaan haluttua lopputulosta. Tilanteissa, joissa UVW Map -modifikaattori ei ole riittävä, käytetään Unwrap UVW -modifikaattoria. Unwrap UVW -modifikaattorilla saadaan luotua tekstuurikartoitusta monimuotoisille ja kaareville 3D-kappaleille. Unwrap UVW -modifikaattorilla pystytään valitsemaan ja jakamaan 3D-kappaleen pinnan osia sekä tekemään muutoksia pienemmillä osa-alueilla. Muutoksia 3D-kappaleen pinnan tekstuurikoordinaatistoon voidaan tehdä käsityönä ja/tai käyttämällä Unwrap UVW -modifikaattorin sisältämiä työkaluja. Kuvassa 6 on nähtävissä Unwrap UVW -modifikaattorin avulla kaareville muodoille toteutettu tekstuurikartoitus. Tehtäessä tekstuurikartoitusta Unwrap UVW -modifikaattorilla muokkauksia voi tehdä joko modifikaattori-näkymässä tai UV Editor -näkymässä, joka on 3Ds Max -mallinnusohjelmiston sisäänrakennettu tekstuurikarttaeditori. UV Editor -näkymässä näkyy uusi Unwrap UVW -modifikaattorilla luotu tekstuurikartoitus tai kappaleella ennestään olemassa oleva tekstuurikartoitus. (Autodesk Inc. 2017a.)

### 3.2 3D-mallien vienti ja tuonti

On olemassa useita eri tiedostoformaatteja, joilla mallinnusohjelmistot, simulaattorit ja pelimoottorit pystyvät käsittelemään 3D-malleja. 3D-malleja viedään ja tuodaan eri ohjelmistojen tukemiin tiedostoformaateihin, joilla muodostuu yhteiskytkettävyys eri ohjelmistojen välille. Ohjelmistojen vientiasetuksilla rajataan tiedostoformaattiin tallennettavaa tietoa 3D-mallista. Vastaavasti tuontiasetuksista pystytään rajaamaan vastaanotettavaa tietoa liittyen 3D-malliin. Noudattaen hyviä työtapoja, 3D-mallien vientiä ja tuontia ohjelmistojen vä-

lillä tulee myös optimoida. Vienti- ja tuontiasetuksilla eliminoidaan pois 3D-mallin tai työskentelytiedoston sisältämät turhat elementit ja eteenpäin viedään vain olennainen tieto. (Unity Technologies 2017a.)

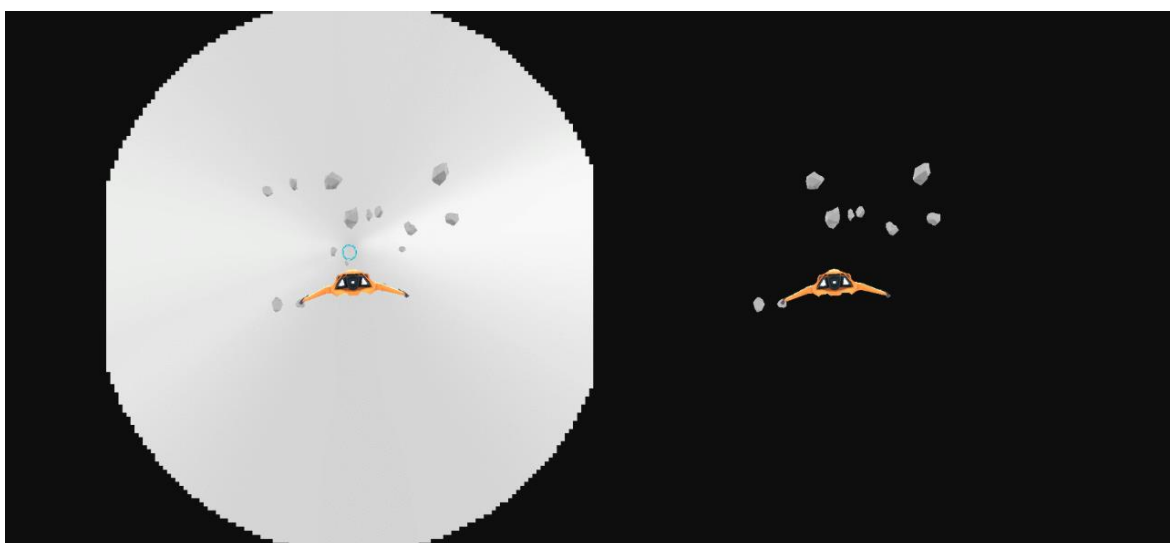


Kuva 7. 3Ds Max -mallinnusohjelmiston Export-valikko (Autodesk 2018)

3Ds Max -mallinnusohjelmisto tukee useita eri tiedostoformaatteja, joihin alkuperäisesti 3Ds Maxilla luodut, eli natiivit 3D-mallit on mahdollista kääntää. Yksi yleisimmistä 3D-tiedostoformaateista on FBX-tiedostoformaatti. FBX-tiedostoformaatti mainitaan ensisijaisena tiedostoformaattina Unity Technologiesin toimesta siirrettäessä 3D-malleja 3Ds Max -mallinnusohjelmiston ja Unity-pelimoottorin välillä (Unity Technologies 2019x). Viedessä 3D-malleja FBX-tiedostoformaattiin 3Ds Maxin vientivalikosta, eli Export-valikosta, joka on nähtävissä kuvassa 7, määritellään FBX-tiedostoksi tallennettavan 3D-mallin ominaisuudet. FBX-tiedostoformaatin sisältäviä ominaisuuksia ovat esimerkiksi geometriaan liittyvät asetuserittelyt, animaatiot, kamerat, valot, äännet, upotetut mediat, mittajärjestelmän skaala ja koordinaatiston orientaatio. (Unity Technologies 2017b.)

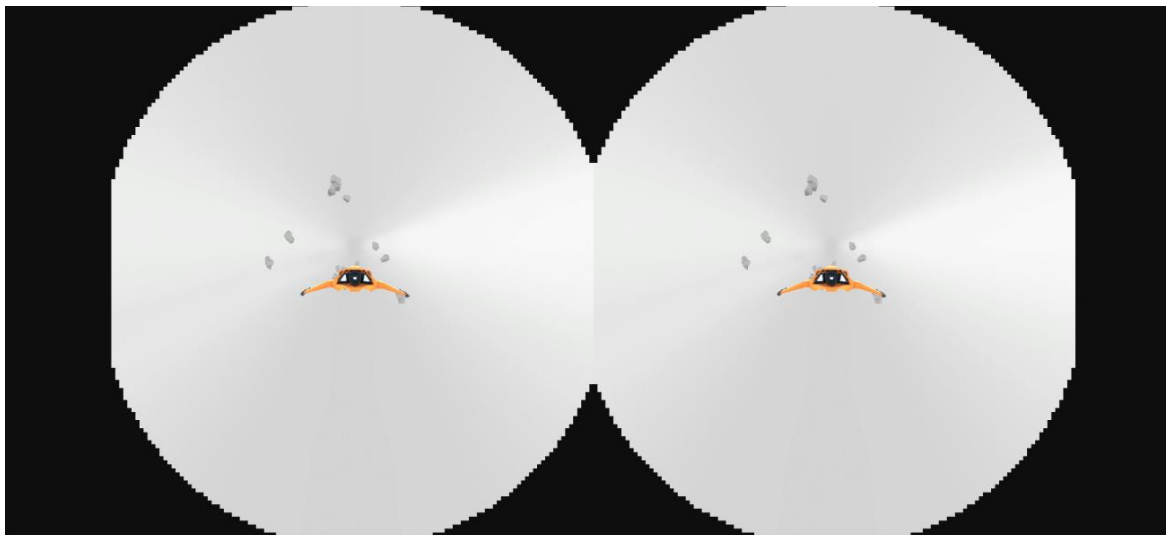
### 3.3 Renderöinti Unity-pelimoottorissa

Virtuaalitodellisuuden sovelluskehityksessä renderöinnin optimointi on yksi tärkeimpiä suorituskykyyn vaikuttavia tekijöitä. Unity-pelimoottorissa tapahtuvaan renderöintiprosessiin liittyy useita vaiheita ja prosessointitapoja, joiden asetuksia muokkaamalla vaikutetaan renderöidyn kuvan lopputulokseen. Oletusasetuksena kuva renderöidään peräkkäin virtuaalitodellisuuslasien molempien silmien näytöille yksi kerrallaan, mikä on esitettyinä kuvassa 8. Peräkkäinen renderöinti ei ole virtuaalitodellisuuden sovelluksissa optimaalinen vaihtoehto. Virtuaalisen ympäristön kaikki elementit joudutaan käsittelemään kahdesti renderöitäessä molempien silmien näytöille. Kaksinkertainen määrä renderöintejä rasittaa sovelluksen suoritettavuutta. (Unity Technologies 2019k.)



Kuva 8. Molempien silmien näytölle piirtyvän kuvan peräkkäinen renderöinti (Unity Technologies 2019)

Virtuaalitodellisuuslaseille renderöitävien kuvien renderöintitapaan pystytään vaikuttamaan Unity-pelimoottorin toistin-, eli player-asetuksista. Toistinasetuksista saadaan vaihdettua käyttöön virtuaalitodellisuuden sovelluskehittämiseen käytetty Single Pass Stereo -renderöinti. Single Pass Stereo -renderöintiasetuksella Unity-pelimoottori prosessoi renderöitävän ympäristön vain kerran kuvaa kohden. Renderöity kuva muodostetaan pala palalta yhtä aikaisesti molempien silmien näytöille, mikä on esitettyinä kuvassa 9. Renderöidyn kuvan korkeusresoluutio on sama kuin virtuaalitodellisuuslasien näyttöjen resoluutio, mutta kaksi kertaa leveämpi, eli sama, kuin molempien silmien näyttöjen leveysresoluutio yhteensä. (Unity Technologies 2019k.)



Kuva 9. Molempien silmien näytöille piirtyvän kuvan yhtäaikainen renderöinti (Unity Technologies 2019)

Kuvan tuottamiseksi virtuaalitodellisuuslasien näytöille Unity-pelimoottorin täytyy lähettää piirtopyyntöjä tietokonegrafiikan tuottamisesta vastaavalle rajapinnalle. Renderöinnin optimoinnissa rajapinnalle lähetettävien piirtopyyntöjen määrää pyritään vähentämään. Piirtopyyntöjen määrää vähennetään niputtamalla piirtopyynnöt renderöitävien elementtien yhteisillä tekijöillä. Piirtopyyntöjen niputtamista kutsutaan termillä Draw call batching. 3D-kappaleiden määrittelemine dynaamisiksi tai staattisiksi on tapa optimoida renderöintiä. 3D-kappaleet on oletusasetuksena määritetty liikkuviksi, eli dynaamisiksi objekteiksi. 3D-kappaleet, jotka eivät tule liikkumaan, määritellään liikkumattomiksi, eli staattisiksi objekteiksi. 3D-kappaleiden määrittäminen staattiseksi auttaa piirtopyyntöjen niputtamisessa ja on välttämätön asetus valaistuskarttojen esilaskennassa. Piirtopyyntöjä voidaan niputtaa myös käyttämällä samaa materiaalia eri 3D-malleissa. (Unity Technologies 2019c.)

