

KARELIA-AMMATTIKORKEAKOULU  
Tieto- ja viestintäteknikan koulutus

Emma Timonen

VALOTAITEEN SUUNNITTELUTYÖKALU  
VIRTUAALITODELLISUUDESSA PELIMOOTTORILLA  
TOTEUTETTUNA

Opinnäytetyö  
Syyskuu 2018



**OPINNÄYTETYÖ**  
**Syyskuu 2018**  
**Tieto- ja viestintäteknikan koulutus**

Tikkarinne 9  
80200 JOENSUU  
+358 13 260 600

Tekijä  
Emma Timonen

Nimeke  
Valotaiteen suunnittelutyökalu virtuaalitodellisuudessa pelimoottorilla toteutettuna

Toimeksiantaja  
Joensuu Games OSK

**Tiivistelmä**

Tässä toiminnallisessa opinnäytetyössä esitellään virtuaalitodellisuuteen kohdistuvaan sovellukseen sisällytetty työkalu, joka luotiin taiteellista valaistuksensuunnittelua varten. Opinnäytetyön tietoperusta käy lävitse työkalun tekoon käytettyjä välineitä, kuten Unreal Engine 4 -pelimoottoria ja visuaalisen grafiikkaan liittyviä ominaisuuksia. Lisäksi opinnäytetyössä kerrotaan nykyaikaisesta virtuaalitodellisuudesta kuin sen historiastakin ja käsitellään valaistusta taiteen muotona esimerkein.

Työkalun toteutuksen tärkeimmät toiminnallisuudet esitellään. Niiden valmistaminen pelimoottorissa ja ohjelmakoodissa käsitellään erikseen ja lopputuloksia virtuaalitodellisuus-sovelluksessa näytetään lisäksi kuvakaappauksin.

Opinnäytetyössä pohditaan lopputulosta ja siitä heränneitä ajatuksia tuotoksen tekemisestä, sekä teon aikana läpikäytyistä haasteista. Valmista työtä verrataan alussa asetettuihin tavoitteisiin. Pohdintaosiossa katsastetaan työn myötä tapahtunutta ammatillista oppimista ja käydään lävitse kehitysideoita.

Kieli  
suomi

Sivuja 41

**Asiasanat**

virtuaalitodellisuus, pelimoottori, valaistuksensuunnittelu, valotaide, unreal engine



**THESIS**  
**August 2018**  
**Degree Programme in Information and  
Communications Engineering**  
Tikkarinne 9  
80200 JOENSUU  
FINLAND  
+ 358 13 260 600

Author  
Emma Timonen

Title  
Planning Tool for Light Art in Virtual Reality, Implemented via a Game Engine

Commissioned by  
Joensuu Games OSK

#### Abstract

This functional thesis presents a tool, which is included in an application targeted for virtual reality and was created for the use of planning of lighting as a form of art. The information content of this thesis goes through instruments used for creation of the tool, such as the game engine called Unreal Engine 4 and attributes related to visual graphics. Modern time virtual reality along with its history is reviewed as well and light art is covered with examples.

The most important functionalities of the tool are exhibited. Their production in game engine and script is presented individually and the results in the virtual reality application are demonstrated with screenshots as well.

In the thesis the results and thoughts brought by them as well as the challenges gone through during producing tasks are deliberated. The final product is contrasted with the objectives set in the beginning. In the conclusions section possible professional learning assimilated during production is inspected and ideas of developing the tool further are reviewed.

Language

Finnish

Pages 41

Keywords

virtual reality, game engine, planning of lighting, light art, unreal engine

## Sisältö

Lyhenteet .....	5
1 Johdanto .....	6
2 Valaistus taiteen muotona.....	7
3 Virtuaalitodellisuus .....	8
3.1 Historia .....	9
3.2 Nykytila .....	10
3.3 HTC Vive .....	12
4 Pelimoottori.....	14
4.1 Unreal Engine 4 .....	15
4.2 Blueprint Visual Scripting.....	16
4.3 Renderöinti ja shaderit.....	17
4.4 Valaistus pelimoottorissa .....	18
5 Lähtötilanne .....	19
6 Toteutus.....	21
6.1 Luominen.....	22
6.2 Poistaminen.....	23
6.3 Kääntäminen.....	24
6.4 Siirtäminen.....	26
6.5 Kopioiminen.....	27
6.6 Mukauttaminen .....	29
6.7 Undo ja Redo.....	32
7 Lopputulos .....	34
8 Pohdinta.....	35
8.1 Vertailu tavoitteisiin.....	36
8.2 Opinnäytetyön haasteet ja ammatillinen kehitys .....	37
8.3 Kehitysideat .....	39
Lähteet.....	40

