

Tuomas Riiste

Pelimoottorin käyttö valaistussuunnittelun visualisoinnin työkaluna

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Talotekniikka

Insinöörityö

03.03.2018

Tekijä Otsikko	Tuomas Riiste Pelimoottorin käyttö valaistussuunnittelun visualisoinnin työkaluna
Sivumäärä Aika	34 sivua 03.03.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	talotekniikka
Ammatillinen pääaine	sähköinen talotekniikka
Ohjaajat	yksikönpäällikkö Hannu Virkkunen lehtori Jarmo Tapio
<p>Insinööriyön aiheena oli tutkia pelimoottori Unreal Enginen käyttöä valaistussuunnittelun visualisoinnissa. Tavoitteena oli saada tietoa siitä, mitä uutta tai hyödyllistä pelimoottori tuo valaistussuunnittelun visualisointiin, tutkia, saadaanko pelimoottorilla tarpeeksi hyvää jälkeä suunnittelua varten, sekä tutkia, onko sen käyttö ajallisesti järkevää tuloksiin nähden.</p> <p>Työssä käytettiin suunnitteluohjelmaa Autodesk Revitiä tilojen luomiseen, 3D-mallintamisohjelmaa 3dsMaxia, jotta 3D-objektit pystyttiin muokkaamaan pelimoottoria varten, sekä pelimoottori Unreal Engineä itse pelin tekemiseen. Lisäksi vertailussa käytettiin Dialux Evoa.</p> <p>Työssä tehtiin kaksi erillistä peliä Unreal Enginellä, jotta tulokset olisivat helposti tarkasteltavissa. Ensimmäisessä pelissä suoritettiin pelkkää visuaalista vertailua Unreal Enginen ja Dialux Evon välillä. Siinä osoitettiin, että vaikka pieniä visuaalisia eroja on ohjelmien välillä, ovat ne kuitenkin tarpeeksi samanlaisia, jotta Unreal Engineä voitaisiin harkita valaistussuunnittelun visualisoinnin yhtenä työkaluna.</p> <p>Toisessa pelissä luotiin kolme erilaista tilaa, joissa kaikissa esitellään erilaisia valaistusratkaisuja, joita pystytään vaihtamaan pelin aikana edestakaisin. Tämä peli toi hyvin esille Unreal Enginen tuomat lisämahdollisuudet valaistuksen visualisointiin, kuten nopeat eri valaisinvaihtojen esittelyyn, liikkuvat valot sekä väriä vaihtavat valot. Kaiken pystyy tekemään reaaliaikaisesti pelin sisällä.</p>	
Avainsanat	pelimoottori, Unreal Engine, valaistussuunnittelu, visualisointi

Author Title	Tuomas Riiste Game Engine as a Tool for Visualization in Lighting Planning
Number of Pages Date	34 pages 3 March 2018
Degree	Bachelor of Engineering
Degree Programme	Building Services Engineering
Professional Major	Electrical Engineering for Building Services
Instructors	Hannu Virkkunen, Head of Department Jarmo Tapio, Senior Lecturer
<p>The goal of this thesis was to establish whether the game engine Unreal Engine could bring new tools to the visualization of lighting planning, establish whether it can produce adequate results and whether its use is feasible time-wise. For this, two games were created. One compared the visual similarity between Unreal Engine and Dialux Evo, the commonly used software. The other game demonstrated whether the game engine can quickly change between views with different lighting conditions.</p> <p>The spaces for the first game were made with Autodesk Revit, and 3dsMax was used to modify the objects to suit Unreal Engine. The game showed that Unreal Engine and Dialux Evo only have some minor visual differences. The second game showed that the game engine brings various new tools for visualization, since the changes in illumination works in real time.</p> <p>Although there were some problems when working the games; for example, it took time to get the 3D models to work properly, and actual lighting data, like illuminance figures, was missing, the games proved that Unreal Engine is a serious alternative to be used as additional tool for the visualization of lighting planning.</p>	
Keywords	game engine, Unreal Engine, lighting planning, visualization

Sisällys

Lyhenteet

1	Johdanto	1
2	Valaistuksen visualisointi	2
2.1	3dsMax	2
2.2	Dialux	4
2.3	Pelimoottorit	6
3	Pelimoottori	7
3.1	Pelimoottorin historia	7
3.2	Pelimoottorin käyttö muilla aloilla	9
4	Pelimoottorin käyttö valaistussuunnittelussa	11
4.1	Pelimoottorin valinta	11
4.2	Valolähteet	11
4.2.1	UV-mappaus	14
5	Vertailu	18
5.1	Tavoite	18
5.2	Kohteet ja valaisimet	18
5.2.1	Valaisin 1	20
5.2.2	Valaisin 2	20
5.2.3	Valaisin 3	21
5.3	Tulokset	21
5.4	Tulosten pohdinta	24
6	Peli	25
6.1	Tavoite	25
6.2	Pelin tilojen luominen	26
6.3	Aula	26
6.4	Neuvotteluhuone	27
6.5	Ulkotila	28

7 Loppupäätelmät

31

Lähteet

33

Lyhenteet

3dsMax	3D-mallinnus- ja renderöintiohjelma.
.dwg	CAD-ohjelmien käyttämä tiedostomuoto.
.fbx	Tiedostomuoto 3D-objekteille.
UV-mappaus	3D-objektien teksturoinnin prosessi.
LED	Valonlähde, Loistediodi. Light-Emitting Diode
.ies / .ldt	Valonjakokäyrän tiedostomuodot

1 Johdanto

Oman kokemukseni mukaan valaistussimulointi on nousemassa esiin osana valaistus-suunnittelua koko ajan isommin. Tapoja, joilla voitaisiin erottua joukosta valaistussuunnittelun maailmassa, etsitään koko ajan. Viimeisimpänä asiana, joka on noussut esille koulussa luennoilla ja työpaikan puheenaiheina, on pelimoottorien käyttö. Niitä on aiemmin käytetty, kuten nimi sen sanoo, pelien tekemiseen. Pelimoottorien kehittyminen mahdollisti niiden käytön esimerkiksi arkkitehti- tai maisemasuunnittelussa, jossa haluttiin tehdä tiloja, joissa voidaan ikään kuin kävellä sisällä ja katsella erilaisia yksityiskohtia eri kuvakulmista.

Itse kuulin ensimmäisen kerran pelimoottorin käytöstä valaistussuunnittelussa luennolla, jossa esiteltiin Helsingin Kruunuvuoren sillan valaistussuunnitelmaa. Projektiin tehtävä valaistussuunnitelma oli nähtävissä pelissä, joka oli tehty pelimoottori Unitylla. Tässä pelissä pystyy kulkemaan kävelysiltaa pitkin ja tarkkailemaan sillan vaihtuvaa valaistusta sekä vaihtuvaa päivän aikaa.

Omat aiemmat kokemukset valaistuksen simuloinnissa ovat olleet lähinnä PDF-muodossa esiteltäviä periaatekuvia suunniteltavasta valaistuksesta. Halusin tuoda lisää syvyyttä tuohon 2D-malliseen simulointiin, joten ajattelin tutkia lisää pelimoottoreita ja niiden mahdollisuuksia valaistussimuloinnissa.

Opinnäytetyön tavoitteena on tehdä kaksi peliä. Ensimmäisen pelin tarkoitus on vertailla pelimoottorin ja Dialux Evon tuloksia visuaalisesti muutamassa yksinkertaisessa tilassa. Toinen peli sijoittuu Rambollin uuden pääkonttorin tiloihin, joihin itse kehitellen erilaisia valaistusratkaisuja, joita voitaisiin vaihdella lennosta. Peliin tuleva valaistus on minun itseni suunnittelema, eikä se vastaa Rambollin pääkonttoriin virallisia valaistussuunnitelmia. Aion tuoda tällä pelillä esille niitä ominaisuuksia pelimoottorista, joita ei muilla perinteisemmällä valaistussuunnittelun ohjelmilla ole.

2 Valaistuksen visualisointi

Valaistuksen suunnittelun tärkeimpiä tavoitteita on saavuttaa tarpeeksi hyvä taso asioihin, kuten valaistusvoimakkuuteen, tasaisuuteen ja värintoistoon. Valaistus on iso osa myös arkkitehtuuria ja sisustussuunnittelua, jolloin itse valaisimen ulkonäkö ja sen valaistusominaisuudet ovat myös tärkeitä.

Entistä enemmän valaistuksella ja valaisimilla voidaan saada aikaan suuria muutoksia siihen miltä suunniteltava tila näyttää. Tässä osaksi on ollut mukana LED-tekniikan kehittyminen, joka mahdollistaa valaisimien monimuotoisemman ulkonäön. Kun suunniteltuun ja rakennettuun tilaan kävellään ensimmäisen kerran, kiinnittyy katse hyvin nopeasti valoon ja valaistukseen. Miten valaistus toimii yhdessä muun sisustuksen ja rakenteiden kanssa?

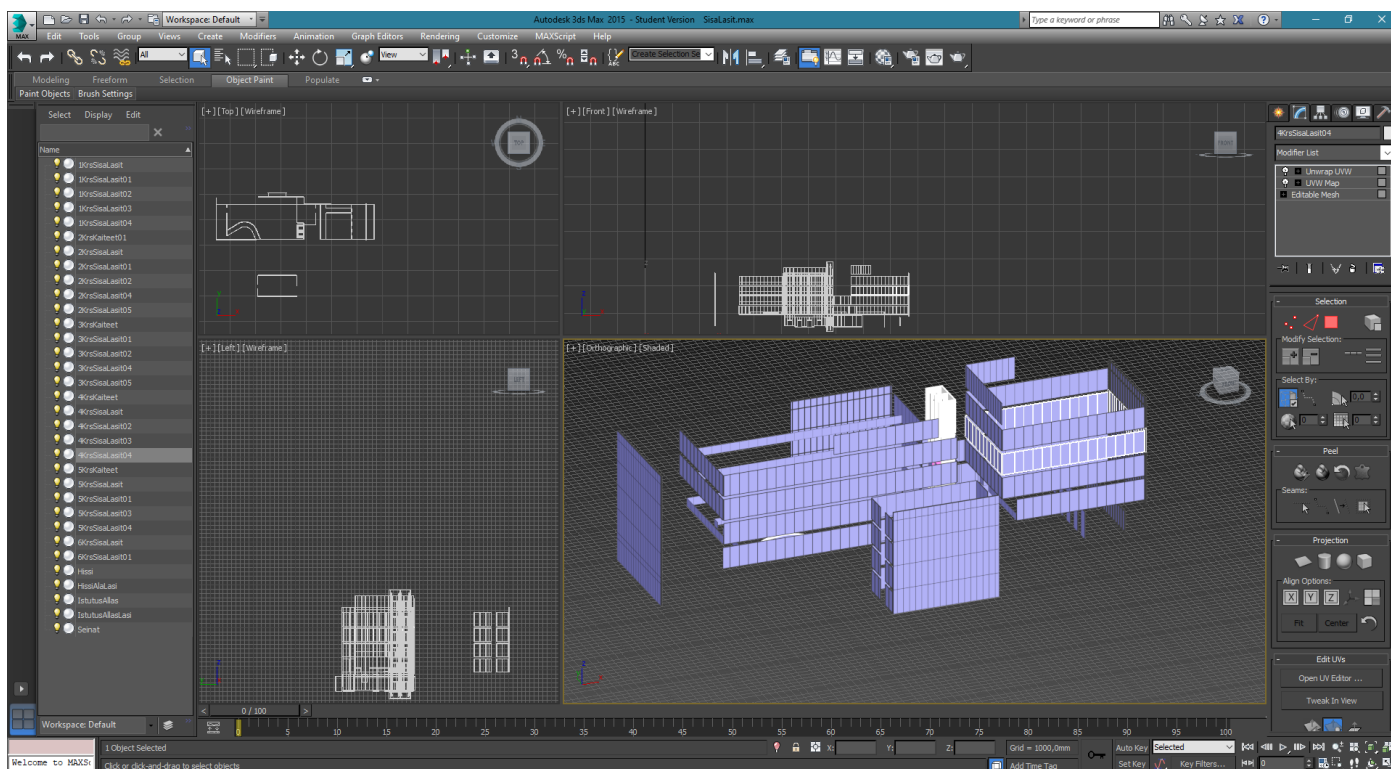
Jos kyseessä on pienempi tila yksinkertaisilla kalusteilla, voidaan visualisointi toteuttaa, esimerkiksi Dialux Evoa käyttäen. Jos asiakas haluaa hyvin realistista kuvaa kohteesta, työkaluna voi toimia hyvin esimerkiksi mallintamistyökalu 3dsMax. Muitakin työkaluja visualisoinnissa on, mutta rajoitan tässä työssä vertailun näihin kahteen, koska näitä työkaluja käytetään myös Rambollissa.