Valaistuksen laskemiseen tarkoitetut renderöintipolut ovat yksi osa renderöintiprosessia. Valonlähteiden käsittelyyn ja laskentatapaan vaikutetaan valitsemalla renderöintipolku. Renderöintipolun valinta tehdään Unity-projektin hierarkiassa sijaitsevan kamerakomponentin asetuksista, joka on vastuussa kaikesta Unity-pelimoottorissa tapahtuvasta renderöinnistä. Kameran renderöintiasetuksista voidaan valita valaistuksen laskentatavaksi neljä eri renderöintipolkua. Vaihtoehtoja ovat Deferred shading-, Legacy Deferred Lighting-, Forward- ja Vertex Lit -renderöintipolut. Yksinkertaisin ja suorituskyvyn kannalta tehokkain tapa laskea valaistus on Vertex Lit -renderöintipolku. Vertex Lit -renderöintipolkua käyttämällä valaistus lasketaan vain kerran ja jokaista verteksiä kohti. Vertex Lit -renderöintipolku ei tue varjojen, tekstuurikarttojen tai korkean tarkkuuden valaistusta. (Unity Technologies 2019m.)

Luotettavaa ja ennustettavaa valaistusta haettaessa käytetään Deferred Shading- ja Legacy Deferred Lighting -renderöintipolkuja. Renderöintipolut ovat samalla tekniikalla toimivia valaistuslaskentatapoja, joista jälkimmäisenä mainittu on vanhentunut versio. Vanhemmasta Legacy Deferred Lighting -renderöintipolusta versiosta uupuu Unity-pelimoottorin sisältämien materiaalien Standard shader -pienoisohjelman tuki sekä mahdollisuus käyttää heijastusantureita. Uudemmassa Deferred Shading -renderöintipolussa edellä mainitut ominaisuudet ovat tuettuina. Deferred -renderöintipolut ovat toimivia ratkaisuja tekstuurikarttojen valaisemisessa ja varjojen luomisessa, mutta eivät sisällä tukea reunan pehmenyksen käytölle. (Unity Technologies 2019e.)

Valaistuksen laskemiseen virtuaalitodellisuuden sovelluksissa suositellaan käytettäväksi Forward Path -renderöintipolkua. Forward Path -renderöintipolulla ympäristön objektit käsitellään laskennallisesti vain kerran tai useammin riippuen niihin vaikuttavista valonlähteistä. Kirkkaimmat valonlähteet renderöidään pikselikohtaisina valoina. Jos kirkkaita valonlähteitä on enemmän kuin neljä, siirrytään verteksikohtaiseen valojen laskemiseen, mikä on kevyempi tapa prosessoida valaistus pikselikohtaiseen valojen laskemiseen verrattuna. Loput ympäristön valonlähteistä lasketaan arviollisena Spherical Harmonics -valaistuksena, joka on arviollisesti laskettu ja nopea prosessoida. Forward Rendering -renderöintipolulla saavutetaan osittain Vertex Lit -renderöintipolun tarjoamaa suorituskykyä ja osittain Deferred -renderöintipolkujen tarjoamaa interaktiota materiaalien ja varjojen kanssa. (Unity Technologies 2019d.)

3D-kappaleiden renderöinti on yksi suorituskykyyn vaikuttava tekijä. Suorituskyvyn laskeminen seuraa kuvaan renderöitävien polygonien määrästä. Ylimääräisten ja näkökentältä piilossa olevien polygonien renderöimistä kutsutaan yliiirroksi, eli Overdraw. Unity-pelimoottorissa 3D-kappaleille on mahdollista asettaa peittyvyys- ja peittävyysmäärittely. Peittyvyys- ja peittävyysmäärittelyä kutsutaan yhteisellä termillä Culling-määrittelyksi. Culling-määrittely auttaa vähentämään yliiirron lisäämää polygonimäärää. Virtuaalitodellisuuslasien näkökentästä jätetään renderöimättä kaikki polygonit, jotka peittyvät muiden polygonien taakse. Yliiirron aiheuttamaa rasitetta voi vähentää etukäteen poistamalla 3D-mal-leista ylimääräiset polygonit ja yksittäiset verteksit. (Unity Technologies 2019h.)

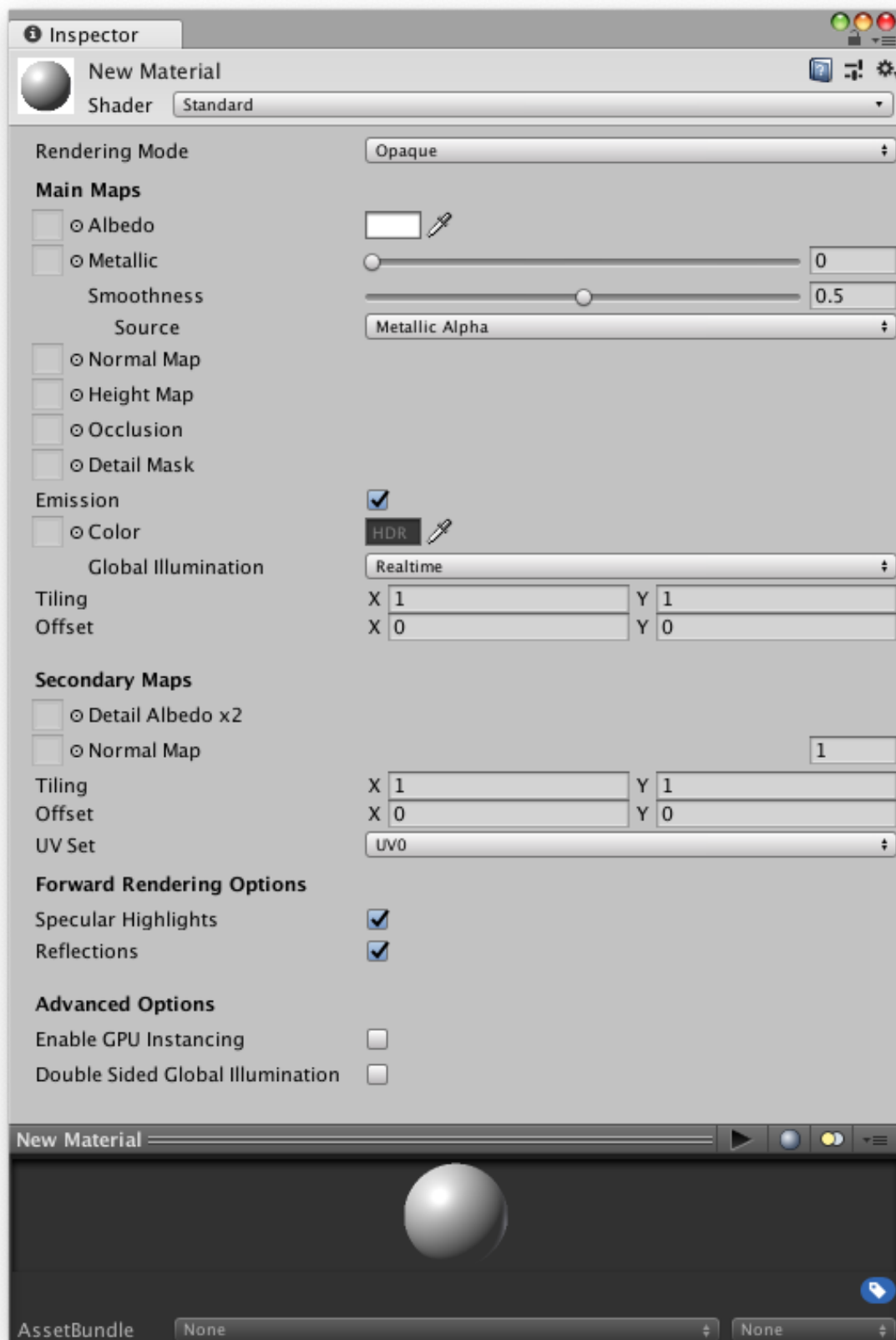
Määriteltäessä kohdekäyttölaitetta Unity-pelimoottorin renderöintiprosessin lopputulos määritellään laatu-, eli Quality-asetuksista. Laatuasetuksilla määritellään kaikkien sovelluksesta kehitettävien versioiden visuaalinen laatu. Samasta sovelluksesta olemassa oleville eri päätelaitteille voidaan määritellä erilaiset asetukset ottaen huomioon laitteiden spesifikaatiot. Laatuasetukset sisältävät useita muokattavia parametreja, jotka vaikuttavat renderöidyn kuvalaadun eri ominaisuuksiin. Laatuasetuksien renderöintiosiossa voidaan



säätää esimerkiksi valonlähteiden luoman valaistuksen tarkkuutta, materiaaleissa käytettyjen tekstuurien tarkkuutta sekä 3D-malleihin käytettävää reunan pehennystä. Varjot-osiossa voidaan tehdä muutoksia varjojen kulmaan, intensiivisyyteen, pehmeeyteen ja tarkkuuteen. Muut-osio sisältää sekalaisen kokoelman tekstuurien ja kuvataajuuden käsitteeseen liittyviä asetuksia. (Unity Technologies 2019i.)

### 3.4 Materiaalit Unity-pelimootorissa

Virtuaalisia ympäristöjä luodessa 3D-mallit ja pintojen ulkonäkö ovat avainasemassa visuaalisen vaikutelman luomisessa. Visuaalisen vaikutelman luomiseksi 3D-mallien pinnoille luodaan käytettäväksi materiaaleja. Materiaalit ovat Unity-pelimootorin komponenttiresursseja, joilla vaikutetaan Unity-projektin hierarkiassa sijaitsevien 3D-kappaleiden pintojen ulkonäköön. Materiaalia voidaan muokata joko valitsemalla se resurssivalikosta tai valitsemalla hierarkiasta 3D-kappale, johon materiaali on sidottu. Valittuna materiaalikomponentin parametrit näkyvät tarkastelu-, eli Inspector-näkymässä. Materiaalin parametreissa valitaan listasta haluttu shader-pienoisohjelma. Shader-pienoisohjelmia käytetään grafiikan laskemisen määrittelyyn. Shader-pienoisohjelmat sisältävät matemaattisia määrittelyitä ja algoritmin tavalle, jolla lasketaan värien piirtymistä kappaleiden pinnalle. Erilaisilla shader-pienoisohjelmilla saadaan aikaan erinäköisiä visuaalisia lopputuloksia. (Unity Technologies 2019f.)



Kuva 10. Materiaalikomponentin parametrinäkö (Unity Technologies 2019g)

Materiaalikomponentin shader-pienoisohjelmat sisältävät erilaisia parametreja riippuen, minkä tyyllisen materiaalin luomiseen shader-pienoisohjelma on tarkoitettu käytettäväksi. Parametrien arvoja muokataan antamalla numeraalisia arvoja tai käyttämällä liukusäädintä numeraaliarvon muuttamiseen. Parametrin arvon määrittelyyn voidaan käyttää myös tekstuurikarttoja. Materiaalin säädettävät parametrit ovat nähtävissä kuvassa 10. Materiaalien eri parametreille olemassa olevia tekstuurikarttoja on useita, jotka on tarkoitettu asetettavaksi yhteensopiville parametrille. Tekstuurikartat ovat erinäköisiä bittikarttakuvia, jotka sisältävät kuvioita ja väriarvoja määrittäen luotavan materiaalin ulkonäön. Tekstuurikartoilla vaikutetaan materiaalin väräihin, kuvioihin, rosoisuuteen, heijastuksiin ja muihin valon käyttäytymiseen liittyviin tekijöihin. (Unity Technologies 2019f.)