## Lyhenteet

AI	Artificial Intelligence, tekoäly
API	Application Programming Interface, ohjelmointirajapinta
Bugi	Bug, ohjelmointivirhe
BVS	Blueprints Visual Scripting; Epic Gamesin kehittämä Unreal Enginen visuaalinen ohjelmointijärjestelmä
DX11	DirectX 11, grafiikkateknologiaan tarkoitettu ohjelmointirajapinta
FOV	Field of View, näkökenttä
GB	Giga Byte, gigatavu, muistin kokoyksikkö, 1 gigatavu = 1000 000 000 tavua
HDMI	High Definition Multimedia Interface, liitännästandardi kuvalle ja äänelle
HMD	Head-mounted display, päähän puettavat virtuaalitodellisuuslasit
RAM	Random Access Memory, luku-/kirjoitusmuisti, varastoi tietokoneen käytössä olevaa dataa
Redo	Erinäisissä tietokonesovelluksissa käytetty toiminto, tee uudestaan
UE4	Unreal Engine 4, pelimoottori
Undo	Erinäisissä tietokonesovelluksissa käytetty toiminto, kumoa
USB	Universal Serial Bus, liitännästandardi oheislaitteille
VR	Virtual Reality, virtuaalitodellisuus

# 1 Johdanto

Opinnäytetyönäni esittelen pelimoottorin kautta toteutetun sovelluksen, jonka avulla on mahdollista suunnitella maisemointiin ja valotaiteeseen tarkoitettua valaistusta virtuaalisessa todellisuudessa. Työn ohessa käsitellään seuraavia pääaiheita: valaistuksen suunnittelua ja sen toteutusta pelimoottorissa, virtuaalitodellisuutta, sekä pelimoottoria.

Toteutin ammattikorkeakouluopintojeni vaatiman viiden kuukauden harjoittelujakson yrityksessä Joensuu Games OSK keväällä 2017. Kyseessä on joensuulainen yritys, joka toteuttaa 3D- ja VR-visualisointeja. Aloitettuani yrityksessä työharjoittelun minut keskitettiin projektiin, jossa kehitettiin virtuaalilaitteilla toimivaa sovellusta. Sovelluksessa oli ideana toteuttaa tapa, jolla valotaiteilija Kari Kola pystyisi suunnittelemaan Luminous Finland 100 -projektia varten valaistuksen Saana-tunturille helposti, ilman konkreettista läsnäoloa Pohjois-Suomessa. Virtuaalilaitteet mahdollistaisivat todenmukaisen näkymän valaistusta tunturista. Sain Joensuu Games OSK:lta toimeksiannon ohjelmoida projektiin työkalun valojen luomiseen ja muokkaamiseen virtuaalitodellisuudessa.

Maisemavalauksen suunnittelussa yksi haasteita tuovista tekijöistä on se, ettei välttämättä aina pystytä tarpeeksi havainnollistamaan tai kuvittelemaan selkeästi, miltä valaistus lopulta tulisi näyttämään. Suunnittelussa on monia asioita, joita täytyy ottaa huomioon ja riskinä on aina se, ettei lopputulos näyttäisikään lopputuloksessa niin hyvältä kuin konseptina. Valaistuksensuunnittelutyökalun tarkoitus on näin ollen toimia juuri nimensä mukaisesti työkaluna, jonka avulla voi suunnitella valaistusta. Työkalu mahdollistaa virtuaalisessa todellisuudessa valaistuksen suunnittelussa välittömän visualisoinnin, joka vastaa mahdollisimman paljon näkyä oikeassa maailmassa. Toisin sanoen virtuaalisesti luodussa, oikeaa sijaintia mukailtavassa ympäristössä, on mahdollista suunnitella valaistusta ja nähdä suoraan, miltä tietynlainen valaistus kyseisessä kohteessa tulisi näyttämään. Sen avulla voi testata jo suunniteltua valaistusta sitä hiotakseen, tai aloittaa valaistuksen suunnittelu tyhjästä kokeilemalla erilaisilla

valoilla, mikä mahdollisesti näyttäisi hyvältä. Todenmukaisten tulosten saamisessa onkin tärkeää kehittää sovelluksen valaisimet, valaistus ja visualisointi niin tarkoiksi ja luonnonmukaisiksi kuin mahdollista.