2.1 3dsMax

3dsMax on ohjelma, jolla voidaan tehdä 3D-mallintamista, animointia, kuvia ja pelejä. Aiemmin tunnettu 3D Studiona ja 3D Studio Maxina. Ohjelman takana on Autodesk, joka aiemmin toimi vain ohjelman julkaisijana, mutta myöhemmin otti haltuunsa myös ohjelman kehittämisen. Autodeskin tuotteita ovat myös AutoCAD ja Revit, jota käytetään myös Rambollilla. Tästä oli minun työssäni hyötyä, koska 3dsMax, AutoCAD ja Revit toimivat keskenään hyvin ja käyttävät samoja tiedostomuotoja, kuten dwg ja fbx. Ohjelman käyttöliittymän (kuva 1) ymmärsin nopeasti, koska Autodeskin tuotteita olin ennenkin käyttänyt.

Omat kokemukset 3dsMaxista ennen tämän projektin aloittamista olivat täysin olemattomat. Pelimoottoria käytettäessä ollaan paljon tekemisissä 3D-mallien kanssa. Unreal Enginen 3D-mallintamisominaisuudet ovat suhteellisen pienet, joten tätä työtä tehdessä 3dsMax oli se ainoa työkalu 3D-mallintamiseen. Aloitin ohjelman opettelun täysin

alkeista käyttäen Youtubesta löydettyjä opetusvideoita. Suurin ongelma oli mallien UV-mappaus, joka mahdollistaa, että valaistus ja varjot näyttäytyvät oikeanlaisena objektin pinnalla.



Kuva 1. 3dsMax, ulkoasu. Kuvassa Rambollin pääkonttorin ikkunoita.

3dsMax voidaan luoda 3D-mallin pintamateriaalille oikeanlaisen heijastussuhteen, rosioisuuden ja kuvioinnin, jotta valaistus näyttäisi mahdollisimman oikealta. Esimerkiksi jos kyseessä on marmoripinta, sille voidaan asettaa tilanteen mukaan vaikkapa mattamainen hiottu pinta tai kiillotettu melkein peilimäinen pinta. Tarvittavia materiaaleja on esillä internetissä suuri määrä ilmaisina ja maksullisina, mutta ohjelmaan syventynyt henkilö voi luoda omat materiaalit ohjelman sisäisellä Material Editorilla. Material Editorin avulla voidaan luoda sileästä täysin heijastavasta peilimateriaalista aina karheaan ulkoverhoustiiliin asti. Tämä on tärkeä asia, kun mallinnetaan heijastuvaa valoa.

Tero Myhrberg tutki opinnäytetyössään 3dsMaxin käyttöä arkkitehtuurisessa valaistus suunnittelussa. Opinnäytetyö tehtiin myös Rambollille. Insinöörityössä tutkittiin laskennallisesti ja ulkonäöllisesti, kuinka lähelle todellisia tuloksia päästään 3dsMaxia käyttäen. Tuloksista kävi ilmi, että laskennallisesti arvot olivat hieman suurempia, lukuun ottamatta muutamaa poikkeusta, mutta tulokset kuitenkin myötäilivät mitattuja arvoja.

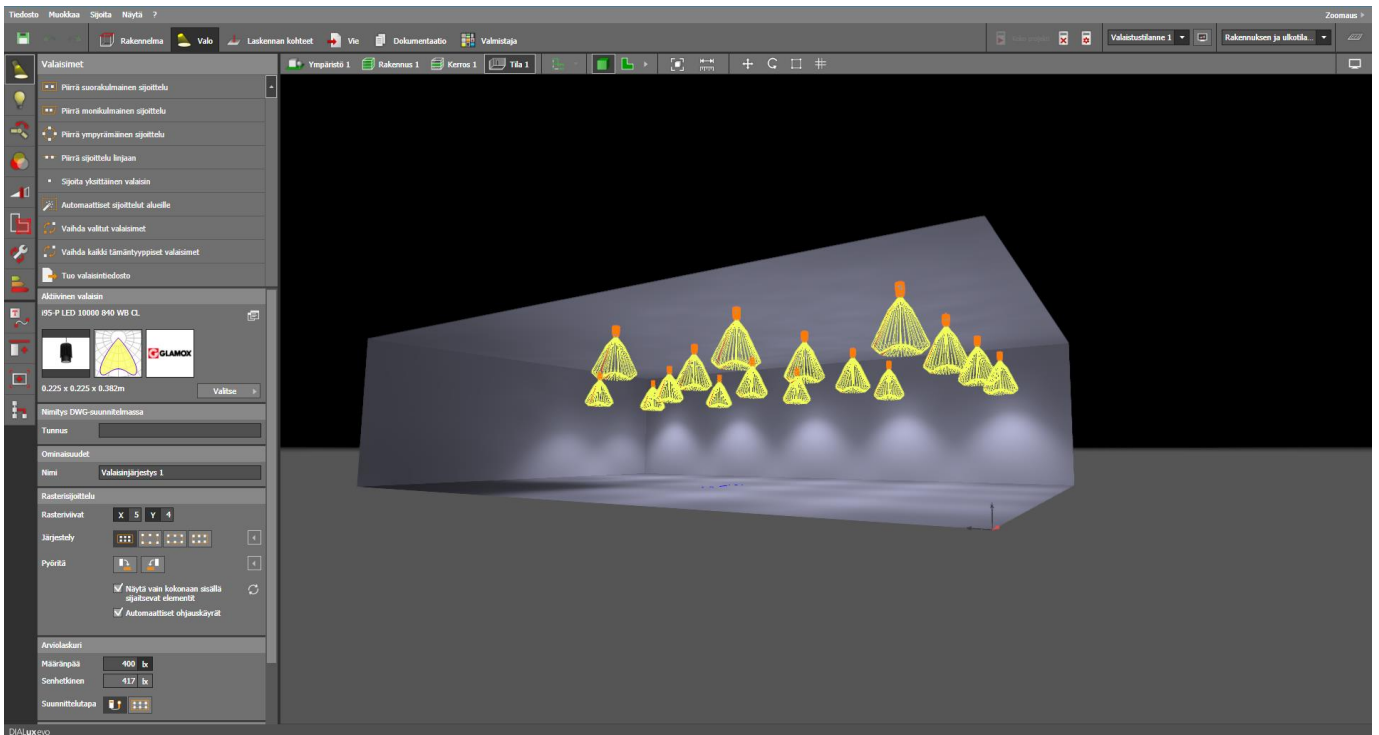
Ulkonäöllisesti 3dsMax pystyi havainnollistamaan hyvin oikeaa tulosta, vaikka kuvat olivat hieman himmeämpiä kuin mittaustilanteessa otetut valokuvat. [1.]

2.2 Dialux

Dialux on ammattikäyttöön kehitetty ilmainen valaistussuunnitteluohjelma, jonka avulla voidaan saada visuaalisia ja numeerisia tuloksia tilojen valaistuksesta. Dialux on yleisimpiä Suomessa käytettyjä ohjelmia valaistussuunnittelussa. Ohjelmasta on julkaistu kaksi versiota: Dialux 4 ja Dialux Evo. Kummallakin pystytään tekemään valaistuslaskelmia, mutta Dialux 4 on tarkoitettu yksinkertaisimpiin sisä- ja ulkotiloihin, kun taas Dialux Evolla voidaan tehdä laskelmat kokonaiseen rakennukseen ja sen ulkotiloihin. Oman kokemuksen mukaan Dialux Evolla voidaan tehdä käytännössä samat asiat kuin Dialux 4:llä, mutta se on hieman raskaampi ja hitaampi käyttää, jos halutaan tehdä yksinkertaiseen tilaan laskelmat. Valaistuksen visualisoinnissa kuitenkin Dialux Evo on kehittyneempi työkalu.

Valaistuksella on käytössä omat standardinsa, jotka täytyy ottaa huomioon valaistusta suunniteltaessa. Kuten aiemmin mainittu Tero Myhrbergin tutkimus osoitti, on 3dsMax hieman epäluotettava pelkkiä lukuja katseltaessa. Valaistussuunnittelun laskennallisessa puolessa Dialuxit ovat yksiä parhaita työkaluja, koska niillä voidaan luoda helposti erilaisia tiloja materiaaleineen ja ohjelmasta saadaan ulos helposti tietoa valaistuksen esimerkiksi numeerisista arvoista. Dialuxilla voidaan ottaa huomioon myös auringon- tai kuunvalo, jos halutaan luoda tiloille päivä- ja iltanäkymät.

Valaisimien määrää valittaessa voidaan ohjelmaan määritellä, millainen tila on kyseessä, jolloin ohjelma ilmoittaa tilalle tarvittavat lux-määrät sekä ehdottaa valaisimien määrää ja järjestystä. Myös laskennasta saatava raportti viittaa sisävalaistusstandardiin, jolloin voidaan varmistua siitä, että tarvittavat luvut täyttyvät. Ohjelman avulla on helposti tarkasteltavissa valaisimen tekniset tiedot ja valonjakokäyrät, kuten kuvassa 2 näkyy.



Kuva 2. Dialux Evo, ulkoasu. Kuvassa vertailussa käytettävä hallitila.

Pelkkää visualisointia ajatellen on Dialux Evo parempi kuin Dialux 4, mutta kuitenkin puutteellinen. Materiaaleja pystyy valitsemaan ohjelman sisällä, mutta niiden muokkaus täytyy tehdä ulkopuolisilla ohjelmilla. Itse valon luominen vaatii käytännössä myös ulkopuolisen lähteen. Dialux Evo on siis riippuvainen tiedostomuodoista kuten IES tai ULD, jotta se voi luoda valonlähteen. Suunniteltaessa periaatteellista valoa, jolle ei ole varsinaisesti valmista tuotetta, on käytettävä jotain valaisinta, joka on lähellä sitä, mitä suunnittelija mielessään ajattelee. Myös kalusteiden ja muiden 3D-mallien luominen on Dialuxissa rajattua, joten tarvittaessa pitää käyttää ulkopuolista mallintamisohjelmaa apuna.

2.3 Pelimoottorit

Näiden kahden edellä mainitun ohjelman rajoitteena, visualisointia ajatellen, on kuitenkin staattisuus. Esimerkiksi Dialux on loistava työkalu, kun halutaan yksinkertaista visualisointia ja raakaa dataa valaistuksesta, kuten lux-määriä tietyssä kohdassa ja esimerkiksi valaistuksen tasaisuus autotietä suunnitellessa. Ulkonäöllisesti kuitenkin jäädyään kuitenkin mallintamishjelmien taakse. 3dsMaxilla saadaan erittäin realistista kuvaa, jopa liikkuvaa kuvaa, mutta sekin liikkuvuus on ennalta määriteltyä, raiteilla kulkevaa.

Suunniteltaessa tilaa, joka on kooltaan suuri tai arkkitehtuurisesti erikoinen, joka sisältää monenlaista valoa yhdessä suuressa tilassa ja jonka keskipisteenä halutaan suuri ja näyttävä valaistus, olisi suuri etu, jos tilassa voitaisiin liikkua vapaasti ja tarkastella valaistusta eri kuvakulmista. Haluttaessa voitaisiin myös pienellä vaivalla kokeilla erilaisia valaistusvaihtoehtoja reaaliaikaisesti. Tässä tilanteessa pelimoottorin käyttö tulee hyödylliseksi.

Hyvä yhdistelmä on käyttää pelimoottoria 3dsMaxin kanssa. 3dsMaxilla voidaan luoda käytännössä minkäläistä 3d-muotoa tahansa ja asettaa materiaalit ja luoda UV-mappauksen helposti. 3dsMax osaa myös lukea dwg-, 3D-dwg- ja fbx-tiedostoja, joten yhteistyö Autodeskin ohjelmistojen kanssa, kuten tässä työssä käytetyn Revitin, onnistuu. Usealla valaisinvalmistajalla on ilmaiseksi saatavilla näitä tiedostomuotoja valaisimistaan, joten visualisointiin saadaan oikean näköiset tuotteet esille. Tästä työn kulusta kerroon enemmän myöhemmin.