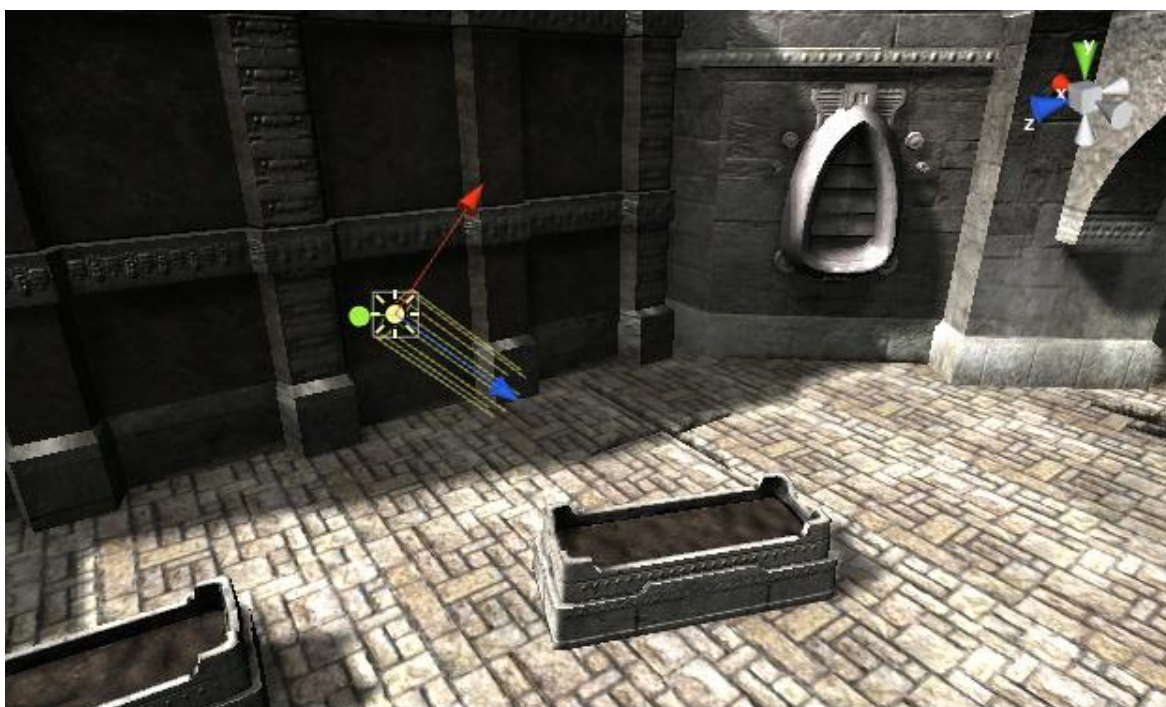
Kuva 11. Esimerkki cubemap-, 360 asteen-, HDRI Map -kuvasta (Aversis 3D 2019)

Jotta materiaalien esittämät heijastukset vastaisivat ympärillä olevaa virtuaalista ympäristöä, käytetään apuna heijastusantureita. Heijastusanturi, eli Reflection Probe, on Unity-pelimootorissa oleva objekti, joka tutkailee ympäristöä kameran kaltaisesti. Heijastusanturin parametreissa määritellään tarkasteltavan alueen koko. Yleisesti, tarkkailtavan alueen koko määritellään huoneen kokoiseksi. Heijastusanturin sijainti on määritellyn tarkkailualueen keskikohdassa. Kun heijastusanturi on lisätty Unity-projektiin, tarkkailualueen sisällä olevat materiaalit kommunikoivat heijastusanturin luomalla tiedolla. Materiaalit peilaavat ympäristöä heijastusanturin luoman heijastuskuvan avulla. Heijastuskuvan muoto on kuutiobittikarttakuva, eli cubemap-bittikarttakuva, joka on esitettyä kuvassa 11. Kuutiobittikartastakuvasta puhutaan toisilla nimillä 360-asteen kuvana tai High Dynamic Range Imaging Map (HDRI Map) -kuvana. Unity-pelimootori pystyy käsittelemään heijastuksia sekä reaaliaikaisesti että esilasketusti. Optimoinnin kannalta voidaan tehdä tarpeenmukainen valinta esilasketun ja reaaliaikaisen heijastumisen välillä. (Unity Technologies 2019j.)

### 3.5 Valaistus Unity-pelimoottorissa

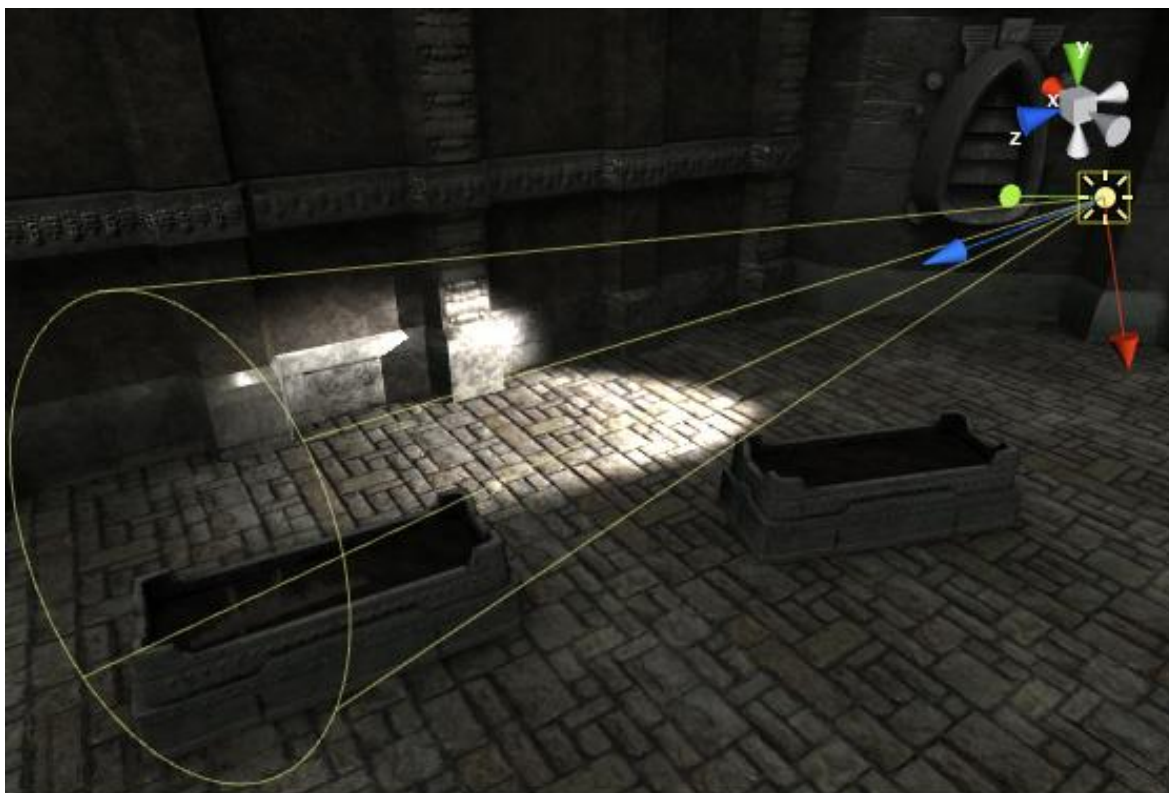
Valaistuksen laskeminen Unity-pelimoottorissa tapahtuu kahden eri toteutustavan avulla. Valaistus lasketaan joko reaaliaikaisesti sovelluksen ollessa käynnissä, tai valaistus on luotu esilaskettujen valaistuskarttojen avulla. Reaaliaikaisen valaistuksen hyödyt tulevat esille dynaamisen geometrian, kuten liikkuvien objektien ja hahmojen valaisemissa. Reaaliaikainen valaistus päivittyy kuvataajuuden mukana jokaiselle renderöitävälle kuvalle. Valaistuksen laskeminen jokaiselle renderöidylle kuvalle vaikuttaa sovelluksen suoritettavuuteen, eikä näin ollen ole optimaalinen vaihtoehto virtuaalitodellisuuden sovelluskehittämisessä. (Unity Technologies 2019b.)

Optimaalisempi vaihtoehto virtuaalitodellisuuden sovelluskehittämisessä on valaistuksen esilaskenta, eli light baking. Esilaskettu valaistus soveltuu liikkumattomien, eli staattisten 3D-mallien, kuten ympäristön valaisemiseen. Valaistuksen esilaskennassa valonlähteiden tuottamasta valosta ja 3D-kappaleiden luomista varjoista piirretään valaistuskarttakuvia. Valaistuskarttakuvat kiinnitetään liikkumattomien 3D-kappaleiden pinnalle. Esilasketun valaistuksen etuna on alhaisempi rasitus suorituskyvylle ja sisävalaistuksessa saavutettu realistisempi lopputulos verrattuna reaaliaikaiseen valaistukseen. Realistisempi sisävalaistus syntyy pinnoilla kimpoilevien fotonien takia, mikä ei ole mahdollista reaaliaikaisella valaistuksella. Molempia valaistustapoja on mahdollista käyttää yhdistetysti tai yksinään. Yhdistetyllä valaistustavalla pystytään eliminoimaan molempien valaistustapojen heikkoudet. (Unity Technologies 2019a.)



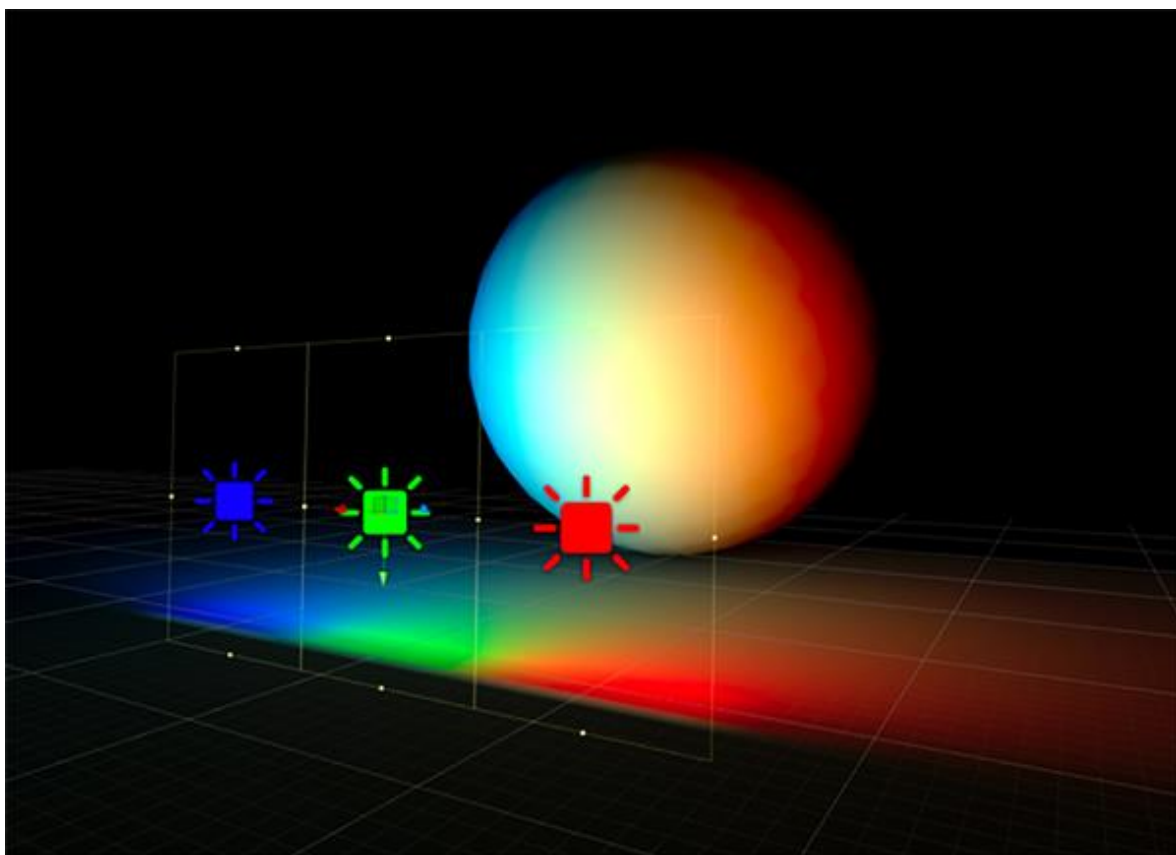
Kuva 12. Directional Light -valonlähde (Unity Technologies 2019)