Opinnäytetyön tarkoitus on näin ollen esitellä virtuaalitodellisuuden tuomia mahdollisuuksia ympäristön valaistuksen suunnittelussa, sekä yksi toteutustapa. Sovellusta voi käyttää viihdyttävänä ajanvietteenä, mutta pääasiassa se on luotu hyötytarkoitusta ajatellen.

## **2 Valaistus taiteen muotona**

Valotaide on taiteen muoto, jossa nimensä mukaisesti keskiössä on valo. Se voi olla luonnonvaraista valoa, esimerkiksi auringonvaloa, tai vaihtoehtoisesti keinotekoisista valoa lähtöisin erinäisistä valaisimista. Valotaideteoksia voi nähdä rajatussa tilassa, kuten galleriassa tavanomaisen taiteen tapaan, tai rajaamattomassa tilassa, josta tyypilliset esimerkit ovat luonto ja kaupunkitilat. Valotaideteoksia voidaan jakaa ryhmiin niiden ajallisuuden ja liikkeen mukaan. Ryhmät ovat paikallaan pysyvät, muuttuvat ja liikkuvat valoteokset. (Salovaara 2007, 3.)

Kaupunkiympäristössä valotaiteella koristellaan usein kaupunkia tai korostetaan tärkeää arkkitehtuuria. Tyypillinen käytötapa tällaisessa tapauksessa on rakennuksen yksityiskohtien kohdistaminen valoilla, mahdollisesti värien kera, jolloin ohikulkija noteeraa sen herkemmin.

Suomen 100-vuotisjuhlavuoden kunniaksi järjestettiin Luminous Finland 100 -valotaidekokonaisuus vuoden 2017 lopulla. Joensuulainen valotaiteilija Kari Kola suunnitteli kuuteen suomalaisille tuttuun ja tärkeään kohteeseen valaistuksen. Nämä kohteet olivat Helsingin Töölönlahti, Olavinlinna, Näsinneula, Oulun Kuusisaari, Saana-tunturi sekä Turun linna. Taideteoksilla korostettiin kohteiden myötä itsenäistä Suomea ja taideteosten valojen värinä käytettiin tästä syystä

sinistä. Tempaus on ollut tämän työn kirjoittamiseen mennessä maailman suurin valotaidekokonaisuus. (Valoparta Oy 2017.)

### **3 Virtuaalitodellisuus**

Kolmiulotteinen esittäminen, virtuaalinen tila, on ihmiselle luonnollinen tapa havainnoida ympäristöä, sillä meitä ympäröivä todellisuuskin on kolmiulotteinen. Virtuaalisen todellisuuden etu ja suurin ero meille tuttuun todellisuuteen verraten ovat rajoittamaton fysiikka ja tilaisuus kokea hypertodellisia asioita, jotka eivät ole maailmassamme mahdollisia. (Hintikka, Kojo & Metsämäki 1998, 5.)

Ihminen kokee maailmaa aisteillaan. Korvaamalla nähdyt kuvat, äänet ja käsinkosketeltavat asiat tietokoneella toteutetuilla illuusioilla voidaan luoda interaktiivinen, virtuaalinen maailma ihmiselle koettavaksi (Bates-Brkljac 2012, Alkusanat vii). Ihmismieltä kiehtoo kokea ja tutkia tuntemattomia maailmoja mahdollisimman todentuntuisesti ja virtuaalitodellisuutta hyödynnetäänkin suuresti viihdeteollisuudessa, esimerkiksi videopeleissä. Käyttäjä saa virtuaalitodellisuutta hyödyntäen todentuntuisemman kokemuksen pelimaailmasta ja siinä tapahtuvista asioista. VR on kykenevä sulkemaan käyttäjältä ympäröivän maailman, jolloin tämä asettuu välittömämmin virtuaaliseen maailmaan ja todellisuuteen, tuntien enemmän olevansa osa sitä. Kokemukset virtuaalitodellisuudessa voivat näin ollen olla syvemmät ja tunteita herättävämmät, mieleenpainuvammat.