3 Pelimoottori

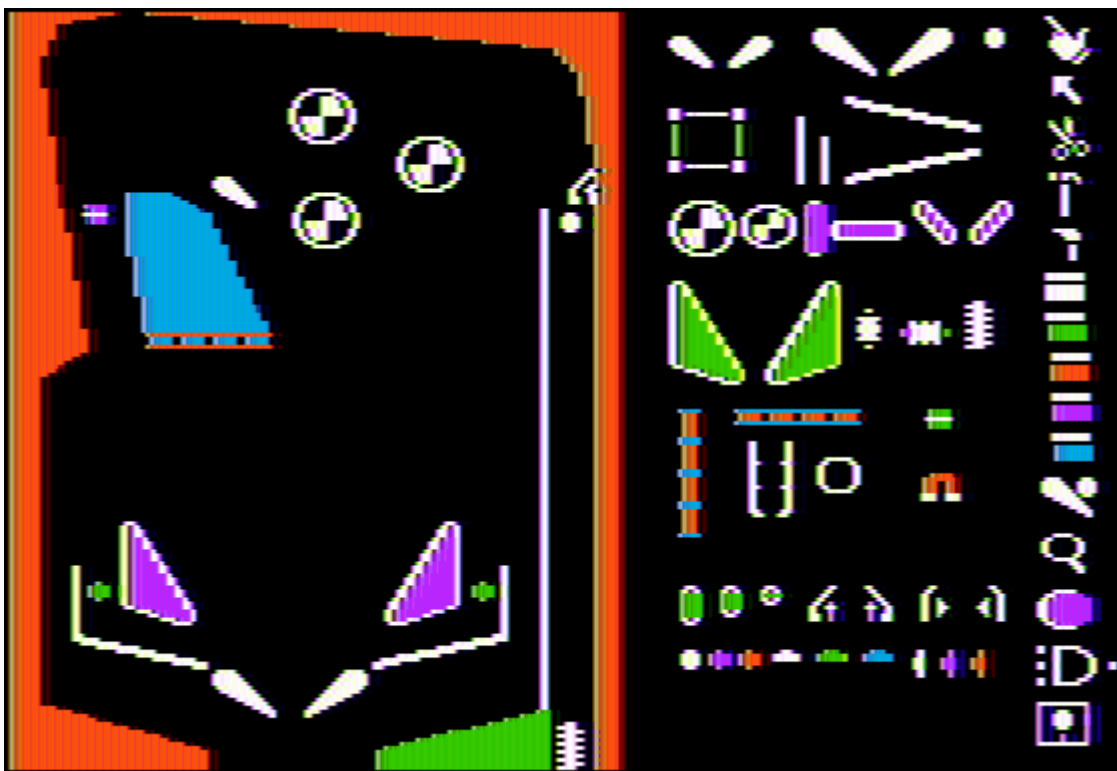
Pelimoottori on ohjelmistokehys, jonka päälle pelintekijät voivat rakentaa haluamansa pelin. Pelimoottorit yleensä sisältävät fysiikkamoottorin, törmäyksen tunnistuksen, äänet, komentosarjakielen ohjelmointia varten, animoinnin, valaistuksen sekä renderöintimoottorin 2D- ja 3D-grafiikalle. Renderöinti on yksinkertaisimmillaan kuvan luomista mallista tietokoneohjelman kanssa. Pelimoottorin tapauksessa luodaan liikkuvaa ja ohjattavaa kuvaa. Fysiikkamoottori ja törmäyksen tunnistus antavat pelissä käytettäville objekteille massan ja painovoiman, jotta ne voivat käyttäytyä pelissä kuten oikeassa elämässä. [2.]

Erilaiset moottorit voivat olla erillisiä ohjelmistoja ohjelmiston sisällä. Esimerkiksi tässä projektissa käytettävä Unreal Engine -pelimoottori käyttää sisällään PhysX-fysiikkamoottoria, jota ylläpitää NVIDIA, joka on tunnettu mm. näytönohjaimista.

Nykyään pelimoottorit ovat niin kehittyneitä, että pelisuunnittelija voi alkaa tekemään peliä ilman suurempaa tietämystä koodaamisesta, koska ohjelma sisältää jo niin paljon tekniikkaa. Aiemmin vaikkapa jonkun objektin putoaminen ja sen jääminen lattialle vaati koodin kirjoittamista, mutta nykyään tämä kaikki on jo valmiina ohjelmassa.

3.1 Pelimoottorin historia

Ihan ensimmäisinä pelimoottoreina voitaisiin pitää GCS:ää (game creation system), joka suoraan suomennettuna tarkoittaa pelinluontijärjestelmää. GCS:t tulivat markkinoille 80-luvulla, ja ne olivat käytännössä setti yksinkertaisia työkaluja, joilla pystyi luomaan hyvin tietynlaisen pelin. GCS:t olivat tarkoitettu kuluttajille. Tästä esimerkkinä on EA:n kehittämä GCS, nimeltä Pinball Construction Set (kuva 3). Kuten nimi kertoo, se oli käytännössä peli, jolla pystyi tekemään erilaisia flipper-pelejä omassa kodissa. Se julkaistiin vuonna 1983 Atari 8-bittiselle, Apple II:lle sekä Commodore 64:lle. [3.]



Kuva 3. Game Creation System, Pinball Construction set [4].

90-luvulla alkoi kuitenkin yleistymään 3D-grafiikka, joka myös yleistyi sellaiset pelimoottorit, joita nykyään käytetään. Puolesta välissä 90-lukua Id Software julkaisi suosittu pelit Doom (kuva 4) ja Quake. Näiden pelien ydintoiminnot, kuten törmäyshavainnointi ja fysiikkamoottori, lisensoitiin muiden pelintekijöiden käytettäväksi. Tämän avulla pelintekijöiden ei tarvinnut lähteä nolasta kehittämään peliään, vaan pystyttiin heti keskittymään asioihin kuten grafiikka ja hahmot. [5.] Tämä auttaa myös luomaan jatko-osia helpommin, koska itse pelin rungon raaka koodaaminen on jo tehty ja voidaan keskittyä pelin ulkonäölliseen puoleen. Vuonna 2003 tehdyssä tutkimuksessa tarkasteltiin pelintekijätiimejä ja huomattiin, että varsinaisten ohjelmoijien määrä oli vähentynyt suhteessa taiteilijoihin ja suunnittelijoihin. [6.]



Kuva 4. Kuvakaappaus alkuperäisestä Doom-pelistä. [7.]

3.2 Pelimoottorin käyttö muilla aloilla

Pelimoottorien kehittyessä koko ajan teknisemmäksi ja käyttäjäystävällisemmäksi on niiden käyttötarkoitus myös laajentunut muille aloille sekä erityylisiin, vakavampiin, peleihin. Pelimoottorilla voidaan tehdä esimerkiksi oikeita lentosimulaattoreita, joilla koulutetaan lentäjiä, sekä armeijasimulaattoreita, joilla voidaan testata erilaisia sotatilanteita. Käyttötavat ovat myös laajentuneet myös esimerkiksi elokuvateollisuuteen ja arkkitehtuuriseen suunnitteluun.

Pelistudio Epic Games julkaisi vuonna 2015 lyhytanimaatioelokuvan, A Boy & His Kite, joka on tehty täysin käyttäen firman omaa pelimoottoria Unreal Engineä. Elokuvan tarkoitus oli rohkaista elokuvantekijöitä käyttämään pelimoottoria animaatioelokuvien tekemiseen. Toisena hieman erilaisesta pelimoottorin käytöstä elokuvassa on vuonna 2016 julkaistu scifi-elokuva, The Passengers, jossa näyttelijöiden käyttämien tablettien käyttöliittymä oli tehty ja suunniteltu Unity-pelimoottorilla. [8.]

Arkkitehtuurisessa suunnittelussa pelimoottori on hyvä lisätyökalu visualisointiin. Suunnittelija voi itse vaeltaa kohteessaan tarkastellen tilaa eri kuvakulmista, kokeillen eri materiaaleja tai rakenteita. Pelimoottorit voivat toimia myös hyvänä tapana osoittaa asiakkaalle tarkemmin ja näyttävämmin, millaista kohdetta ollaan suunnittelemassa (kuva 5). Pelimoottorin avulla tilassa voidaan kävellä käyttäen normaalia näyttöä tai vaikka virtuaalilaseja, jolloin kokemus on täysin uudenlainen. Lisäksi tarkasteluun voidaan tulevaisuudessa yhdistää BIM-tietomalli, jolloin tarkastellessa esimerkiksi rakennuksen ikkunaan, tulevat samalla näytölle näkyviin ikkunan oikeat tekniset tiedot.



Kuva 5. Unreal Engine -pelimoottorin keskustelualueen käyttäjän K.M.A:n, Unreal Enginellä toteutettu asunto [9].

4 Pelimoottorin käyttö valaistussuunnittelussa

4.1 Pelimoottorin valinta

Projektia aloittaessa minulla oli kaksi isoa kriteeriä, joiden mukaan pelimoottorin valinta täytyi tehdä. Ensimmäinen oli hinta. Pelimoottorin siis täytyi olla ilmainen. Se ei onneksi tuottanut ongelmia, koska tiesin heti, että kaksi suosittua vaihtoehtoa Unity ja Unreal Engine ovat kummatkin ilmaisia. Toisena kriteerinä oli minun olematon kokemus pelimoottoreista ja skriptaamisesta. En ollut ikinä opetellut mitään koodaamiskieltä tai koskenutkaan pelimoottoriin. Näiden kriteerien perusteella päädyin Unreal Engineen, koska se käyttää niin sanottua visual scriptingiä, Unreal Enginen tapauksessa Blueprints Visual Scripting. Se on pohjimmiltaan samantyyppinen kuin esimerkiksi Unityn käyttämä C#-kieli, mutta sisältää omasta mielestäni käyttäjäystävällisemmän lähestymistavan. Kokeilin aloittaa työskentelyn Unitylla, mutta koin perinteisen tekstillä to teuttavan koodaamisen opetteluun vievän liikaa aikaa.

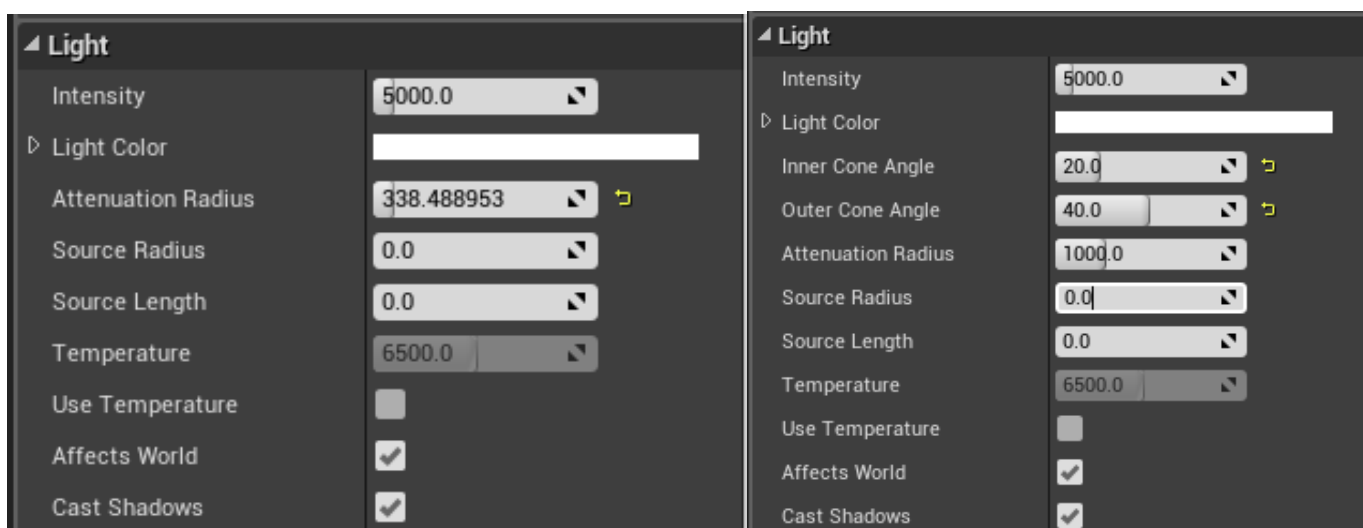
Unreal Enginen skriptaamisessa eri osien näkyvyys ja niiden yhteys ovat helpommin nähtävissä. Unreal Enginen omassa ohjesivustossa on paljon erilaisia tutoriaaleja, joilla pääsin alkuun, sekä erilaisilta forumeilta löytyi myös erittäin paljon valmiita esimerkkejä, joita sitten pystyin soveltamaan työssäni.