Valaistustavasta riippuen, Unity-pelimoottorissa on käytettävänä kuusi erilaista valonlähdettä. Valonlähteet sisältävät toisistaan erilaisia parametreja ja soveltuvat erilaisiin käyttötarkoituksiin. Neljä Unity-pelimoottorin kuudesta valonlähteestä on hierarkiaan lisättäviä valonlähdeobjekteja. Ambient Light -valonlähdettä ei lasketa valonlähdeobjektiksi, vaan on abstrakti valonlähde ja läsnä jokaisessa luodussa Unity-projektissa. Ambient Light -valonlähteen asetuksia muokataan valaistus- eli Lighting-näkymässä. Ambient Light -valonlähteen asetuksilla pystytään vaikuttamaan koko projektiympäristön valaistuksen luomaan tunnelmaan ja taiteelliseen tyyliinsuuntaan. Ambient Light -valonlähteen lisäksi toinen ympäristön valaistuksen tunnelmaan vaikuttava valonlähde on Directional Light -valonlähde. Directional Light -valonlähteen käyttö projektissa ei ole välttämätöntä, mutta löytyy oletuksena jokaisen Unity-projektin hierarkiasta. Directional Light -valonlähdettä käytetään simuloimaan auringon valon tuottamaa valaistusta. Nimensä mukaisesti Directional Light -valonlähde suuntaa valoa määrätystä suunnasta määrättyyn suuntaan kuvan 12 esittämällä tavalla. Directional Light -valonlähteen sijainnilla ei ole vaikutusta valon käyttäytymiseen, mutta valon tulokulmalla on. Directional Light- ja Ambient Light -valonlähteitä käytetään yleensä virtuaalisten ympäristöjen ulkovalaistuksessa. (Unity Technologies 2019l.)



Kuva 13. Spot Light -valonlähde (Unity Technologies 2019)

Virtuaalisen sisätilan valaistuksessa voidaan käyttää hyödyksi neljää yleisemmin sisätila-valaistukselle soveltuvaa valonlähdetä. Valonlähteiden tuottaman valon muoto ja intensiivisyys vaihtelevat valonlähteestä riippuen. Point Light -valonlähteellä saadaan tuotettua pallon muotoista valaistusta asetetusta sijainnista jokaiseen suuntaan. Tuotetun valon piirteisiin vaikutetaan muokkaamalla säde- ja voimakkuusparametrien arvoja. Point Light -valonlähde soveltuu käytettäväksi lamppujen tai erikoistehosteiden, kuten kipinöiden ja räjähdysten valonlähteenä. Vastakohtaisesti Point Light -valonlähteen sijasta, suunnattujen valojen luomiseen käytetään Spot Light -valonlähdetä. Spot Light -valonlähde luo kartion muotoisen valokeilan, joka on nähtävissä kuvassa 13. Valokeilalla saadaan luotua tunnelmaa painottamalla ympäristöstä olennaisia kohteita. Spot Light -valonlähde soveltuu myös eri laitteiden luomiin valaistuksiin. Valovoimakkuus vähenee Point Light- ja Spot Light -valonlähteissä käänteisessä suhteessa etäisyyden neliöön nähden. (Unity Technologies 2019k.)



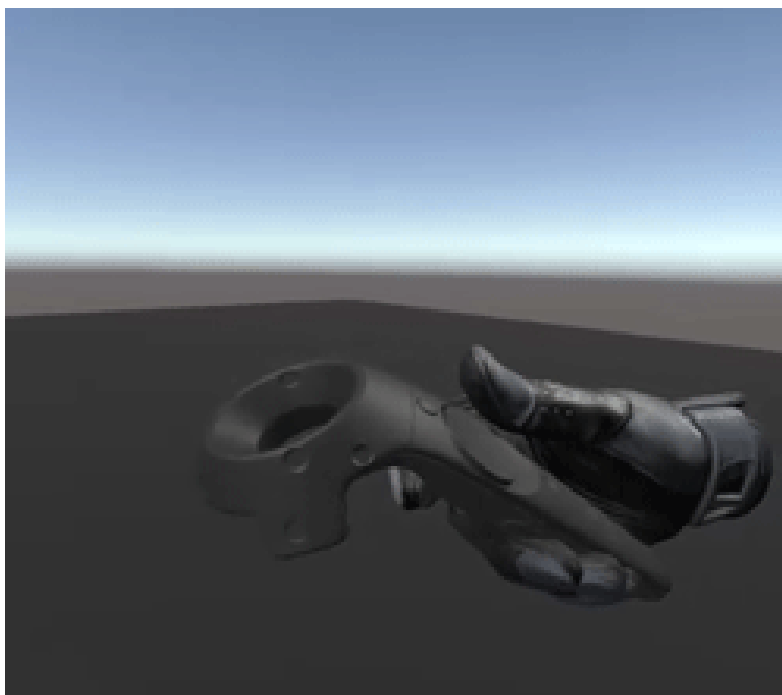
Kuva 14. Eri värisiä Area Light -valonlähteitä (Unity Technologies 2019)

Tasaiseen tilojen valaisemiseen soveltuva Area Light -valonlähde on valaistustavaltaan poikkeava verrattuna Unity-pelimoottorin muihin valonlähteisiin. Area Light -valonlähde tuottaa valoa tasaisesti suorakulmaisen muotoisella valonlähteellä kuvassa 14 nähtävällä

tavalla, mikä tekee valonlähteestä laskennallisesti resurssi-intensiivisen. Resurssi-intensiivisyyden takia Area Light -valonlähde on mahdollista käyttää eksklusiivisesti vain esilasketussa valaistuksessa. Toisella tavalla muista valonlähteistä poikkeava valonlähde on Unity-pelimoottorin materiaalien mahdollisuus valon tuottamiseen, eli emissioon. Materiaalien parametreista pystytään määrittelemään materiaalin tuottaman valaistuksen voimakkuutta ja muuttamaan materiaalin tuottaman valaistuksen väriä. (Unity Technologies 2019k)

### 3.6 SteamVR- ja Virtual Reality Toolkit -liitännäiset

SteamVR on Valve Corporation -yhtiön kehittämä ilmainen liitännäinen Unity-pelimoottorille. Liitännäinen toimii yhteensopivana rajapintana usean eri valmistajan virtuaaliodellisuuslaitteiden ja Unity-pelimoottorin välillä. Liitännäisen avulla Unity-pelimoottori kykenee visualisoimaan 3D-mallit virtuaaliodellisuuden ohjaimista sekä käyttäjän käsistä ohjainten ympärillä. 3D-mallit liikkuvat virtuaalisessa ympäristössä reaaliaikaisesti ohjainten sijaintiin nähden. Ohjainten 3D-mallin muoto määräytyy tunnistetun laitevalmistajan mukaan. Kuvasessa 15 on nähtävissä virtuaalisessa ympäristössä simuloitu HTC Vive -virtuaaliodellisuusohjain sekä käyttäjän käsi. Sormien liikkeitä voidaan arvioida ja esittää virtuaalisessa tilassa nyrkin ja avokämmenen välillä. Liitännäisen avulla ohjaimiin voidaan ohjelmoida eri toiminnollisuuksia ja ohjainten painikkeiden käyttäytymistä voidaan muokata eri tavoin. (Valve Corporation 2019.)



Kuva 15. HTC Vive -ohjaimen ja käyttäjän käden 3D-esitys virtuaalimaailmassa (Joe van den Heuvel 2018)

Virtual Reality Toolkit, eli VRTK, on Sysdia Solutions LTD -yrityksen kehittämä ilmainen liitännäinen Unity-pelimootorille. Liitännäisen sisältää valmiiksi ohjelmoituja toiminallisia kokonaisuuksia. Valmiiksi ohjelmoituja toiminnallisia kokonaisuuksia ovat esimerkiksi laserosoitin, käyttöliittymän kanssa toimiva osoitin, yleiset interaktiot mekaanisten 3D-objektien kanssa, teleportaatio ja haptinen palaute. Käyttämällä Virtual Reality Toolkit -liitännäisen valmiita toimintoja, virtuaalitodellisuuden sovelluksien rakentaminen ja kehittämiseen vaadittu tietotaitokynnys laskee. Kehittäminen Virtual Reality Toolkit -liitännäisellä ei vaadi VR-laitteiden omistamista, koska liitännäinen sisältää virtuaalitodellisuussimulaattorin. (Sysdia Solutions LTD. 2019.)



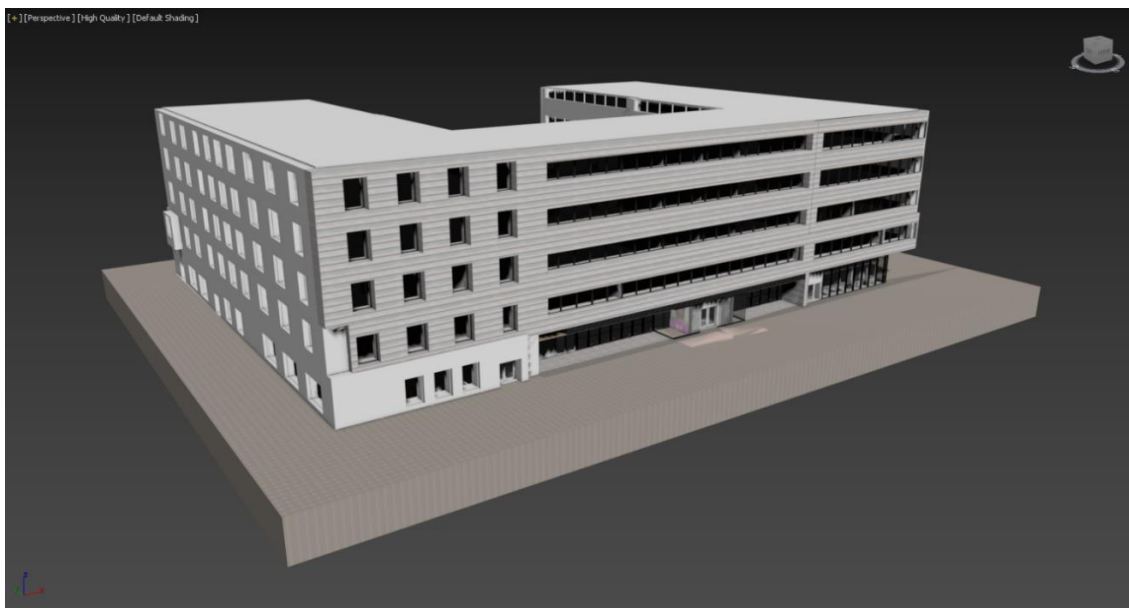
## 4 CASE: VIRTUAALINEN PANKKISALI

Tämän opinnäytetyön case-osuus suoritettiin osana työharjoittelua lahtelaisessa yrityksessä nimeltä Revulon. Työn tilaajana oli toimitusjohtaja Veli-Pekka Puolakka Torikapitaali Oy:stä, joka hallinnoi Alekski 10 -liikekiinteistöä Lahden kaupungin keskustassa. Esiohjeistus työhön tapahtui Revulonin toimitusjohtajan, Jukka Saarion toimesta. Suunnitelma työvaiheista ja lopputuotteesta tehtiin yhdessä Puolakan ja Saarion kanssa. Tuotteen kehittämiseen käytettiin 3Ds Max -mallinnusohjelmistoa ja Unity-pelimoottoria. Tuotteen tarkasteluun ja kehittämisen apuna käytettiin HTC Vive -virtuaalitodellisuuslaitteita.