Viihdetarkoitukseen virtuaalitodellisuus on siis omiaan, mutta sitä sovelletaan myös erityisen paljon hyötykäytössä. VR:ää hyödynnetään monenlaisissa simulaattoreissa, joissa käyttäjä voi mm. tutustua johonkin, mitä hänen ei välttämättä ole mahdollista tositilanteessa päästä kokemaan. Myös opetus- ja koulutustarkoitukseen virtuaalitodellisuutta käytetään; esimerkiksi lentäjäkoulutuksissa käyttäjä voi päästä lentosimulaattorilla tutustumaan lentotilanteeseen itse ennen varsinaista ilmaan nousemista ja konkreettista lentämistä. Näin todellisia tilanteita varten voi harjoitella ennalta, mikä kasvattaa

itseluottamusta ja madaltaa kynnyistä ottaa ohjat käsiinsä oikeassa lentokoneessa. Virtuaalitodellisuus voi siis toimia erinomaisena esiasteena uuden asian harjoittelussa.

Monia vapaa-ajan harrastuksiakin on mahdollista kokeilla VR-simulaattoreilla. Yksi esimerkki tästä on ampumaharrastus, joka on kallis, eikä välttämättä helposti saatavilla oleva harrastus. Virtuaalitodellisuutta hyödyntäen on tehty tätä varten ampumaratasimulaattori, jonka avulla ampumaharrastukseen voi tutustua ilman varsinaisella ampumaradalla käyntiä. (Tapanainen 2017, 30.)

Itse pääsin kokeilemaan virtuaalitodellisuutta ensimmäisen kerran työharjoitteluni alkaessa. Kokeilun kohteena oli HTC Vive -laitteen kehittäjän, Valven, tekemä The Lab -niminen sovellus, jossa on useita erilaisia virtuaalitodellisuuden testikohteita. Se sisältää pelejä, kuten jousiammunta, matkailua, jonka kautta pääsee katselemaan mm. Islannin kallioperäistä erämaata kuin olisi paikan päällä, ja tieteellisiä kohteita, kuten aurinkokunta, jossa pääsee tutustumaan Linnunrataan. The Lab on erittäin monipuolinen ensikosketus virtuaalitodellisuuteen ja jo sen omat pienet testikohteet kertovat, että erilaisia käyttötarkoituksia on lukuisia.

### **3.1 Historia**

Virtuaalitodellisuus ei ole niin tuore käsite kuin ensi alkuun ajattelisi. Sen historiallinen alkutaipale juontaa juurensa jo 1800-luvulle, jolloin ihmiset pääsivät kokemaan esimerkiksi laivamatkan Nizzasta Konstantinopoliin mekaanisen simulaattorin avulla. 1969—1970 ovat kuitenkin ne nykymallisen virtuaalitodellisuuden syntyvuodet – samoja aikoja kuin internetinkin syntyajat. Ensimmäiset nykylaitteistoja myötäilevät virtuaalitodellisuusjärjestelmät ovat Ivan Sutherlandin ja Myron K. Kruegerin käsialaa. Sutherland rakensi Damokleen miekka -nimisen järjestelmän, jossa oli katosta roikkuva datakypärä, jota hän kutsui ”ikkunaksi virtuaalitodellisuuteen” (The Franklin Institute 2018). Krueger taas kehitti Glowflow’n, seinälle projisoitavan virtuaalitalan, jossa oli tunnistimet seuraamassa käyttäjän liikkeitä. Tällainen kaksijakoisuus on säilynyt koko

virtuaalitodellisuuden historian ajan. Datakypäriä käytetään yhä sulkemaan ulkomaailma käyttäjän aisteilta ja tunnistimilla seurataan käyttäjän liikkeitä ja sijaintia, jolloin ei tarvita, Kruegerin sanoin, "vaatekaappia" käyttäjän ylle. (Hintikka, Kojo & Metsämäki 1998, 7.)