4.2 Valolähteet

Valon tuottaminen pelimoottorissa tapahtuu sen omilla valolähteillä, jotka ovat spotlight, pointlight, directional light ja skylight. Pelimoottorin valonlähteet ovat näkymättömiä pisteitä, mitkä säteilevät valoa pisteestä ulospäin. Jos tarkoituksena on luoda tilaan oikea valaisin, täytyy valaisimille luoda erillisellä ohjelmalla 3D-malli. 3D-mallintamisesta keron työssäni myöhemmin. Tilan valaistusta luodessa näistä vaihtoehdoista käytetään spotlighttia ja pointlightia, kun halutaan luoda valoa, joka säteilee ikään kuin valaisimesta. Kummallakin valolähteellä on omat asetuksensa, joita säätämällä haluttu tulos saadaan.

Point light on pistevalo, joka säteilee valoa tasaisesti joka suuntaan yhdestä pallon pisteestä. Sitä voi ajatella perinteisenä hehkulamppuna. Point light pystyy käyttämään valonjakokäyrätiedostoa IES, jolloin valo jakautuu sillä tavalla, kuin valaisimen valmistaja on sen ilmoittanut. Käytännössä siis, vaikka tilaan on luotu valonlähde, joka säteilee valoa kuin sen ympärillä olisi fyysinen valaisin, se ei välttämättä tarvitse ympärille mitään.

Spotlight on kohdevalo, joka säteilee valoa pisteestä kartiomaisessa muodossa. Kohdevaloon ei pysty asettamaan IES-tiedostoa ohjaamaan valoa. Vaikka kyseessä on kohdevalo, jos halutaan tehdä kohdevalo, joka pohjautuu oikeaan tuotteeseen, on parempi valita pistevalo, koska valaisimen IES-tiedostolla saadaan aikaan oikeanlainen valo. Kuvassa 6 näkyy Unreal Enginen valolähteiden asetuksia



Kuva 6. Pistevalon ja kohdevalon asetukset.

Intensity määrittelee valon kirkkauden. Pelimoottori käyttää kirkkauden arvona lumeneja. Pistevalolla ja kohdevalolla arvo 1 700 lm vastaa 100 W:n hehkulamppua. Muilla valolähteillä intensity tarkoittaa vain kirkkauden kerrointa.

Valon väriä voidaan säätää light color -palkista sekä temperature-säätimestä. Valon väriä säätämällä palkista voidaan mukaila esimerkiksi RGBW-valaisimia. Väriämpötilan säädöt numeerisesti ovat kelvineissä, kuten oikeissakin lamputissa. Oman kokemuksen mukaan tätä ei kuitenkaan kannata sokeasti noudattaa. Kerron näiden asetusten oikeanmukaisuudesta myöhemmin vertailuosioissa.

Attenuation Radius tarkoittaa vaikutusalueen kokoa. Tällä säädetään, kuinka pitkälle valolähteestä valo vaikuttaa. Kuvassa 7 nähdään vaikutusalueen säätämisen vaikutus, heistavalla ja tummalla pinnalla.



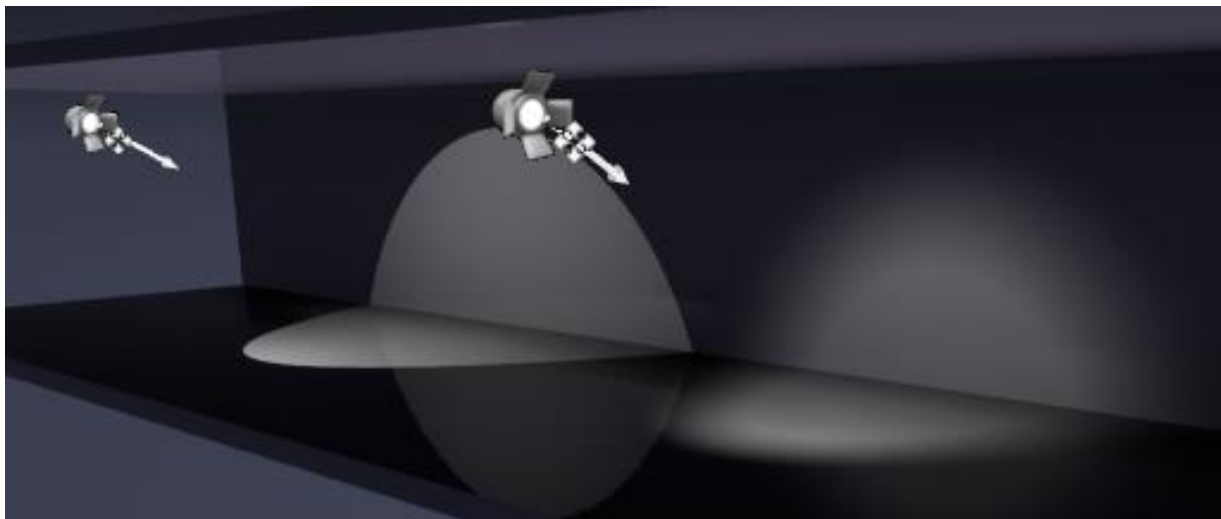
Kuva 7. Pistevaloja eri vaikutusalueen asetuksilla. Valo 1: 200, valo 2: 500, valo 3: 1000.

Source radius ja length tarkoittavat valolähteen sädettä ja pituutta. Kuvassa 9 havainnollistan pistevalon kokoasetusten merkitystä. Pituutta muuttamalla voidaan luoda esimerkiksi valolähde, joka muistuttaa loisteputkilamppua säteilemällä koko pituuden matkalta. Valon koon määrittely vaikuttaa myös sen heijastumiseen kirkkaista pinnoista. Tässä tapauksessa valonlähteen pituus vaikuttaa vain heijastukseen ja alueeseen mistä valoa säteilee. Kuvassa 8 nähdään, että valonlähde on käytännössä näkymätön, ellei sille erikseen luoda 3D-mallia



Kuva 8. Pistevaloja eri kokoasetuksilla. Valo 1: 20*20, valo 2: 0*0, valo 3: 10*30.

Inner cone angle ja outer cone angle ovat pelkästään kohdevalon asetuksia, jotka määrittävät lähtevän valon kulman ja reunojen pehmeiden. Jos sisäkartion ja ulkokartion kulmat ovat samat, on valon raja terävä. Kuvasta 9 näkee, miten kulmien vaihtaminen vaikuttaa ulkonäköön.

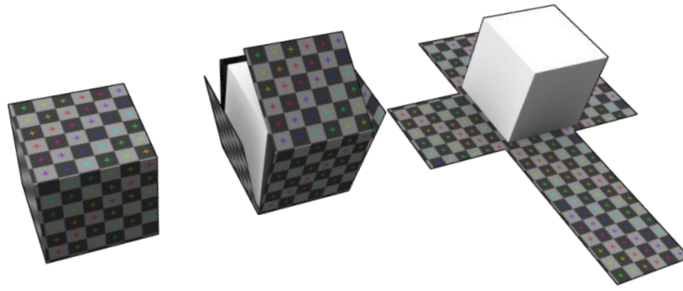


Kuva 9. Kohdevalon kartion kulmien merkitys. Valo 1: 30°, 30°. Valo 2: 15°, 30°.

Directional light ja skylight ovat ympäristön valolähteitä. Directional light toimii käytännössä auringon tai kuun valona. Skylight on ympäröivän taivaan tuottama tasainen valo. Nämä valot ovat työkaluja, joilla halutaan luoda todellisen näköinen ympäristö valaistus-suunnittelun visualisointiin.

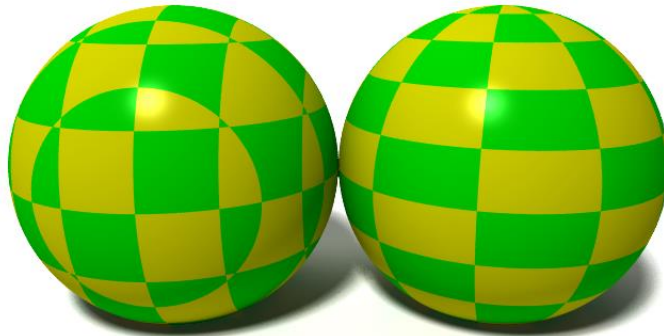
4.2.1 UV-mappaus

Kun 3D-objektille asetetaan 2D-kuva pintamateriaaliksi, tätä kutsutaan UV-mappaukseksi. Jos objektille halutaan teksturi tai jos se on staattisen valaistuksen vaikutusalueella, täytyy sille tehdä UV-mappaus. Kirjaimet UV-kertovat sen akselit kuten X ja Y, mutta nämä ovat jo käytössä 3D-objektin akseleissa. UV-teksturoinnin avulla 3D-objekti voidaan maalata jollain värillä tai kuvalla. Esimerkkinä (kuva 10) yksinkertaisen kuu-tion UV-mappaus.



Kuva 10. Kuution UV-mappaus. [10.]

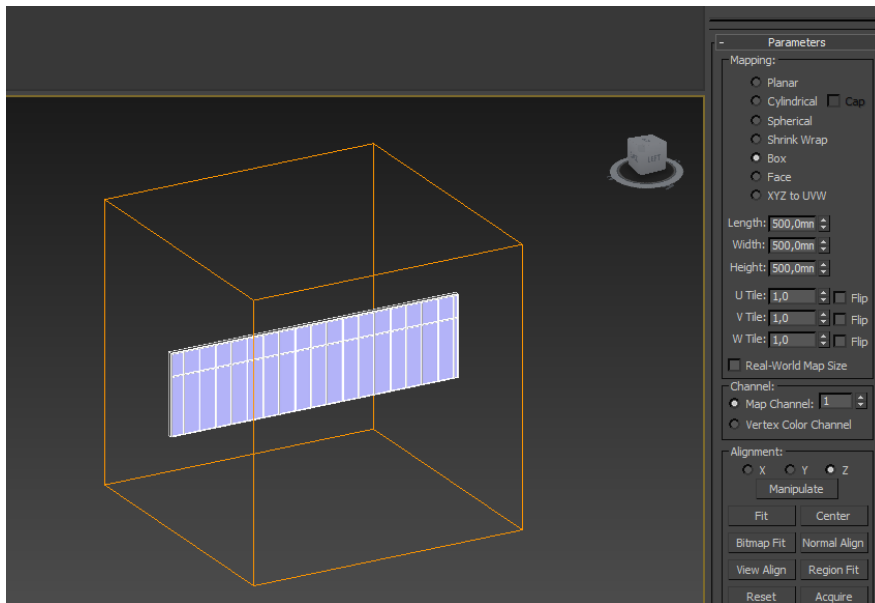
Toisena esimerkkinä voidaan tarkastella tilannetta, jossa pallo teksturoidaan ilman UV-mappausta sekä UV-mappauksen kanssa (kuva 11).



Kuva 11. Pallon teksturointi. Vasemmalla ilman UV-mappausta ja oikealla UV-mappauksen kanssa. [11.]

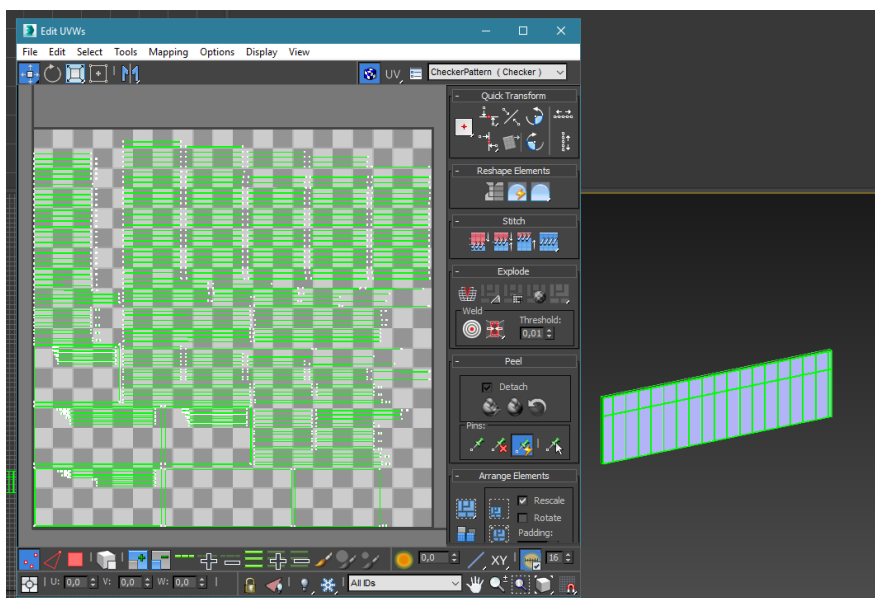
Kuten kuvan 11 vasemmalta puolelta näkyy, vaikka teksturoinnin voi tehdä ilman UV-mappausta, on lopputulos vääränlainen tehtäessä realistista materiaalia.