Lähtökohtana oli luoda virtuaalinen tila kiinteistössä sijaitsevasta vanhasta pankkisalista. Projektin ensimmäiseksi työvaiheeksi suunniteltiin virtuaalisen todellisuuden laitteille optimoitu tyhjä tila. Tilassa olisi mahdollista katsella ympärille ja liikkua ohjaimien ohjelmoidulla teleporttitoiminnolla. Toisessa työvaiheessa pankkisaliin lisättäisiin toimistosisustus. Toimistosisustuksen toteutukseen kuuluisi huonekalujen mallintaminen, paikalleen järjestäminen sekä materiaalien asettelu. Suunnitteluvaiheessa keskusteltiin myös pankkisalin muista sisustusratkaisuista, joita harkittaisiin toisen työvaiheen valmistumisen jälkeen.

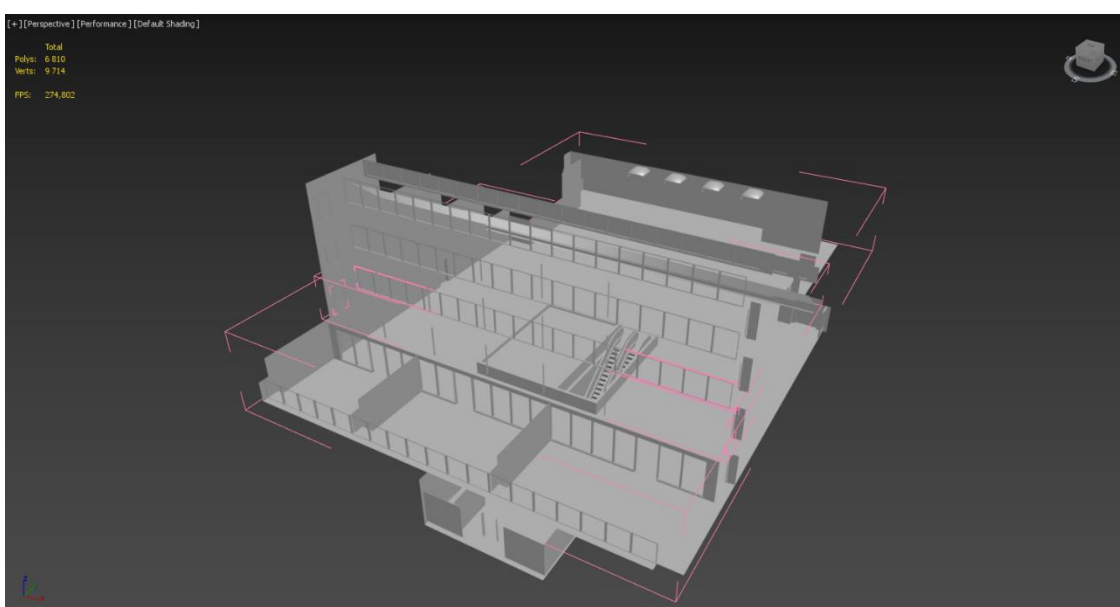
### 4.1 Pankkisalin 3D-mallin optimointi ja tekstuurikartoitus

3D-malleihin liittyvä työskentely suoritettiin 3Ds Max -mallinnusohjelmistolla laajimman kokemuksen takia. Työn tilaaja Veli-Pekka Puolakka toimitti kiinteistöstä ennestään olemassa olevan 3D-mallin, joka on nähtävissä kuvassa 16. Kiinteistön 3D-malli oli alkuperäisesti, eli natiivisti luotu 3Ds Max -mallinnusohjelmistoa käyttäen. Toimitetusta 3D-mallista karsittiin kiinteistön muut tilat pankkisalia lukuun ottamatta. Pankkisalin koostavat 3D-kappaleet käytiin läpi, korjattiin ja optimoitiin yksi kerrallaan. Korjattavat kappaleet sisälsivät muutamia verteksejä, jotka olivat irrallaan kappaleen muista osista sekä joidenkin verteksien väliltä puuttui reunayhteyksiä. Yksittäiset yhteydettömät verteksit poistettiin. Kappaleiden loogista pinnan rakennetta korjattiin lisäämällä reunayhteyksiä verteksien välille sekä yhdistämällä verteksejä kiinni toisiinsa 3D-kappaleiden yhtenäistämiseksi. Valtaosa pankkisalin rakenteista muokattiin olemassa olevasta mallista, mutta osa mallinnettiin alusta alkaen ajankäytön ja työnkulun edesauttamiseksi. Näitä alusta alkaen mallinnettuja kappaleita olivat pääasiassa seinät, lattiat sekä osa pankkisalin kattoelementeistä.



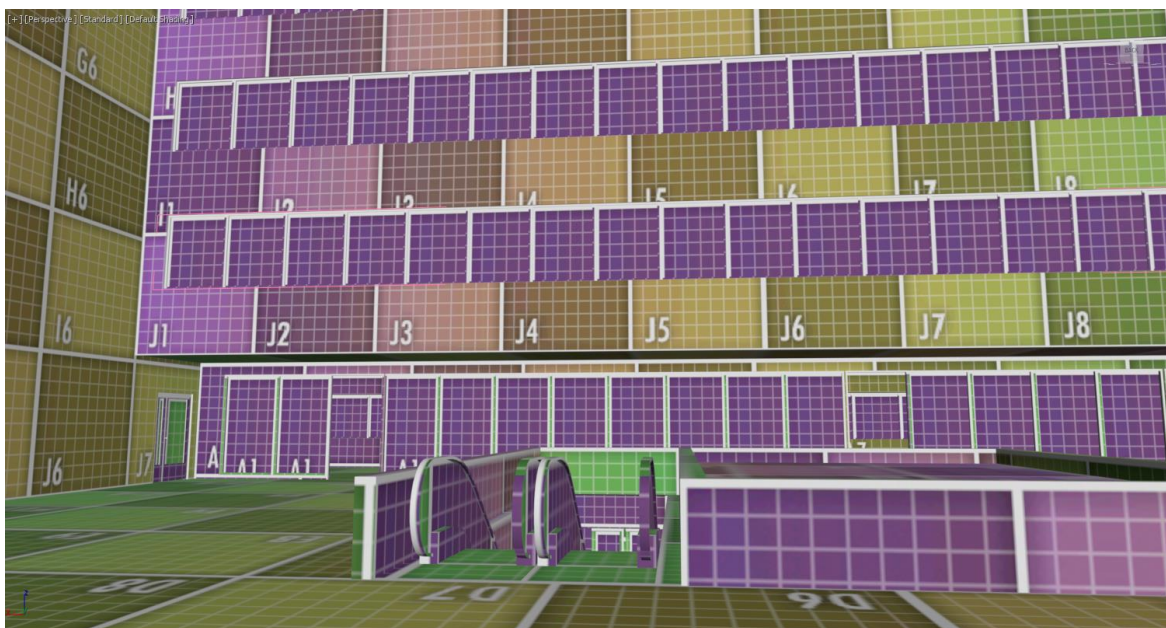
Kuva 16. Tilaajalta saatu Aleksio10 -liikekiinteistön 3D-malli

Mallinnettaessa ja optimoinnissa panostettiin kappaleiden minimaaliseen polygoni määrään sekä yliiirron minimointiin. Ennakoiden yliiirron minimointia, lähes kaikki rakenteelliset 3D-kappaleet muokattiin tai mallinnettiin yksipuolisiksi, minkä tulos on näkyvillä kuvassa 17. Kappaleiden korjaamisella ja optimoinnilla haluttiin välttää mahdollisia Unity-pelimoottorissa ilmeneviä visuaalisia virheitä, sekä vähentää sovelluksen suorituskykyä rasittavia tekijöitä. Visuaalisia virheitä olisi mahdollisesti ilmennyt materiaalien asettelussa 3D-kappaleiden pinnoille ja valaistuskarttakuvien laskemisessa. Suorituskykyyn vaikuttava tekijä olisi ollut virtuaalitodellisuuslasien näkökentässä yliiirrosta aiheutunut renderöitävien polygonien määrän nousu.



Kuva 17. Optimoitu pankkisali yksipuolisilla 3D-pinnoilla

Pankkisalın tekstuurikartoitukseen käytettiin 3Ds Max -mallinnusohjelmiston omia työkaluja. Tilan yksinkertaisten muotojen vuoksi kaikkiin pankkisalın rakenteita koskeviin 3D-kappaleisiin luotiin tekstuurikartoitus käyttämällä UVW Map -modifikaattoria. Rakenteellisten 3D-kappaleiden, kuten seinien, lattioiden ja kattoelementtien tekstuurikoordinaatistojen luonnissa käytettiin UVW Map -modifikaattorin laatikko- tai taso-esiasetusta. Laatikko-esiasetuksella luotiin tekstuurikartoitus kaikille kappaleet, jotka sisälsivät yhdeksänkymmenen asteen kulmia polygonien välillä. Taso-esiasetusta käytettiin yksipuolisissa pinoissa, kuten seinissä, lattioissa ja kattoelementeissä. Pankkisalın 3D-mallille luodun tekstuurikartoituksen tulos nähtävissä kuvassa 18.