Useat eri tekijät aloittivat virtuaalikokeet vuonna 1984 ja erityisesti mainittakoon vuoden 1985 Yhdysvaltain avaruusasemahallinnon, NASA:n, kehitysprojekti edullisempien virtuaalilaitteiden kehittämiseksi. Sen VIEW (Virtual Interface Environment Workstation) -järjestelmä yhdisti virtuaalitodellisuuslasit ja tuntoaistiin perustuvan interaktion mahdollistavat käsineet (The Franklin Institute 2018). Sotateollisuuden käyttämät mallit eivät soveltuneet siviilikäyttöön, jolloin Michael McGreevy aloitti VIVED (Virtual Environment Display) -projektin kehittääkseen immersiiivisen tietokonegraafisen simulointijärjestelmän (immersive computer graphics simulation system) ja ensimmäiset kaupalliset sovellukset tulivat markkinoille vuonna 1988. Tällöin nykymuotoiset virtuaalikypäri ja datakäsine tulivat yleiseen virtuaalitodellisuuskäyttöön. (Hintikka, Kojo & Metsämäki 1998, 7.)

Kolmiulotteinen virtuaalitila koki suuren hyppäyksen vuoden 1991 id Softwaren Doom-pelin myötä, joka oli monien kotikäyttäjien ensimmäinen kokemus kolmiulotteisessa virtuaalitilassa. Sittemmin kolmiulotteisuutta ja sitä myöten virtuaalitodellisuutta on kehitetty suuresti eteenpäin, mutta järkkyyviä muutoksia ei kolmeen vuosikymmeneen ole enää nähty, vaikka laitteet ja todentuntuisuus ovat toki menneet paljon eteenpäin. (Hintikka, Kojo & Metsämäki 1998, 8.)

### **3.2 Nykytila**

Virtuaalitodellisuutta käytetään ja hyödynnetään laajalti eri osa-alueilla, kuten puolustus- ja ilma-alalla, lääketieteellisessä koulutuksessa ja harjoittelussa, sekä taiteessa (Freeman 2008). Kuten jo todettua, se ei ole ainoastaan viihteellistä käyttöä varten, vaan virtuaalitodellisuus mahdollistaa suuria hyötykäyttötarkoituksia realismia tavoitellessaan.

Mielestäni mielenkiintoinen kertomus VR:än hyötykäytöstä on Daniel Freemanin artikkelissa ”Studying and Treating Schizophrenia Using Virtual Reality: A New Paradigm”. Artikkelissa kerrotaan, kuinka virtuaalitodellisuutta on hyödynnetty mielenterveyspotilaiden tutkimiseen ja erinäisistä fobioista, esimerkiksi akrofobiasta, korkeanpaikankammosta, kärsivien henkilöiden auttamiseen. Siinä kuvaillaan, kuinka virtuaalitodellisuudessa käytetyt korkeat tilat aiheuttivat korkeanpaikankammoa poteville kauhureaktioita ja fyysistä oireilua – yhtä lailla kuin jos he olisivat reaali maailmassa olleet korkealla. Toinen esimerkki artikkelissa on eräästä ihmisryhmästä, jota pyydettiin esittämään puheita kolmen virtuaalisen, mutta eri tavalla reagoivan yleisön edessä. Yleisölle puhumisen pelko on hyvin yleinen. Yleisö oli ohjelmoitu reagoimaan puhujaan joko positiivisesti, neutraalisti tai negatiivisesti. Sen tarjoamat reaktiot vaikuttivat puhujaan kuin tilanne olisi ollut oikea. Positiivisesti reagoiva yleisö sai puhujan tuntemaan olonsa hyväksi ja kuunnelluksi, kun taas negatiiviset reaktiot saivat aikaan ajatuksen karkailua ja huomattavaa epävarmuutta. Tämän lisäksi esiintymisen pelkoa potevat oireilivat fyysisesti, mistä kertoivat silmänräpäytysten lisääntyminen ja sydämensykkeen tiheneminen. (Freeman 2008.)

Virtuaalitodellisuudella voidaan näin ollen luoda aitoja pelko- ja realistisuutta vastaavia tiloja, joiden avulla ihmismieltä voidaan parhaimmillaan muovata ja auttaa heitä kohtaamaan pelkonsa uudella tavalla. Artikkelissa todetaan, että potilaiden interaktiot sosiaalisessa maailmassa ovat avain psykoosin ymmärtämiseen ja hoitamiseen. Virtuaalitodellisuuden avulla esiintymispelosta kärsiviä on hoidettu onnistuneesti. (Freeman 2008.)