Pelimoottori tarvitsee objekteilleen, jotka ovat tekemisissä valaistuksen kanssa, kaksi UV-kanavaa. Käytännössä se tarkoittaa UV-mappauksen tekemistä kaksi kertaa. Toisen avulla saadaan tekstuurit oikeannäköisenä objektin pinnalle. Toinen on staattisen valaistuksen renderöintiä varten.



Kuva 12. 3dsMax. Ikkunoiden UV-mappaus.

Kuvassa 12 näkyy yksi ikkunapaneeli, jolle olen asettanut kuution (box) muotoisen UV-kartan. UV-kartan muoto riippuu muokattavan objektin muodosta. Pallolle asetetaan pallon (spherical) muotoinen UV-kartta ja esimerkiksi lattialle voidaan laittaa vain tasokartta (planar), koska siinä on kuitenkin näkyvissä vain yksi puoli.



Kuva 13. Ikkunaobjektin UV-unwrapping.

Kuvassa 13 on objekteille tehty UV-unwrapping. Tämä toiminto ottaa objektin jokaisen sivun ja levittää ne yhdelle tasolle. Tämän avulla pelimootorissa tehtävä valaistus ja varjot saadaan näkymään oikealla tavalla. Tässä kohdassa on tärkeää asettaa näiden tasojen välille tarpeeksi iso etäisyys, koska muuten esimerkiksi seinien nurkista saattaa vuotaa valoa läpi.

5 Vertailu

5.1 Tavoite

Valaistussuunnittelijoiden yksi käytettävissä oleva ohjelma Rambollilla on Dialux Evo, jolla voi tehdä valaistuslaskelmien lisäksi valaistussimulointeja. Tarkoituksena tässä osiossa on tutkia, kuinka lähelle Dialux Evon tuloksia, ulkonäöllisesti, päästään Unreal Engineä käytettäessä. Tavoitteena on tarkastella ensinnäkin visuaalisia tuloksia, mutta myös sitä, kuinka suuren työn takana on tuottaa yksinkertainen muutaman erilaisen valaistustilanteen luominen Unreal Enginellä. Olen alkanut opettelemaan pelimoottorin käyttöä sen jälkeen, kun tämän idean opinnäytetyöhön sain, eli vain muutama kuukausi sitten. Aion tehdä tässä vertailussa kaikki työn vaiheet itse, koska haluan työssäni tuoda esille sitä, että jos pelimoottori otetaan käyttöön yrityksessä yhtenä valaistussuunnittelun työkaluna, saadaan tietää, kuinka suuren työn takana on suunnittelussa tarvittavan osaamisen saavuttaminen. Samasta syystä teen valaistuksen rakentamisen Unreal Enginessä vaihtamatta mitään asetuksia, jotka vaikuttavat siihen miltä valaistus näyttää, jotta nähdään, mihin tuloksiin päästään mahdollisimman nopealla tavalla.

5.2 Kohteet ja valaisimet

Käytän vertailussa kahta erilaista tilaa. Ensimmäinen tila on yksinkertainen huone, jossa aion vertailla kolmea erilaista valaisinta. Toinen tila on halli, jonne sijoitan tehokkaampia valaisimia mutta jätän myös jonkun osan valaitsematta, jotta vertailua voidaan tehdä mahdollisimman tarkasti. Kummassakin tilassa valaisimiksi valitsin vain sellaisia vaihtoehtoja, joista on saatavilla ULD-tiedosto, IES-tiedosto sekä Revitissä avattava RFA-tiedosto tai muu 3D-tiedosto. Lisäksi valitsin tuotteet, joiden valonjakotiedostot ovat jollain tavalla erotettavissa, jotta tuloksia voitaisiin vertailla tarpeeksi hyvin. Kaikki tuotteet ovat Glamoxin valaisimia, koska oman kokemukseni perusteella hyvin usealla Glamoxin tuotteella on tässä vertailussa tarvittavat tiedostot.

ULD-tiedosto on Dialuxin käyttämä tiedostomuoto, jonka avulla valaisimen ulkomuoto, tekniset tiedot sekä valonjako saadaan ohjelmaan. IES-tiedosto on valaisimen valonjakokäyrätiedosto, jota aion käyttää pelimoottorissa. En aio muokata 3D-malleja täysin realistisen näköisesti, pintamateriaaleja ajatellen, koska se ei ole tämän tutkimuksen tarkoitus. Tarkoituksena on vertailla vain valaisimen valon vaikutusta tiloihin.

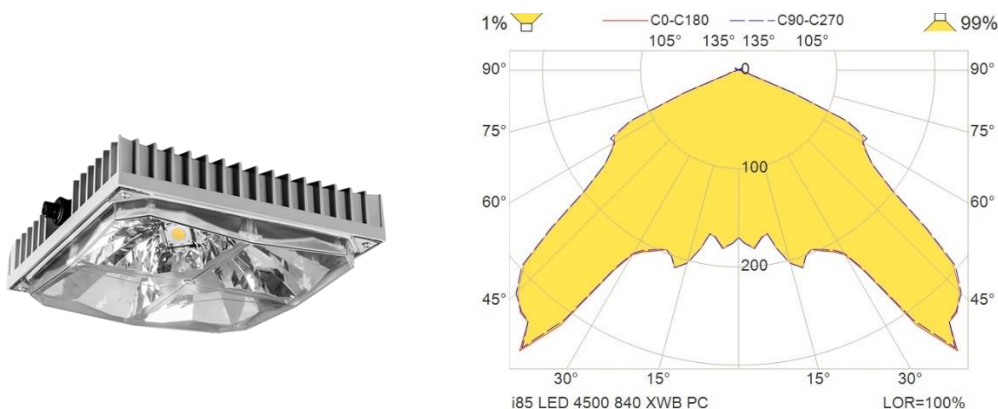
Ensimmäisenä tilana toimii Autodesk Revitissä tehty yksinkertainen 5 m * 5 m * 2,7 m -kokoinen tila. Pintamateriaalin valitsen kummassakin ohjelmassa erikseen, mutta kuitenkin tarkoitus on saada niistä mahdollisimman samanlaiset. Teen tiloista Revitissä FBX-tiedoston sekä IFC-tiedoston. FBX-tiedostoa muokkaan 3dsMaxissa, koska suoraan vietynä Unreal Engineen FBX-tiedosto näyttäytyy väärän kokoisena. Kerron työssäni myöhemmin tarkemmin, mitä muokkauksia täytyy tehdä, jotta tilaa voi hyödyntää Unreal Engineessä. Dialux Evo puolestaan pystyy käyttämään Revitistä vietyä IFC-tiedostoa suoraan. Tällä tavalla tilat ovat samanlaiset kummassakin ohjelmassa, ja tulokset ovat vertailukelpoisia.

Teen valoista Unreal Engineessä liikkumattomia (static), koska silloin ohjelma pystyy laskemaan, kuinka valo heijastuisi huoneen pinnoista realistisesti. Käytettäessä liikuteltavia (movable) valoja pystyisin pelaamisen aikana vaihtamaan esimerkiksi valon sijaintia. Tässä vertailussa tuota ominaisuutta ei kuitenkaan tarvita. Ongelmaksi liikuteltavasta valosta tulee kuitenkin se, että Unreal Engine ei kykene luomaan realistisia heijastuksia tai epäsuoraa valoa liikkuvista valoista, joten ne kohdat mihin valo ei suoraan valaise, jäävät tummaksi. Siitä pääsee osittain eroon käyttämällä skylightia tai sijoittamalla ylimääräisiä valolähteitä tilaan simuloimaan heijastunutta valoa. Kunnollisen vertailun takia pysyn siis liikkumattomissa valoissa.

Unreal Engineessä tulen käyttämään kaikkina valolähteinä pistevaloa (point light), koska se on ainoa valomuoto, joka tulee valonjakotiedostoja. Siinä missä Dialux Evo säteilee valoa suoraan siinä muodossa, joka on kyseiselle valaisimelle määritetty, on Unreal Enginen tapa erilainen. Pistevalo, joka säteilee valoa joka puolelle tasaisesti, käytännössä "peitetään" osittain, jotta valo jakautuu sillä tavalla kuin valonjakokäyrä sen määrää. Tulos on siis hyvin samanlainen kummallakin ohjelmalla, mutta tekniikka eri.

5.2.1 Valaisin 1

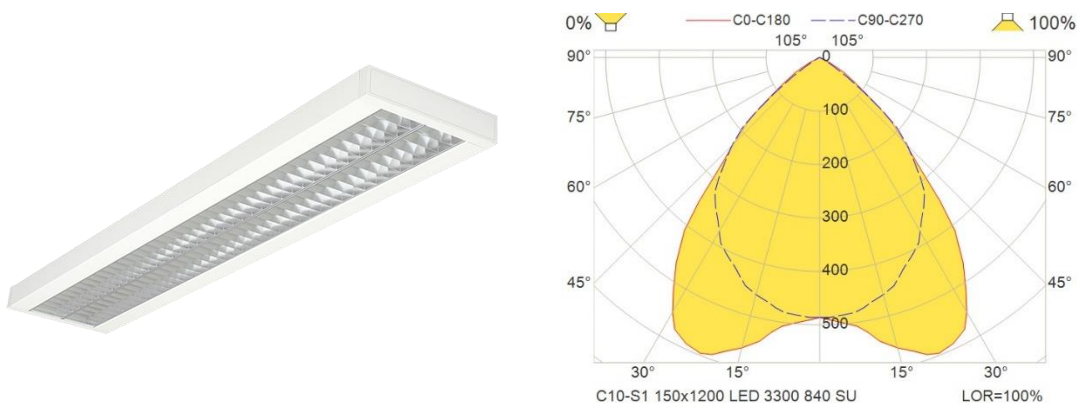
Ensimmäinen valaisin on Glamoxin i85 LED 4500 840 XWB PC. Kyseessä on pinta- tai ripustusasenteinen LED-valaisin, jonka esimerkkikohteiksi luetellaan paikoitushallit, katonkaset ja muut vaativat ympäristöt. Valaisimella on mielestäni erottuva valonjakokäyrä, joten se soveltuu hyvin tähän vertailuun. (Kuva 14.)



Kuva 14. Glamox i85 ja sen valonjakokäyrä [12].

5.2.2 Valaisin 2

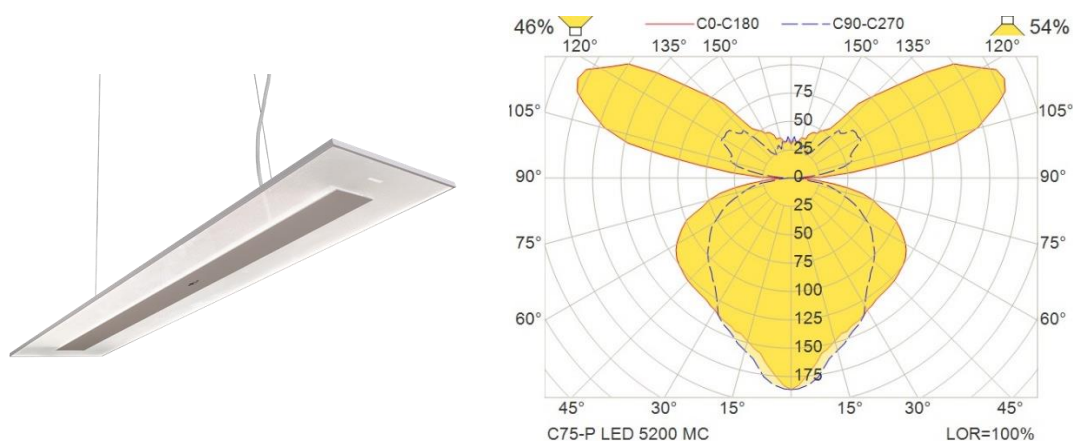
Toinen valaisin on Glamoxin C10-S1 150x1200 LED 3300HF 840 SU. Valaisin on hyvin perinteinen pinta-asenteinen LED-putkivalaisin. Valitsin tämän valaisimen vertailukohteeksi, koska Unreal Enginellä ei voi käyttää IES-valonjakotiedostoa loisteputkivalaisimen tyypisessä valolähteessä, jossa valolähde on pitkä. Unreal Enginessä IES-tiedostoa käytettäessä on valonlähde valittava pistevaloksi, joka valaisee yhdestä pisteestä.