Kuva 18. Pankkisalın tekstuurikartoitettu 3D-malli sisältä katsottuna

#### 4.2 Pankkisalın 3D-mallin tuonti ja vienti

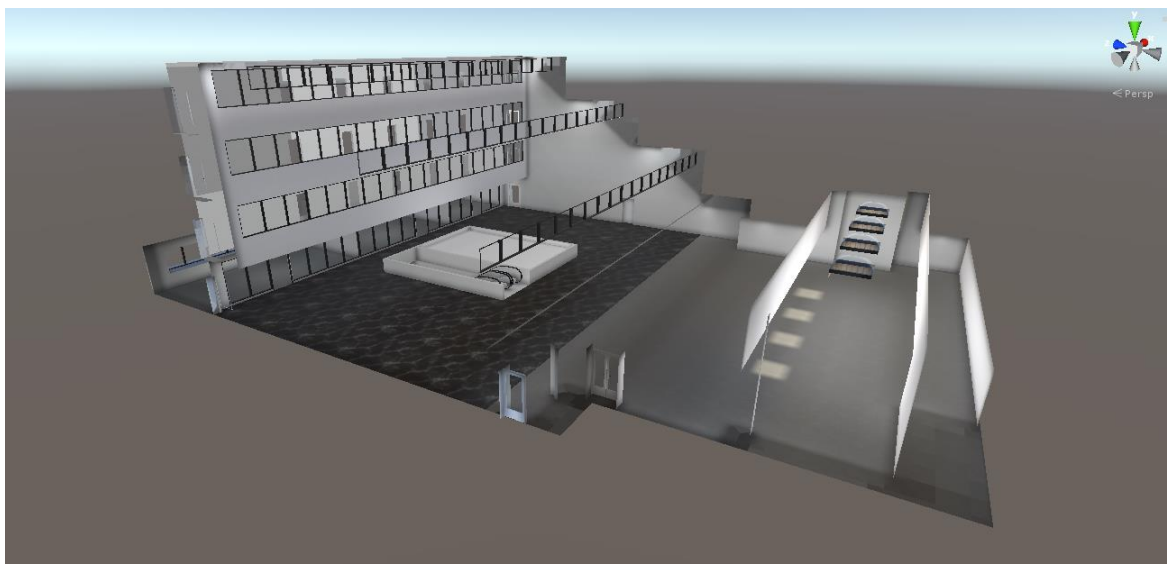
3D-mallin optimoinnin ja tekstuurikartoitus-määrittelyiden jälkeen virtuaaliodellisuussovellusta aloitettiin kehittämään Unity-pelimoottorilla. 3D-mallit käännettiin FBX-tiedostoformaattiin Unity-pelimoottoria varten. Tiedostoformaatin kääntäminen tapahtui 3Ds Max -mallinnusohjelmiston sisäänrakennetulla liitännäisellä. Toinen mahdollinen vaihtoehto olisi ollut OBJ-tiedostoformaatti, mutta henkilökohtainen kokemus oli osoittanut FBX-tiedostoformaatin paremmaksi vaihtoehdoksi. Vientiasetukset, eli Export-asetukset käytiin läpi ennen kääntämistä. Asetuksista jätettiin pois tarpeettomat ominaisuudet, kuten animaatioiden vienti, kameroiden vienti, valaistuksen vienti sekä materiaalien vienti. Vientiasetuksista kokoluokan ja orientaation asetukset säilytettiin oletuksena sekä kääntämiseen käytettävän liitännäisen versio. Vientiasetusten ollessa kunnossa pankkisalın 3D-malli käännettiin ja tallennettiin FBX-tiedostoformaattiin.

FBX-tiedostoformaattissa oleva 3D-malli tuotiin Unity-projektin resursseihin. Unity -pelimoottorin resurssinäkyvässä määriteltiin 3D-malliin liittyvät tuontiasetukset. Tuontiasetuksista määriteltiin pois tarpeettomat ominaisuudet, kuten valot, kamerat, materiaalit ja animaatiot. Tuontiasetuksista valittiin päälle Generate Lightmap UV's -asetus, jotta kappaleille esilaskettavat valaistuskartat tulisivat renderöitymään ja asettumaan oikealla tavalla. Swap UV's -asetus valittiin päälle, jotta 3Ds Max -mallinnusohjelmistossa luodut tekstuurikoordinaatit olisivat käytössä ja materiaalit asettuisivat 3D-kappaleiden pinnoille oikein. Loput tuontiasetuksista säilytettiin oletusasetuksissa ja asetusten muutokset tallennettiin. Asetusmäärittelyiden jälkeen 3D-malli lisättiin Unity-projektinäkyvän hierarkiaan.

### 4.3 Materiaalien ja valaistuksen luonti

Kun 3D-malli oli asetettu paikoilleen Unity-pelimoottorissa, kappaleille aloitettiin lisäämään materiaaleja. Materiaalit luotiin käyttäen referenssinä osittain reaali maailman tilan ulkonäköä, käyttäen apuna kiinteistöstä olemassa olleen 3D-mallin materiaaleja sekä suunniteltaessa keskustellun "demo"-ulkonäön perusteella. Materiaalien määrä pyrittiin pitämään minimissä ennakoiden toisen työosuuden lisäävän materiaalien määrää. Ensimmäiseen työosuuteen luotiin käytettäväksi neljätoista eri materiaalia. Kaikkiin materiaaleihin valittiin Standard (Roughness Setup) -shader, joka oli sopivin vaihtoehto valituille tekstuuriresursseille. Materiaaleihin kuului tekstuurikartoilla luotuja pintoja, kuten mustaa marmoria, rappattua valkoista seinää, kumia, lasia, puuta, sekä erilaisia metalli- ja betonipintoja.

Materiaalien luomisen ja asettelun jälkeen pankkisaliin lisättiin heijastusantureita. Heijastusantureita aseteltiin muutamiin kohtiin tyhjää pankkisalia testiksi tietäen, että niiden paikka ja määrä tullaan muuttamaan toisessa työosuudessa tilan muuttuessa. Heijastusantureiden asettamisen jälkeen pankkisaliin lisättiin valonlähteet. Sovelluksessa ei käytetä reaaliaikaista valaistusta, joten valaistukseen käytettiin esilasketulle valaistukselle sopivaa Area Light -valonlähdettä. Area Light -valonlähde valittiin suorakulmion muotoisen ja tasaisesti valaisevan valonlähteen takia. Virtuaalisen ympäristön sisävalot aseteltiin reaali maailman pankkisalia vastaaville paikoille. Ulkovalaistuksen luoma tunnelma toteutettiin Ambient Light- ja Directional Light -valonlähteiden asetuserittelyiden yhteisvaikutuksella.



Kuva 19. Pankkisali Unity-pelimoottorissa valaistuskarttojen laskemisen jälkeen

Ennen valaistuskartan laskemista Unity-projektin sisältämä 3D-malli määriteltiin liikkumattomaksi objektiksi, eli staattiseksi. Staattisuus-määrittelyllä mahdollistettiin valaistuskartan laskeminen 3D-mallin pinnoille. Valaistuskartta laskettiin useita kertoja uudestaan iteroiden pankkisalin visuaalista ulkonäköä. Valaistuskartan laskuasetusten parametreilla haettiin riittävää laatua laskennalliseen aikaan nähden. Laskettujen valaistuskarttojen tuloksia tarkasteltiin VR-laseilla laskentakertojen välillä. Materiaalien ja valaistuskartan laskemisasetuksien parametreja muokattiin tarvittavaan suuntaan. Ensimmäisen työsuuden päätteeksi tilalle laskettu valaistuskartta nähtävissä kuvassa 19. Valaistuskarttojen iteratiivisen laskemisen lopputuloksena saavutettiin paremmat materiaaliopinnot ja valoisaampi tila.

#### 4.4 Toiminnollisuuksien toteuttaminen

Suunniteltaessa virtuaalisen sovelluksen toiminnollisuuksia päätettiin, että ainut tarvittava toiminnallinen ominaisuus olisi teleporttitoiminto. Visuaalisuuteen liittyvien työvaiheiden ollessa valmiita asennettiin virtuaalitodellisuuden toiminnollisuudelle ja laitteiston käytölle olennaiset liitännäiset. SteamVR ja Virtual Reality Toolkit -liitännäiset asennettiin Unity-projektin resursseihin. Unity-projektiin luotiin pelaajaobjekti, joka sisälsi resursseja molemmista liitännäisistä. Molempien liitännäisten resursseja yhdistelemällä luotu pelaajaobjekti mahdollisti virtuaalitodellisuudessa näkyvät ohjaimet ja kädet sekä suunnitellun teleporttisiirtymisen toiminnollisuuden. Teleporttitoiminnollisuus toteutettiin käyttämällä Virtual Reality Toolkit -liitännäisen sisältämiä resursseja. Teleporttitoiminnollisuus olisi voitu toteuttaa käyttämällä pelkästään SteamVR -liitännäisen resursseja, mutta Virtual Reality Toolkit -liitännäisellä toteuttaminen todettiin paremmaksi vaihtoehdoksi. Valinnalla ennakoitiin tar-

vittaessa tulevaisuudessa sovellukseen lisättäviä käyttöliittymä-toiminnollisuuksia. Toiseen virtuaalitodellisuusohjaimista luotiin laserosoitin ja pankkisalin 3D-kappaleille määritettiin kohtaamistarkastelu-, eli Collider-komponentti. Osoittamalla ja painamalla virtuaalitodellisuusohjaimen Trackpad-nuolinäppäintä pystyttiin teleporttaamaan paikasta toiseen. Kohtaamistarkastelukomponenteilla rajattiin liikkumista pankkisalin sisällä. Kohtaamistarkastelukomponenttien avulla voitiin määritellä pinnat, joille teleporttitoiminnolla siirtyminen oli sallittua. Osoittaessa ohjaimen laser-osoittimella seinille tai kattoon, laserosoittimen väri muuttui punaiseksi. Laserosoittimen osoittaessa lattiapinnoille, laserosoittimen väri oli vihreä.

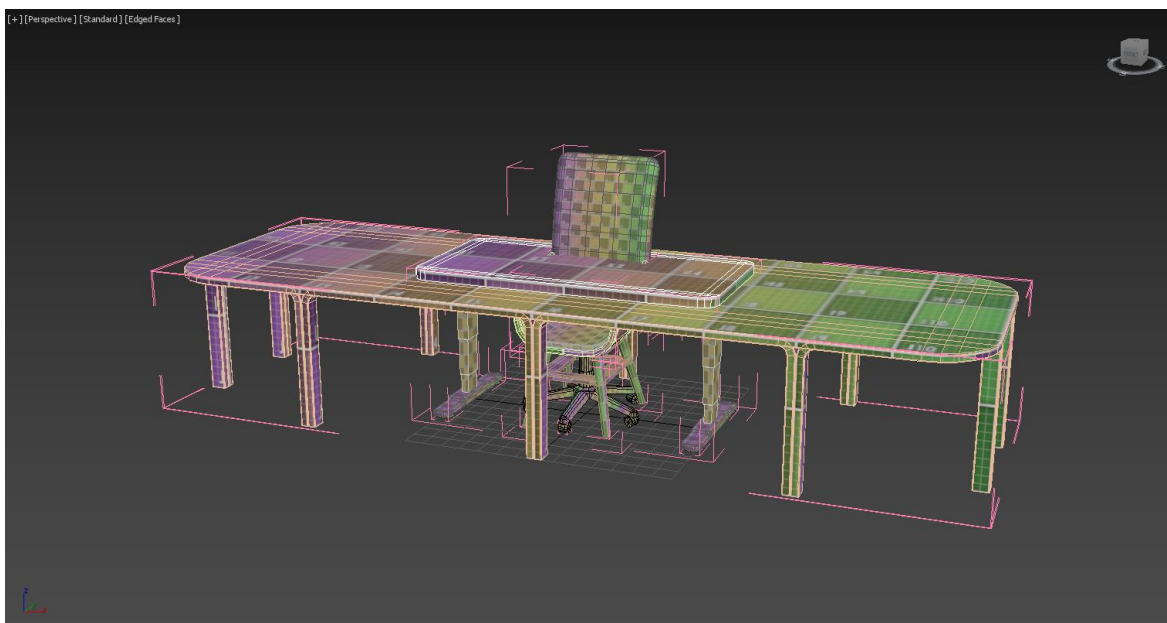
#### 4.5 Ensimmäinen koonti

Ensimmäisen työvaiheen valmistuessa Unity-projekti koostettiin yhtenäiseksi sovellukseksi, eli buildattiin. Sovellus toimi moitteetta. Virtuaalisessa tilassa pystyttiin liikkumaan määrättyjen rajojen sisällä, eikä tilasta pystynyt siirtymään ulos. Laserosoittimen väri näkyi vihreänä ja punaisena määrättyllä tavalla. Pidimme palaverin yhdessä Veli-Pekka Puolakan ja Jukka Saarion kanssa. Ensimmäisen työosuuden tuottama tuote esiteltiin ja tilaaja pääsi kokeilemaan virtuaalista pankkisalia. Huomasimme heittoa pankkisalin mittasuhteissa. Virtuaalisen tilan neljännessä kerroksessa oleva välimatka sisäseinästä ulkoikkunoihin oli liian pieni verrattuna todelliseen tilaan. Todellisuudessa välimatka oli muutamia metrejä suurempi. Ulkoikkunoiden ja sisäseinien sijaintia ei ollut muutettu alkuperäisestä toimitetusta 3D-mallista. Sovimme välimatkaa muokattavan lähemmäksi todellisuutta seuraavan kokoonpanon valmistumiseen mennessä.