Virtuaalilaitteita ei yleensä suositella lasten käyttöön. HTC Viven turvallisuusohjeissa sanotaan, ettei tuotetta ole suunniteltu lasten käytettäväksi (Holly 2016). Playstationin oman VR-laitteiston ohjeissa taas kerrotaan, että ”virtuaalitodellisuuslaseja ei ole tarkoitettu alle 12-vuotiaiden lasten käytettäväksi” (Sony Interactive Entertainment 2016). Lasten silmien ja näön kehittyminen voi vahingoittua suuresta käytöstä ja lisäksi lapsilla on todettu tasapainon pitämisessä ongelmia kymmenien minuuttien virtuaalitodellisuudessa olon jälkeen (Eva 2017). Laitteita kehitetään eteenpäin kaiken aikaa ja lienee sopivaa

olettaa, että tulevaisuudessa lasit saadaan vielä ohittamaan lasten kehitysvaiheet, jolloin myös nuoremmat voisivat käyttää laitteita turvallisesti.

### 3.3 HTC Vive

Tässä työssä käsitellyn sovelluksen tekemiseen on hyödynnetty HTC Vive -virtuaalilaitteistoa. HTC on taiwanilainen, vuonna 1997 perustettu yhtiö, joka kehitti Vive-virtuaalilasit yhdessä pelejä ja pelilaitteita kehittävän yrityksen, Valve Corporationin, kanssa. Viven laitteisto sisältää VR-lasit, kaksi ohjainta sekä kaksi majakkaa. Kuvassa 1 itse käyttämäni Vive-laitteisto.

HTC Vive on markkinoiden ensimmäinen kotikäyttöön tarkoitettu huoneensuuruinen VR-laite. Huoneensuuruisella tarkoitetaan sitä, että toisin kuin muut vastaavat laitteistot, Vive mahdollistaa maksimissaan 5 m x 5 m -kokoisen tilan käytön virtuaalisten maailmojen tutkimiseen. Tilan nurkkiin asetettavissa majakoissa on infrapunavalolähtimet. Lasit ja ohjaimet vastaanottavat valon ja käyttävät siitä saatua dataa hyväkseen tunnistaakseen, missä päin huonetta käyttäjä sillä hetkellä sijaitsee. Tämän myötä käyttäjä voi liikkua tilassa ja samaan aikaan virtuaalisessa ympäristössä vapaasti, mikä korostaa todellisuuden tunnetta. (Holly 2016.)

Laitteessa on Chaperone-niminen järjestelmä, joka estää käyttäjää törmäämästä mihinkään tai kolauttamasta ohjaimia esineisiin tai huonekaluihin ollessaan virtuaalitalassa. Aloittaessaan laitteen käytön käyttäjä määrittää itse pelialueensa tilan. Käyttäjän liikkuessa pelitalassa lähelle omaa määrittämäänsä rajaa virtuaalikuvaan tulee näkyviin kehikko, joka kertoo käyttäjälle hänen piirtämänsä seinän tulevan vastaan ja näin ollen ohjaa käyttäjää liikkumasta sitä pidemmälle kyseiseen suuntaan. Chaperone luo siis turvallisuutta laitteiston käytettävyyteen, eikä käyttäjän tarvitse huolehtia törmäämisestä tai esineiden rikkomisesta lasit päässä ollessaan ja liikkuessaan. (Holly 2016.)



Kuva 1. HTC Vive -laitteet; virtuaalilasit, infrapunamajakat sekä ohjaimet.

Yksi tärkeimmistä ominaisuuksista virtuaalilaseissa on niiden FOV, näkökenttä. Ihmisen kahden silmän näkökenttä on 200—220 astetta. Mitä laajempi näkökenttä laseissa on, sitä enemmän käyttäjä tuntee olevansa virtuaalimaailmassa läsnä. Viven lasien maksimaalinen FOV on 110 astetta. (Jay 2016.)

HTC Vive ei ole yksinään toimintakelpoinen, vaan se vaatii tehokkaan tietokoneen, jolla suorittaa VR-sovelluksia. Alla luettelona Viven minimivaatimukset tietokoneelle:

- Näytönohjain: NVIDIA GeForce GTX 1060 tai AMD Radeon RX 480
- Prosessori: Intel Core i5-4590 tai AMD FX 8350
- Muisti: 4 GB RAM

- Video-ulostuloportti: HDMI 1.4, DisplayPort 1.2
- USB-portit: 1x USB 2.0
- Käyttöjärjestelmä: Windows 7 SP1, Windows 8.1, Windows 10

Viven sivuilta voi myös ladata sovelluksen, joka testaa tietokoneen soveltuvuuden laitteelle. (HTC Corporation 2018.)