Kuva 15. Glamox C10 ja sen valonjakokäyrä. [13.]

5.2.3 Valaisin 3

Kolmas valaisin on Glamoxin C75-P1290 LED 5400 HF 830 PRE CP2 MC. Kyseinen valaisin on tämän testin erikoisin, koska valaisinta ympäröi lasilevy, joka on läpinäkyvä, kun valot ovat pois päältä. LED-valo ohjataan keskeltä valaisinta vaakasuoraan kohti lasilevyn reunoja. Tällä valaisimella on myös omalaatuinen valonjakokäyrä. Valonjakokäyrä näyttää, että iso osa valosta säteilee valaisimen yläpuolelta, epäsuoraa valaistusta varten. (Kuva 16.)



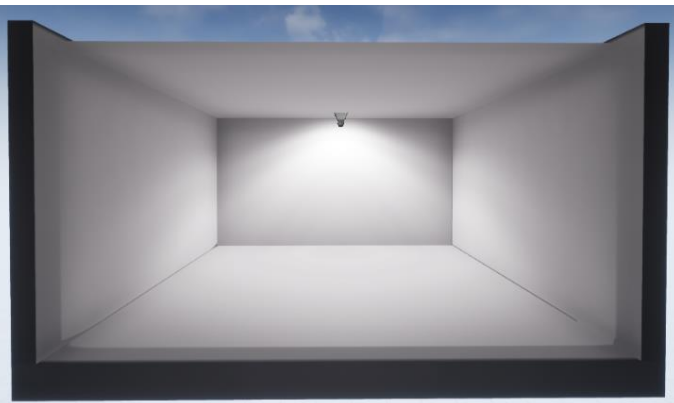
Kuva 16. Glamox C75 ja sen valonjakokäyrä. [14.]

5.3 Tulokset

Kuva 17 on Dialux Evosta otettu kuvakaappaus, jossa näkyy valaisin ja sen valaisema alue. Dialux Evo simuloi valaisimesta lähtevän valon siinä muodossa kuin valonjakokäyrä sille kertoo, ottaa huomioon seinä- ja lattiamateriaalien heijastusarvot ja sen avulla simuloi heijastuvan valon ympäri huonetta. Toisessa kuvassa näkyy Unreal Enginestä tehty kuvakaappaus. Valaisinta en näitä vertailukuvia varten mallintanut. Vertaillen pelkkiä valonjakvoja näiden ohjelmien välillä on nähtävillä, että vaikka kummassakin tulokset ovat erotettavissa, on eroavaisuuksia selvästi. Dialuxin valon jakautuminen on tervämpää, ja sen vaikutusalue on hieman pienempi. Unreal Engineissä valo jakautuu tasaisemmin ja vaikuttaa valaisevan myös hieman enemmän. Unreal Engineissä tila on ylipäättänsä enemmän valaistu. (Kuva 18.)

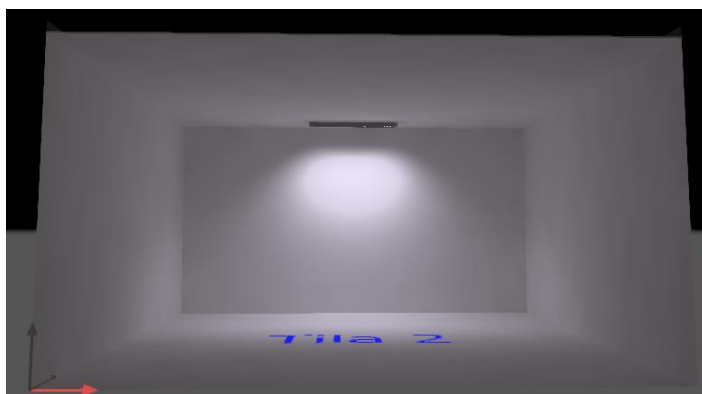


Kuva 17. Tila 1, Dialux Evo

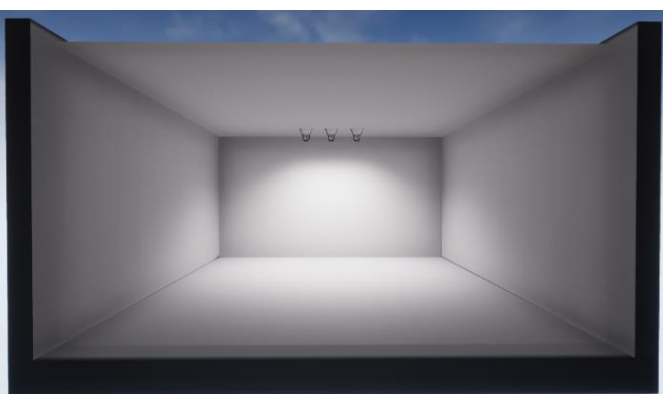


Kuva 18. Tila 1, Unreal Engine

Toisessa valaistustilanteessa (kuva 19, 20) pitkälaisen muodon saavuttamiseksi sijoitin kolme valolähdettä tasaisin välein, jotta valon muoto vaikuttaisi oikealta. Tässä tapauksessa Unreal Enginen valojen tehot täytyy pudottaa kolmannekseen, koska määrä oli kolminkertainen. Ilman tätä olisi valo ollut liian kirkas. Lopputulokset näyttävät tässäkin samanlaisia eroja näiden ohjelmien välillä. Valon jakautuminen on samanmuotoista, Dialuxin valon vaikutus vaikuttaisi hälvenevän nopeammin. Heijastuvan valon määrä näyttäisi kuitenkin olevan kummassakin ohjelmassa lähellä toisiaan. Tilojen katot ovat valaistuna samalla tavalla.



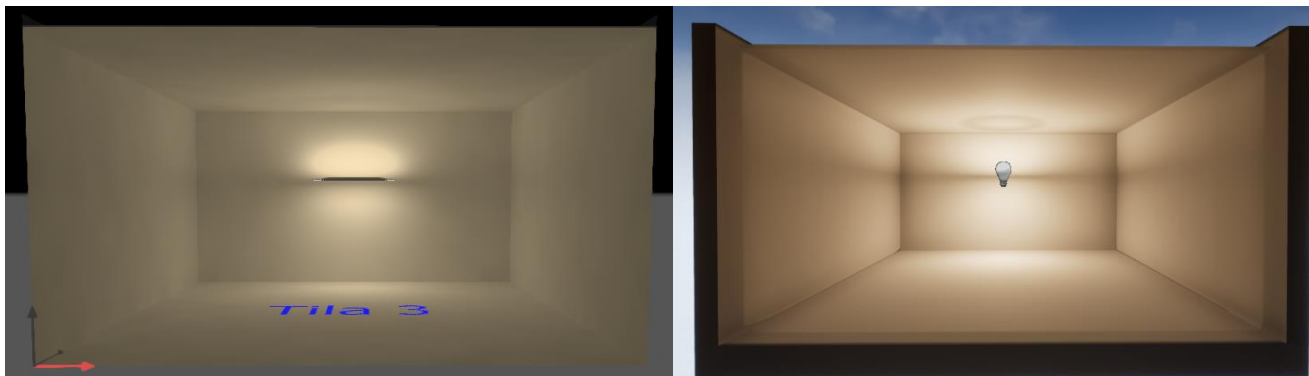
Kuva 19. Tila 2, Dialux Evo



Kuva 20. Tila 2, Unreal Engine

Kolmannen valaistustilanteen (kuva 21, 22) tulokset kertovat samaa tarinaa, vaikka valaisin on epäsuorasti valaiseva ja valonjako eroaa edellisistä vertailuvaloista paljon. Tässä tapauksessa oikean valon värin saavuttamiseksi täytyi Unreal Enginessä muokata värilämpötilaa. Asettaessani valolle valmistajan ilmoittaman 3000 K-arvon tuli

valaisimesta lähes täysin keltaista valoa säteilevä. Kuvassa näkyvässä tilanteessa valolle on asetettu värilämpötilaksi 4000 K, joka normaalisti tarkoittaisi kylmää valkoisempaa valoa. Tämä täytyy ottaa huomioon Unreal Engine:ssä, jos ollaan tekemisissä värilämpötilan kanssa.



Kuva 21. Tila 3, Dialux Evo

Kuva 22. Tila 3, Unreal Engine

Hallille laskin Dialux Evossa valaistusvoimakkuudeksi lattiatasossa 500 lx, minkä jälkeen poistin yhdestä nurkasta neljä valaisinta, vain jotta saisin aikaiseksi hieman enemmän eroavaisuutta tätä vertailua varten. (Kuva 23).



Kuva 23. Tila 4, Dialux Evo



Kuva 24. Tila 4, Unreal Engine

Tässä tapauksessa näkyvät nyt hyvin aiemmissakin valaistustilanteissa esiin tulleet erot. Unreal Enginen tila on taas enemmän valaistuu, mutta valon muoto ja kontrastit suoraan valaistujen pintojen ja epäsuorasti valaistujen pintojen välillä ovat nähtävillä hyvin. Myös neljän poisjätetyn valaisimen jättämä tummempi alue erottuu hyvin. (Kuva 24.)

5.4 Tulosten pohdinta

Vaikka ohjelmien välillä on eroja, mielestäni Unreal Engine pystyy simuloimaan valoa sen verran hyvin, että ohjelmalla saadaan aikaiseksi hyviä visualisointeja, kohtuullisessa ajassa. Täysin realistisia tuloksia visuaalisesti ei kummallakaan ohjelmalla saada, joten nämä esiin tulleet eroavaisuudet ovat mielestäni siedettäviä. Vertailtavien tilojen tekeminen kesti ajallisesti Unreal Enginellä hieman kauemmin, koska itse olen enemmän käyttänyt Dialuxia töissä ja koulussa ja sen toiminnot ja mahdollisuudet ovat helpommin löydettävissä itselleni. Vaadittavia toimenpiteitä testissä käytettävien tilojen aikaansaamiseksi ei kuitenkaan ole kovin paljoa Unreal Enginessä, jotta testissä saavutettavat tulokset saatiin. Unreal Enginen sisäisiin asetuksiin en myöskään tehnyt mitään muutoksia, vaan aloin suoraan rakentamaan tiloja ja valaistusta.

6 Peli

6.1 Tavoite

Opinnäytetyöni toisena Unreal Enginellä tehtynä työnä oli tarkoitus luoda Rambollin uuteen pääkonttoriin sijoittuva peli. Tavoitteena oli näyttää, mitä pelimoottorilla voi tehdä valaistussuunnittelun ja valaistussimuloinnin osalta. Pelin käyttämä tila oli jo aikaisemmin luotu Revitissä Rambollin toimesta. Toin tilat Revitistä 3dsMaxin kautta Unreal Engineen, jossa muokkasin tilat loppuun. Tämän jälkeen lisäsin valoihin ja ympäristöön interaktiivisuutta, kuten valojen kytkemisen päälle ja pois, niiden värin vaihtamisen, valojen liikuttamisen tilassa ja ajan vaihtamisen päivästä iltaan. Tarkoituksena ei ole tehdä täysin fotorealista tilaa, vaan esitellä valaisimia ja niiden ominaisuuksia sekä valaistustapaa. Halusin myös tuoda esille pelimoottorin mahdollistamia asioita, jotka eivät ole olleet mahdollisia perinteisillä valaistussimuloinnin mahdollistamilla ohjelmilla kuten Adobe Photoshop tai 3dsMax.

Tuon tässä osiossa lisäksi esille haluttujen tulosten saavuttamiseen vaadittavan vaivan ja ajan. Tiedän jo sen, että Unreal Enginellä on mahdollista tehdä valaistussimulointia pelkästään sen takia, että olen pelannut monia pelejä, joissa valaistukseen on kiinnitetty paljon huomiota. Jotta Unreal Engineä voitaisiin harkita valaistussuunnittelun yhdeksi työkaluksi, sillä täytyy olla mahdollista tehdä valmis asiakkaalle esiteltävä peli, järkevällä aikataululla. Näen itseni vielä aloittelevana Unreal Enginen osaajana, joten tämä voi toimia hyvänä mittarina, sille kauanko tämän tyyppiseen työhön menee aikaa.

Jokainen valaistusratkaisu, johon päädyin tätä peliä luodessa, on minun itseni ideoima, ja ratkaisut eivät vastaa sitä, mitä Rambollin uuden pääkonttorin valaistusratkaisut tulevat oikeasti olemaan. En ole kyseisessä projektissa mukana missään muodossa. Käytin vain pelin runkona uuden pääkonttorin tiloja, koska ne olivat helposti saatavilla ja hyvä kohde, jossa voiesitellä pelimoottorin tuomia mahdollisuuksia.