Toinen mittasuhteiden huomattiin koskien koko pankkisalin 3D-mallissa. Tilan koko tuntui suuremmalta reaali maailman tilaan verrattuna, etenkin seinien etäisyys toisistaan. Pohdimme mahdollista virtuaalitodellisuuslasien aiheuttamaa vääristymistä. Päätimme kokeilla virtuaalisen tilan skaalan muuttamista ja testaamisen jälkeen tulitaisiin käyttämään parhaaksi todettua skaalaa. Sovimme tehtävistä muutoksista ja keskustelimme seuraavaan työosuuteen liittyvistä asioista. Tilaajalta kysyttiin sisustukseen liittyvistä toiveista ja mielipiteistä. Tilaajalta saadun ohjeistuksen avulla, lähdettiin toteuttamaan seuraavaa työosuutta.

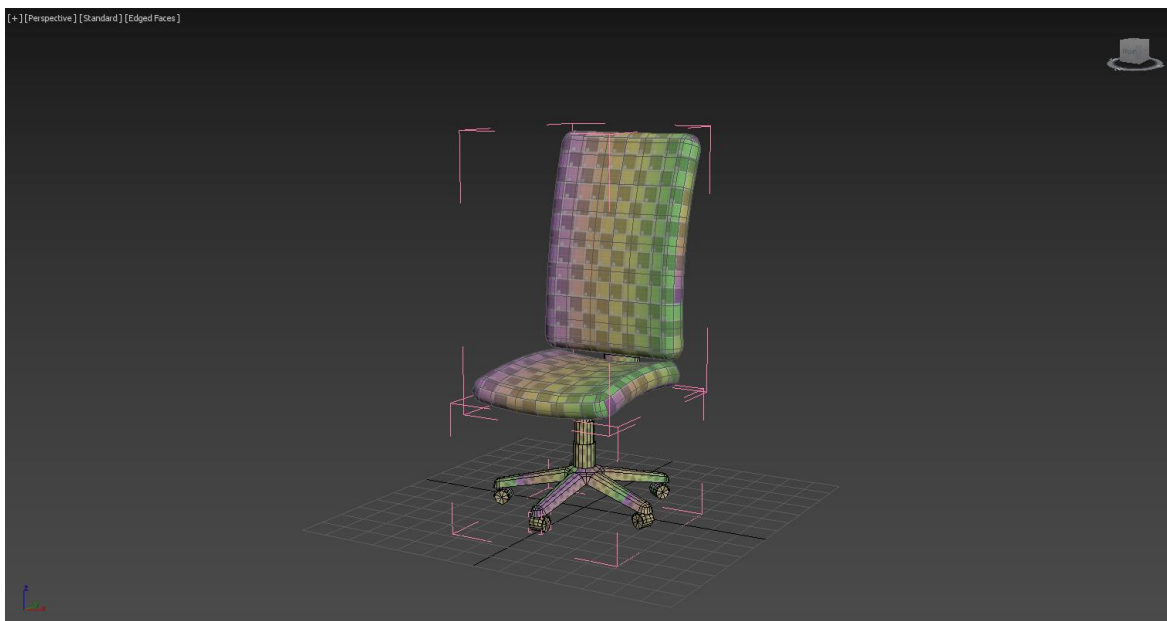
#### 4.6 Pankkisalain sisustuksen 3D-mallintaminen ja tekstuurikartoitus

Toisessa työvaiheessa aloitettiin toteuttamaan toimistosisustusta. Tilaaja antoi referenssit liittyen toimistokalusteiden ulkonäköön ja yleiseen tyyliin. Suunniteltaessa hahmoteltiin pankkisalain jakamista eri osioihin. Huonekalujen mallintaminen aloitettiin omana työtiedostona erillään jo mallinnetusta pankkisalista. Huonekalujen ollessa samassa työtiedostossa, pöytien ja tuolien mittasuhteet pysyivät realistisina suhteessa toisiinsa. Samassa työtiedostossa mallinnetut 3D-mallit nähtävissä kuvassa 20. Tuoleja lähdettiin mallintamaan erillisinä istuin- ja jalustaosioina. Tällä mahdollistettiin tuolien 3D-mallien modulaarisuus. Istuinosia ja jalkoja mallinnettaisiin muutamia erilaisia, joita yhdistelemällä saataisiin luotua erilaisia tuoleja nopeasti. Mallinnettuja tuoleja olivat muun muassa pyöröjalustaiset toimistotuolit ja kuppimuotoiset nelijalkaiset tuolit.



Kuva 20. Samassa työtiedostossa mallinnettuja toimistokalusteita

Pöydissä ei voitu hyödyntää samanlaista modulaarista lähestymistapaa. Suunnitellut pöytäkalusteet sisälsivät useita erilaisia toimistosisustukselle ominaisia pöytämalleja. Näihin mallinnettuihin pöytiin kuului muun muassa iso konferenssipöytä, korkeussäädettävä työpöytä, sohvapöytä ja ruokapöytä. Huonekalujen jälkeen siirryttiin työstämään pankkisalain tilanjakajia sekä muita toimistosisustuksen koristeellisia 3D-kappaleita. Tilanjakajien luomisessa käytettiin hyödyksi pankkisalain 3D-malliin kuuluvia lasipaneeleja ja ovia. Mallintamisen jälkeen huonekaluille luotiin tekstuurikartoitus.



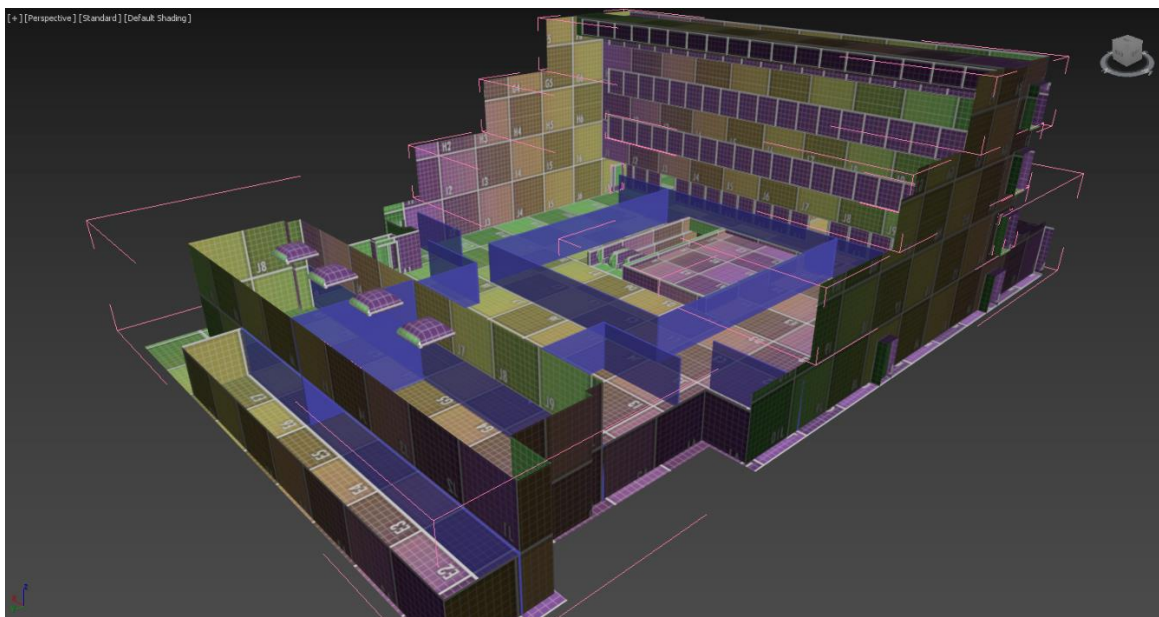
Kuva 21. Unwrap UVW -modifikaattorilla tekstuurikartoitettu toimistotuoli

Kuten ensimmäisessä työosuudessa, 3D-mallien tekstuurikartoitus luotiin 3Ds Max -mallinsohjelmiston sisäänrakennetuilla työkaluilla. UVW Map -modifikaattoria käytettiin suorakulmaisissa 3D-kappaleissa, kuten pöytätasot ja pöytienjalat sekä toimistotilan erinäköisissä seinäelementeissä ja tilanjakajissa. Pankkisalin 3D-mallista poiketen, kaikkia toimistokalusteiden tekstuurikartoituksia ei voitu luoda pelkästään UVW Map -modifikaattorin avulla. Osassa kappaleista tekstuurikoordinaatistojen käsittelyyn tarvittiin Unwrap UVW -modifikaattoria. Unwrap UVW -modifikaattoria käytettiin monimutkaisemmissa 3D-kappaleissa, jotka sisälsivät viistoja tai pyöreitä muotoja, kuten toimistotuolien jaloissa, kangastyynyissä ja koriste-esineissä. Kuvassa 21 nähtävissä toimistotuoli, jonka jalka- ja istuin-osien tekstuurikartoituksessa käytettiin Unwrap UVW -modifikaattoria.

#### 4.7 Sisustuksen asettelu

Sisustuksen asettelussa lähdettiin tavoittelemaan tilan herättämää inspiraatiota. Pankkisalin tilat rajattiin eri osastoihin, osittain yhteistiloiksi, osittain kahdelle eri toimijalle. Tilan käyttötapa suunniteltiin helposti muunnettavaksi käyttötarkoituksen mukaan ja se haluttiin tuoda esille tavalla, jolla tilat oli jaoteltu. Tilanjakajat koostuivat enimmäkseen lasiseinistä, jotta tilaan tuleva luonnonvalo pääsisi valaisemaan mahdollisimman monessa paikassa, mikä on reaali maailman pankkisalille ominainen ominaisuus. Pankkisali jaoteltiin kahteen isokokoiseen konferenssitilaan, kahteen taukotilaan, kahteen pienempään konferenssi- tai ryhmätyötilaan, yksittäisiin työhuoneisiin ja kahteen isoon avoimeen toimistotyöskentelytilaan. Tilajaon suunnittelu ja havainnollistaminen esitettynä kuvassa 22. Tilanjakojen ollessa selvillä pankkisaliin lisäitiin toimistosisustuksen huonekalut ja muut elementit.





Kuva 22. Pankkisalin jakaminen osioihin

#### 4.8 Sisustuksen 3D-mallien tuonti ja vienti

Toimistosisustuksen asettelun ollessa valmis kaikki sisustukseen liittyvät toisen työosuuden aikana luodut 3D-mallit yhdistettiin omaksi ryhmäkseen. Toimistosisustusryhmän 3D-mallit käännettiin ja tallennettiin FBX-tiedostoformaattiin. 3D-mallit tuotiin Unity-projektin resursseihin. Toimistosisustuksen 3D-malleille tehtiin samat tuontiasetusmäärittelyt kuin pankkisalin 3D-mallille. Tuontiasetuksista määriteltiin pois tarpeettomat ominaisuudet, kuten valot, kamerat, materiaalit ja animaatiot. Kuten ensimmäisen työosuuden vienti- ja tuontivaiheessa, Generate Lightmap UV's ja Swap UV's -asetukset valittiin käyttöön valaistuskarttojen ja materiaalien oikeanlaiseksi toimimiseksi. Loput tuontiasetuksista säilytettiin oletusasetuksissa ja muutokset tallennettiin FBX-tiedostoksi. Toimistosisustuksen 3D-mallit lisättiin Unity-projektin pankkisalin sisälle. Sisustuksen 3D-kappaleiden ollessa oikeilla paikoillaan pankkisalin 3D-malliin nähden 3Ds Max -mallinnusohjelmistossa, Unity-projektiin tuotu sisustus asettui myös oikealle paikalle suhteessa tilan 3D-malliin.

#### 4.9 Työn keskeneräisyys ja jatkosuunnitelmat

Toista työosuutta ei ehditty toteuttaa loppuun ennen opinnäytetyön viimeistelyä. Seuraavassa työvaiheessa tullaan lisäämään materiaalit pankkisalin 3D-malleille. Työvaihe tulee koostumaan materiaalien luomisesta ja lisäämisestä huonekaluille, seinille ja lattioille. Virtuaaliseen tilan valaistus luodaan uudelleenlaisiksi lisäämällä ja siirtelemällä ennalta olemassa olevia valonlähdeitä. Heijastusantureiden paikat ja tarkkailualueiden koot muoka-

taan jaettujen tilojen mukaisesti. Valaistukseen liittyvien asetteluiden jälkeen tullaan laske-  
maan uudet valaistuskartat, jotta huonekalut huomioidaan valaistuksessa ja varjojen luon-  
nissa. Valaistuskarttojen valmistuessa sovelluksesta tehdään toinen koonti ja lopputuote  
esitellään tilaajalle. Esittelyn jälkeen tullaan keskustelemaan työprojektin mahdollisesta  
jatkosta tai päättymisestä. Tulevat työvaiheet tullaan suorittamaan toukokuun 2019 ai-  
kana.

## 5 YHTEENVETO

Tämän opinnäytetyön tavoitteena oli tuottaa virtuaalitodellisuuslaseilla katseltava virtuaalinen tila. Virtuaalinen tila pohjautui reaali maailman Aleksis10 -liikekiinteistön tyhjiin pankkisaliin. Ensimmäisessä työosuudessa luotiin tyhjillään oleva virtuaalinen tila käyttäen apuna kiinteistöstä ennestään olemassa ollutta 3D-mallia. Luodussa virtuaalisessa ympäristössä pystyttiin liikkumaan virtuaalitodellisuusohjaimen ohjelmoidulla teleporttitoiminnolla. Ensimmäisen työosuuden päätteeksi saatiin tuotettua toimiva ja suunnitelman mukainen virtuaalitodellisuuden sovellus. Toisessa työosuudessa aloitettiin tuottamaan virtuaalisen tilan toimistosisustusta. Suurin osa toimistosisustuksen 3D-malleista ehdittiin mallintamaan, mutta sisustuksen lisäämistä virtuaaliseen pankkisaliin ei ehditty toteuttaa opinnäytetyön puitteissa. Työ on suunniteltu vietäväksi loppuun toukokuun 2019 aikana.

Opinnäytetyötä tehdessä opittiin virtuaalitodellisuuden sovelluskehittäminen olevan monisyistä ja tapauskohtaista käsityötä. Yksinkertaisenkin tilan luomiseen liittyy useita asioita, jotka on syytä ottaa huomioon. Tietämystä vaaditaan monista eri teknologioista ja ohjelmistojen tavoista toteuttaa virtuaalitodellisuuden sovelluksen ominaisuuksia. Käytettäviin optimointitapoihin tulee myös kiinnittää huomiota. Kaikkien teknologioiden käyttö ei ole aina tarpeellista tai kannattavaa, mikä on riippuvainen projektin kokoluokasta ja tavoitteista. Sovelluksen optimointiin ei ole olemassa valmiita muottia, mutta optimoinnin toimintaperiaatteita noudattamalla selvittää hyvään lopputulokseen. Optimoiminen tulee olemaan tärkeä osa virtuaalitodellisuuden sovelluskehittämistä niin kauan, kuin suoritettavuus on virtuaalitodellisuuden sovellusten ominaisuuksia rajoittava tekijä.

Virtuaalitodellisuuden sovellusta kehitettiin ja käytettiin HTC Vive -virtuaalitodellisuuslaitteiden avulla. HTC Vive -virtuaalitodellisuuslaitteiden käyttäminen ei kuitenkaan ollut sovelluksen kannalta merkittävä tekijä. Pienillä muutoksilla sovellusta pystyttäisiin käyttämään muillakin virtuaalitodellisuuden laitteilla liitännäisten tarjoaman tuen vuoksi. Kehitettyä sovellusta voitaisiin käyttää hyvänä aloituspohjana vastaavanlaisissa tulevaisuuden työtoteutuksissa. Toiminnollisuuksien vaatimattomuus ja saman kokoluokan virtuaalinen ympäristö mahdollistaisivat nopean sovelluskehittämisen. Vanhat visuaaliset elementit, kuten 3D-mallit, tekstuurit, materiaalit ja valonlähteet korvattaisiin uusilla elementeillä.

## LÄHTEET

Abrash, M. 2013. Down the VR rabbit hole: Fixing judder [viitattu 3.5.2019]. Saatavissa:

<http://blogs.valvesoftware.com/abrash/down-the-vr-rabbit-hole-fixing-judder/>

Aukstakalnis, S. 2017. Practical Augmented Reality. a Guide to the Tehnologies, Applications, and Human Factors for AR and VR. Boston: Addison-Wesley.

Autodesk Inc. 2017a. Unwrap UVW Modifier [viitattu 4.5.2019]. Saatavissa:

<https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-EA10E59F-DE7F-497E-B399-6CF213A02C8D-htm.html>

Autodesk Inc. 2017b. UVW Map Modifier [viitattu 4.5.2019]. Saatavissa:

<https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-78327298-4741-470C-848D-4C3618B18FCA-htm.html>

Autodesk Inc. 2018. Export menu [viitattu 10.5.2019]. Saatavissa:

<https://help.autodesk.com/cloudhelp/2019/ENU/3DSMax-Data-Exchange/images/GUID-F19AE7CF-F8FC-40B7-948B-12EDC182EF9F.png>

Aversis 3D 2019. HDRI image [viitattu 10.5.2019]. Saatavissa:

<http://www.aversis.be/hdri/images/individual/ldr/013-weg-naast-wei-ldr.jpg>

Facebook Technologies, LLC. 2014. Oculus Rift - Persistence [viitattu 10.5.2019].

Saatavissa: <https://www.rockpapershotgun.com/images/14/mar/rift.jpg/RPSS/resize/760x-1/format/jpg/quality/90>

Joe van den Heuvel 2018. SteamVR Skeletal Input [viitattu 10.5.2019]. Saatavissa:

<https://steamcommunity.com/games/250820/announcements/detail/1690421280625220068>

Sysdia Solutions LTD. 2019. VRTK - Virtual Reality Toolkit - [ VR Toolkit ] [viitattu

7.5.2019]. Saatavissa: <https://assetstore.unity.com/packages/tools/integration/vrtoolkit-vr-toolkit-64131>

Unity Technologies 2017a. FBX export guide [viitattu 7.5.2019]. Saatavissa:

<https://docs.unity3d.com/2017.4/Documentation/Manual/HOWTO-exportFBX.html>

Unity Technologies 2017b. Importing Objects From 3D Studio Max [viitattu 7.5.2019].

Saatavissa: <https://docs.unity3d.com/2017.4/Documentation/Manual/HOWTO-ImportObjectMax.html>

Unity Technologies 2019a. Baked Lighting [viitattu 9.5.2019]. Saatavissa:

<https://docs.unity3d.com/Manual/LightMode-Baked.html>

Unity Technologies 2019b. Choosing a Lighting Technique [viitattu 9.5.2019]. Saatavissa:

<https://unity3d.com/learn/tutorials/topics/graphics/choosing-lighting-technique>

Unity Technologies 2019c. Draw call batching [viitattu 29.4.2019]. Saatavissa:

<https://docs.unity3d.com/Manual/DrawCallBatching.html>

Unity Technologies 2019d. Forward Rendering Path Details [viitattu 9.5.2019].

Saatavissa: <https://docs.unity3d.com/Manual/RenderTech-ForwardRendering.html>

Unity Technologies 2019e. Legacy Deferred Lighting Rendering Path [viitattu 9.5.2019].

Saatavissa: <https://docs.unity3d.com/Manual/RenderTech-DeferredLighting.html>

Unity Technologies 2019f. Materials, Shaders & Textures [viitattu 6.5.2019]. Saatavissa:

<https://docs.unity3d.com/Manual/Shaders.html>

Unity Technologies 2019g. New Empty Material [viitattu 10.5.2019]. Saatavissa:

<https://docs.unity3d.com/uploads/Main/StandardShaderNewEmptyMaterial.png>

Unity Technologies 2019h. Optimisation for VR in Unity [viitattu 4.5.2019]. Saatavissa:

<https://unity3d.com/learn/tutorials/topics/virtual-reality/optimisation-vr-unity>

Unity Technologies 2019i. Quality [viitattu 8.5.2019]. Saatavissa:

<https://docs.unity3d.com/Manual/class-QualitySettings.html>

Unity Technologies 2019j. Reflection Probe [viitattu 8.5.2019]. Saatavissa:

<https://docs.unity3d.com/Manual/class-ReflectionProbe.html>

Unity Technologies 2019k. Single Pass Stereo rendering (Double-Wide rendering) [viitattu

8.5.2019]. Saatavissa: <https://docs.unity3d.com/Manual/SinglePassStereoRendering.html>

Unity Technologies 2019l. Types of Light [viitattu 8.5.2019]. Saatavissa:

<https://docs.unity3d.com/Manual/Lighting.html>

Unity Technologies 2019m. Vertex Lit Rendering Path Details [viitattu 9.5.2019].

Saatavissa: <https://docs.unity3d.com/Manual/RenderTech-VertexLit.html>

Valve Corporation 2019. SteamVR Unity Plugin [viitattu 5.5.2019]. Saatavissa:

[https://valvesoftware.github.io/steamvr\\_unity\\_plugin/articles/intro.html](https://valvesoftware.github.io/steamvr_unity_plugin/articles/intro.html)

Varjo Technologies 2019. Varjo VR-1 [viitattu 10.5.2019]. Saatavissa:

[https://varjo.com/wp-content/uploads/2019/02/Varjo\\_VR%E2%80%931\\_3.jpg](https://varjo.com/wp-content/uploads/2019/02/Varjo_VR%E2%80%931_3.jpg)

Wikipedia 2018. Kuvataajuus [viitattu 5.5.2019]. Saatavissa:

<https://fi.wikipedia.org/wiki/Kuvataajuus>

Wikipedia 2019a. Jitter (optics) [viitattu 9.5.2019]. Saatavissa:

[https://en.wikipedia.org/wiki/Jitter\\_\(optics\)](https://en.wikipedia.org/wiki/Jitter_(optics))

Wikipedia 2019b. Refresh rate [viitattu 5.5.2019]. Saatavissa:

[https://en.wikipedia.org/wiki/Refresh\\_rate](https://en.wikipedia.org/wiki/Refresh_rate)

Wikipedia 2019c. Texture Mapping [viitattu 5.5.2019]. Saatavissa:

[https://en.wikipedia.org/wiki/Texture\\_mapping](https://en.wikipedia.org/wiki/Texture_mapping)

Ziff Davis, LLC. 2011. Fallout micro stutter [viitattu 10.5.2019]. Saatavissa:

<https://www.extremetech.com/wp-content/uploads/2011/08/fallout-micro-stutter.jpg>