6.2 Pelin tilojen luominen

Sain käyttöni Rambollin tulevan pääkonttorin Revit-tiedoston ja valmiita 3D-malleja, jotka sisälsivät ulkokuoren lisäksi myös sisätilojen seiniä sekä kalusteita. Käytin näitä hyväkseni tässä työssä. Rakennuksen pääkerroksessa on vastaanottotiskin vieressä tila, joka sisältää neuvotteluhuoneita sekä eräänlaisen mediahuoneen. Kaikki tilat täytyi tuoda ulos Revitistä FBX-tiedostona, jota siten muokkasin 3dsMaxilla, jotta ne soveltuvat Unreal Engineessä käytettäväksi.

6.3 Aula

Rakennuksen aula on avoin tila, jossa 6. kerroksen kattorakenne on suuri ikkunaristikko, joka päiväsaikaan valaisee aulassa sijaitsevaa vastaanottotiskiä ja avointa oleskelutilaa. Jätin tilojen kerrokset tyhjäksi tässä tapauksessa, koska tarkoitus on ainoastaan esitellä valaistusta, joka on nähtävissä vain aulan ensimmäisestä kerroksesta katsottaessa. Otan mallia kerrosten valaistuksessa Rambollin Tanskan pääkonttorista, jossa avoimen tilan reunoissa, kerrosten alakatossa, on rivissä upotettuja valaisimia. Nämä valaisimet ovat upotettua Glamoxin C80-mallia. Aion tehdä näistä valoista staattisia, koska suuri määrä liikuteltavia valoja vaatisi tietokoneelta niin paljon lisätehoa, että se ei olisi kannattavaa. Pohjakerroksen valaistusta varten kehitin itse 3dsMaxilla kaksi suurta, hie-man toisistaan eroavaa riippuvalaisinta sekä ylös ja alas liikkuvia putkivalaisimia, joilla voi esitellä liikkuvaa ja vaihtuvaa valoa. Nämä erilaiset aulan valaistusvaihtoehdot ovat nähtävissä päiväsaikaan auringonvalossa sekä ilta-aikaan pelkän kuuvalon valossa. Valaistusvaihtoehtoja sekä päivä- ja iltatilannetta voi vaihdella nappia painamalla, miten haluaa. Ohjeet näkyvät koko ajan ruudulla pelattaessa, kuten kuvasta 25 näkee.

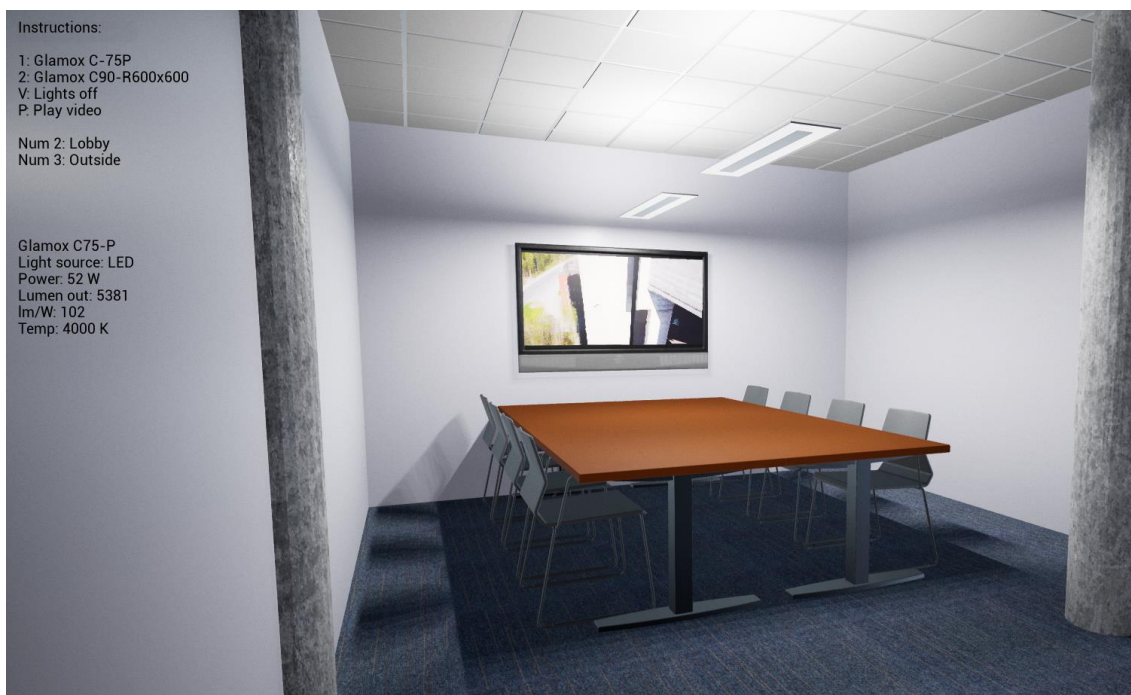


Kuva 25. Kuvakaappaus pelistä. Aula.

Reunavalaisimet on asetettu liikkumattomiksi (static), koska en aio vaihdella niitä mitenkään pelin aikana. Tällä tavalla saadaan myös realistisin tulos. Muut aulassa käytettävät valot on asetettu liikkuviksi (movable), koska halusin nimeomaan vaihdella niitä lennosta edestakaisin. Voisin tehdä niistäkin liikkumattomia, mutta siinä tapauksessa, jos haluaisin esitellä eri vaihtoehtoja, pitäisi minun rakentaa samasta tilasta niin monta eri kenttää kuin vaihtoehtojakin, ja kenttien vaihtelu on hieman hidasta ja kömpelöä tähän tarkoitukseen.

6.4 Neuvotteluhuone

Neuvotteluhuoneen pohjaratkaisu tulee suoraan arkkitehtipiirustuksista, jotka toin Revitistä 3dsMaxin kautta Unreal Engineen. Tila on melko pieni, joten tähän tilaan en lähtenyt suunnittelemaan mitään kovin erikoista valaistusta. Käytin tässä tilassa kahta erilaista valaistustilannetta. Ensimmäinen vaihtoehto on alakattoon asennettava moduulivalaisin. Valaisin on Glamoxin C-90 OD, jossa on heijastimena pudotettu opaali. Kattoon, neuvottelupöydän päälle, sijoitan riippuvalaisimen. Valaisin on Glamoxin C75-P, jonka valaisinosa on läpinäkyvä, kun se on pois päältä. Valaisin on sama riippuvalaisin, jota käytin aiemmin vertailussani valaistustilanteessa 3. Koska kummassakin valaisimessa on ominaisuuksia, jotka näkyvät vain, kun ne ovat päällä tai poissa, ovat ne myös aiemmalla tavalla kytkettävissä pois tai päälle napin painalluksella. (Kuva 26.)



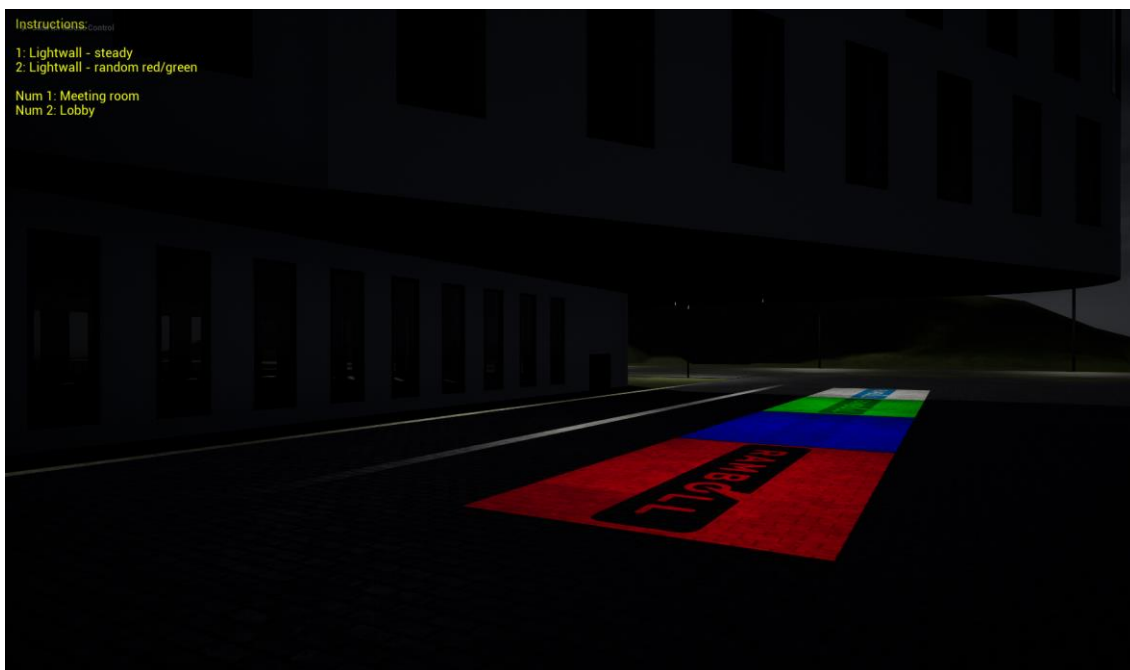
Kuva 26. Kuvakaappaus pelistä. Neuvotteluhuone.

Lisäsin neuvotteluhuoneeseen kalusteiden lisäksi myös television, jossa voidaan pelin aikana pyörittää videota ja ääntä napin painalluksella. Tällä ei varsinaisesti mitään tekemistä valaistuksen kanssa ole, mutta halusin tuoda esille vähän lisää asioita, joihin pelimoottori helposti kykenee.

6.5 Ulkotila

Viimeisenä kenttänä tässä pelissä on pääkonttorin ulkotila. Tähän tilaan en suunnitellut mitään oikeaa tilannetta muistuttavaa valaistusta, vaan halusin tuoda esille muutaman asian, joita en saanut toteutettua niin sanottuihin oikeisiin tiloihin. Ensimmäinen asia, jonka pelaaja näkee, on maata valaisevat valoviivat. Nämä valoviivat on toteutettu samantyyppisellä tavalla kuin aiemminkin nähdyt ”oikeat” valot, mutta sen sijaan, että käytäisin IES-tiedostoja määrittelemään valonjakoa, käytin Unreal Engineissä löytyvää Light Functionia. Light function eli valofunktio on käytännössä materiaali, joka rajoittaa valon voimakkuutta. Tässä tapauksessa valofunktion materiaalina on musta kuva, jonka keskellä kulkee terävä valkoinen viiva. Musta väri kuvassa kertoo valolle, että siinä kohdassa

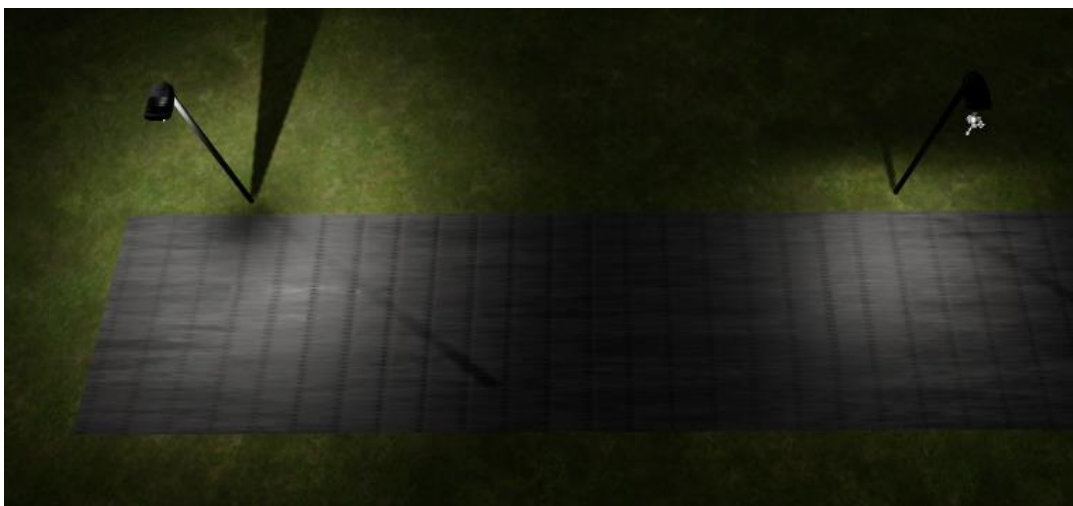
ei pidä olla valoa ollenkaan ja valkoinen on valon täysi teho. Tätä voidaan käyttää esittämään, esimerkiksi liikkeen logoa maahan heijastettuna.



Kuva 27. Kuvakaappaus pelistä. Ulkotila, valofunktioiden käyttö.

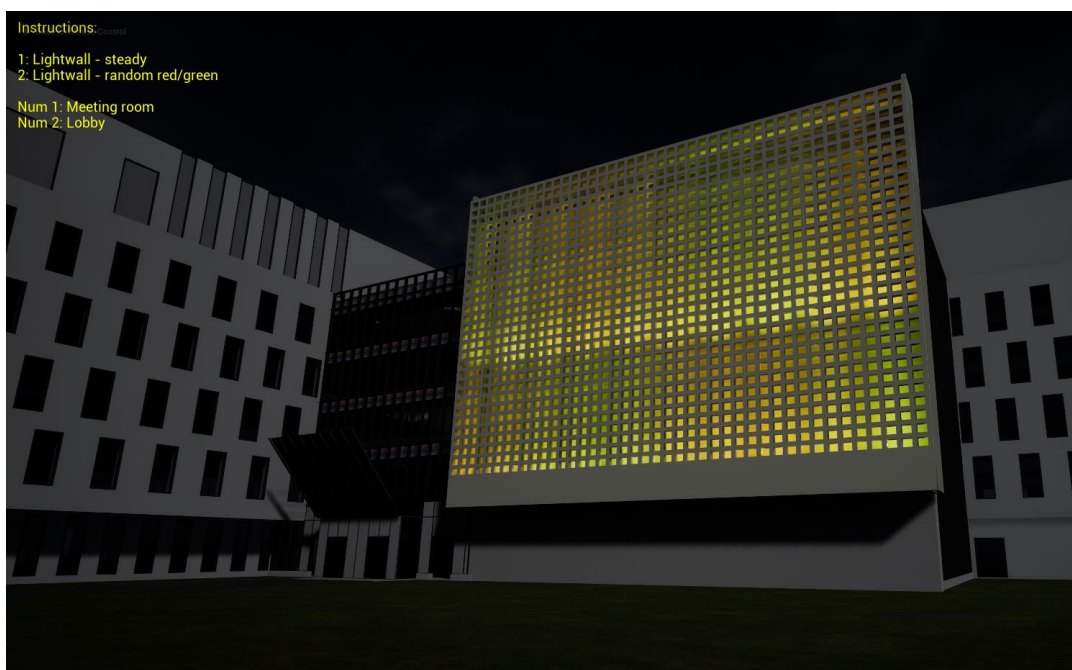
Toinen valofunktio on Rambollin logo heijastettuna maahan kolmella eri värillä, punainen, vihreä ja sininen sekä yksi valo, jossa logo näkyy oikeanlaisena. Tässä halusin vain havainnollistaa sitä, miten Unreal Engine käsittelee valoa. Jos todellisuudessa nämä kolme värivaloa yhdistäisi, tulisi siitä valkoista valoa, kuten tässäkin tapauksessa. Tein kolme eri värivaloa siten, että monistin identtisen valolähteen, jonka valofunktiomateriaalina oli rambollin logo ja otin jokaisesta käyttöön vain yhden värin kanavan. Lopullinen oikean näköinen logo on pelkästään näiden kolmen valon yhdistelmä. (Kuva 27.)

Seuraavassa kentässä on esiteltynä muutama ulkovalaisin. Pienempien puistovalaisimien valaisema alue on pyöreä, joten pystyin käyttämään niissä pistevaloa ja IES-tiedostoa. Ongelmia tuli isompien tievalaisimien kanssa. Kyseessä olevat tievalaisimet valaisevat epäsymmetrisesti valoa eteenpäin, joten valaisimen taakse jäävä alue jää pimeäksi. Pistevalo valaisee vain täysin pyöreän alueen, vaikka siinä on käytetty saman valaisimen IES-tiedostoa. Tämän takia tämäntyyppisissä valaisimissa ei saada samalla tavalla oikeanlaista tulosta. Tämä on vielä puute Unreal Engineissä. Asian pystyy kiertämään käyttämällä valofunktiota, jossa peitetään taakse heijastuva valo. (Kuva 28.)



Kuva 28. IES-tiedoston ja valofunktion ero katuvaloissa

Viimeisen valaistuskohteen malliin on otettu esimerkkiä kauppakeskus Ainoasta, jossa ulkoseinällä on seinäristikko, jonka takana olevaa seinää valaisee RGBW-valaisimet. Halusin simuloida kyseisen valaistustavan, koska kyseessä olevassa valoseinässä valojen värit vaihtelevat ja se näyttää aina hieman erilaiselta, riippuen mistä kuvakulmasta katsoo. Tätä asiaa ei pysty tuomaan kovin helposti esiin pelkkien kuvien avulla. Pelissä olevia valoja voidaan ohjelmoida käyttäytymään samalla tavalla kuin oikeassakin tilanteessa. Tähän peliin kuitenkin laitoin valot vain vaihtamaan tasaisesti väriä satunnaisessa järjestyksessä. (Kuva 29.)



Kuva 29. Kuvakaappaus pelistä. Ulkotila, väriä vaihtava valoseinä

7 Loppupäätelmät

Pelimoottorin käyttö valaistussuunnittelun visualisoinnissa tuo tiettyihin tilanteisiin hyvän lisämausteen. Suunniteltaessa tiloja joihin tulee perinteisiä valaistusratkaisua, kuten upotettuja valaisimia, jotka eivät esimerkiksi vaihda väriä tai liiku, pelimoottorin käyttö yksinään ei tuo siihen mitään uutta. Se olisi vain lisää aikaa vievä työkalu, josta ei voida tuoda ulos valaistusteknisiä asioita, kuten valaistusvoimakkuus tietyssä kohdassa.

Tekemäni peli on hyvin perinteinen ja yksinkertainen asia, jonka pelimoottorilla voi tehdä. En esimerkiksi tehnyt siihen ajanpuutteen takia mahdollisuutta virtuaalilasien käyttöön. Virtuaalilasien käyttö toisi vielä uuden ulottuvuuden, kun asiakas voisi tarkkailla asioita siten kuin ne olisivat nenän edessä. Pelini on myös hieman raskas, joten se vaatii tietokoneelta jonkin verran tehoja pyöriäkseen kunnolla. Tästäkin asiasta olisin saattanut päästä eroon, mutta tämän opinnäytetyön tekemisen aikatauluun sen ongelman ratkaisu ei riittänyt. Peliä tehdessä tuli myös vastaan ongelmia, jotka varsinkin valaistussuunnittelun osalta ovat tärkeitä, kuten IES-tiedoston hieman vääränlainen käyttäytyminen. Tästä kerroin peli-osion ulkotilakohdassa.

Suunniteltaessa esimerkiksi suurta kohdetta, jossa vielä liikutaan ideointiasteella valaistusratkaisua miettiessä ja halutaan pohtia monia eri valaistustapoja, toimii pelimoottori hyvänä lisätyökaluna. Hyvännäköisten 3D-mallien luominen suunniteltavasta tilasta on aikaa vievää, mutta kun ne kerran ovat tehtynä on erilaisten valaistusratkaisujen luominen pelimoottorilla nopeaa. Vaikka kohteeseen pohdittaisiin monimutkaisilta tuntuvia ratkaisuja, kuten liikkuvia valaisimia tai väriä vaihtavia valoseiniä, on niiden visualisoiminen mahdollista ainoastaan pelimoottorissa. Kun kohteessa voi lisäksi liikkua itse ja tarkkailla valaistusta eri kuvakulmista, tämä tuo se ihan uuden näkökulman asiakkaalle, mutta myös suunnittelijalle. Hyvänä esimerkkinä on pelissäni ulkotilaan luotu, väriä vaihtava valoseinä. Halusin saada vertailtavaksi oikeasti toteutetun valaistustilanteen ja mielestäni onnistuin siinä hyvin. Sen tyyppisissä tilanteissa pelimoottori toimii erittäin hyvin. Kun päästään haluttuun ratkaisuun ulkonäöllisesti, voidaan käyttää esimerkiksi Dialux Evoa, jotta saadaan standardien mukainen valaistustaso.

Ajallisesti pelimoottori on myös hyvin vartenotettava työkalu valaistuksen osalta. Kukaan ei haluaisi käyttää pelimoottoria valaistussuunnittelussa, jos yhden kohteen luomiseen menee kuukausia. Tekemässäni pelissä valaistusten luominen kaikkiin eri tiloihin kesti noin viikon verran työpäiviä ajatellen. Siihen sisältyi ideointi, mallien hakeminen

valmistajien internetsivuilta, niiden muokkaaminen 3dsMaxissa valmiiksi pelimoottoria varten, tuominen pelimoottoriin sekä kaiken muokkaaminen sellaiseksi, että sitä on helppo asiakkaankin käsitellä ja ymmärtää pelatessa. 3D-mallin muokkaaminen käytettäväksi oli ehdottomasti eniten aikaa vievä. Siihen käytettyä aikaa oli vaikea mitata, koska aloitin käytännössä jokaisen ohjelman opetteluun täysin alkutekijöistä. Sain myös 3D-mallit osittain valmiina, joten sekin vääristää sitä aikaa mitattaessa. Ajankäyttöongelmat 3D-mallien osalta johtuivat tässä opinnäytetyössä siitä, että tein kaikki loput työt yksin ja aloitin asioiden opetteluun täysin alusta. Muitakin ongelmia toki vielä jää ratkaistavaksi, mutta koska varsinkin Unreal Engineä käytetään koko ajan enemmän ja enemmän arkkitehtuurisessa suunnittelussa, tulee niihin varmasti ratkaisu tulevaisuudessa ohjelmiston päivitysten myötä. Olen kuitenkin sitä mieltä, että pelimoottori on hyvä lisätyökalu valaistussuunnitteluun ja sen visualisointiin.

Lähteet

- 1 Myhrberg, Tero. 2016. 3ds Maxin käyttö arkkitehtuurisessa valaistussuunnittelussa. Insinööriyö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.
- 2 Ward, Jeff. 2008. What is a Game Engine. Verkkoaineisto. Game Career Guide. <https://www.gamecareerguide.com/features/529/what_is_a_game_engine.php>. Luettu 25.11.2017.
- 3 Game Creation System. 2018. Verkkoaineisto. Wikipedia. <https://en.wikipedia.org/wiki/Game_creation_system>. Päivitetty 13.02.2018. Luettu 25.11.2017.
- 4 Game: Pinball Construction Set (Apple II). 2013. Verkkoaineisto. Mobygames. <<http://www.mobygames.com/game/apple2/pinball-construction-set/screenshots/gameShotId,617241/>>. Luettu 25.11.2017
- 5 Game Engine. 2018. Verkkoaineisto. Wikipedia. <https://en.wikipedia.org/wiki/Game_engine>. Päivitetty 27.02.2018. Luettu 25.11.2017.
- 6 Laird & Jamin. 2003. The Game Development Process. Verkkoaineisto. Worcester Polytechnic Institute. <<https://web.cs.wpi.edu/~id111x/c05/slides/intro.pdf>>. Luettu 25.11.2017.
- 7 Game: The Ultimate Doom (DOS). 2016. Verkkoaineisto. Mobygames. <<http://www.mobygames.com/game/ultimate-doom/promo/promolmaged,23936/>>. Luettu 25.11.2017
- 8 Parker, Vince. 2017. Passengers. Verkkoaineisto. <<https://www.vprofx.com/blog/passengers>>. Luettu 25.11.2017
- 9 Archviz Interior. 2014. Verkkoaineisto. <<https://forums.unrealengine.com/development-discussion/architectural-and-design-visualization/23268-archviz-interior>>. Luettu 25.11.2017
- 10 A representation of the UV mapping of a cube. The flattened cube net may then be textured to texture the cube. UV-mapping. 2018. Verkkoaineisto. Wikipedia. <https://en.wikipedia.org/wiki/UV_mapping>. Luettu 25.11.2017
- 11 A checkered sphere, without (left) and with (right) UV mapping (3D checkered or 2D checkered). UV-mapping. 2018. Verkkoaineisto. <https://en.wikipedia.org/wiki/UV_mapping>. Luettu 25.11.2017.
- 12 i85. 2017. Verkkoainesto. <<http://glamox.com/fi/products/i85-led>>. Luettu 25.11.2017.

- 13 C10-S1. 2017. Verkkoaineisto <<http://glamox.com/fi/products/qun35ry>>. Luettu 25.11.2017.
- 14 C75-P. 2017. Verkkoaineisto. <<http://glamox.com/fi/products/c75-p>>. Luettu 25.11.2017.