## 4 Pelimoottori

Pelimoottori on alusta, joka tarjoaa valmiiksi rakennettua sisältöä pelin kehittämiseen, jotta pelintekijät voivat keskittyä paremmin omien peliprojektinsa yksityiskohtiin ja omanlaistensa pelien luontiin. Moottori voi sisältää komponentteja mm. animaatioiden luomiseen, fysiikoihin ja törmäyksen tunnistukseen. Esimerkiksi fysiikkakomponentin avulla kehittäjän ei tarvitse välttämättä ohjelmoinnin puolella tuhlata ajatustakaan siihen, kuinka objektin tulisi käyttäytyä vaikkapa pudotessaan maahan. Pelimoottoriin rakennettujen fysiikoiden avulla hän voi kertoa nämä asiat pelilleen vain arvoja (esim. kitkan voima) säätämällä, jolloin moottori huolehtii lopusta. Se on arkkitehtuuri, joka suorittaa pelin toimintoja.

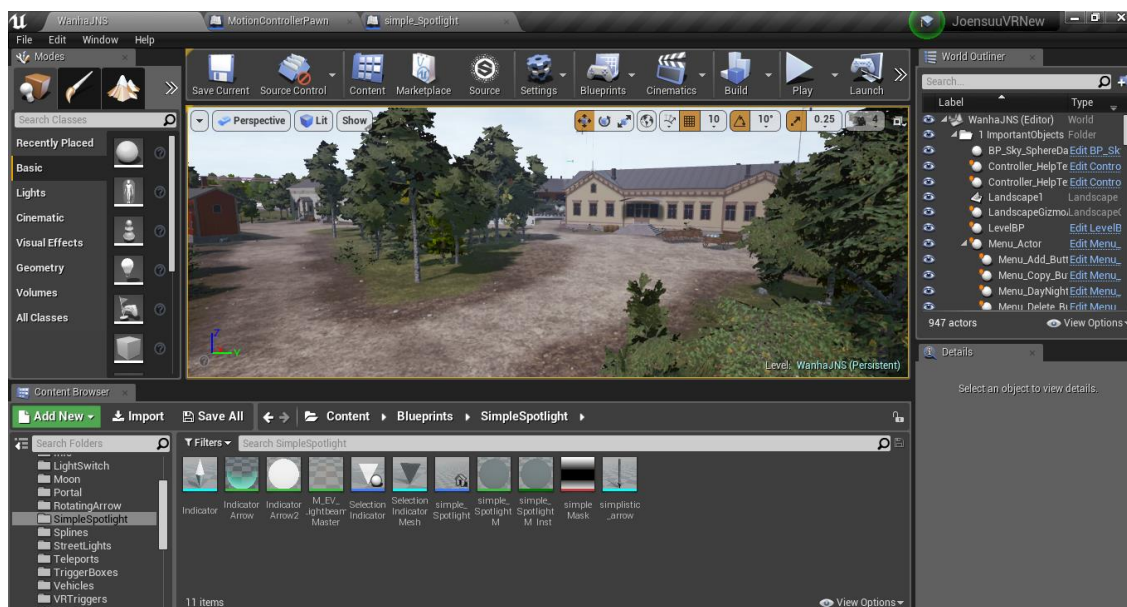
Pelimoottorille ei löydy mitään yhtä, varsinaista määritelmää, mutta sen tehtävänä on helpottaa pelintekijöiden työtä alustamalla yleisiä, peliin oletettavasti kuuluvia osa-alueita, joiden avulla kehittäjät voivat viilata ominaisuuksia peleihinsä ilman, että jokaista asiaa tarvitsisi ohjelmoida erikseen itse. Se vähentää pelin kehittämisen työn määrää avittaen kehittäjiä automatisoimalla perusideat. Alustaan kuuluu myös visuaalinen käyttöliittymä.

Jotkin yritykset ovat luoneet omat pelimoottorinsa saadakseen pelin toteuttamaan täydellisesti heidän näkemyksiään ja vähentääkseen rajoituksia. Moottoreita on myös yleisesti jaossa ja jopa ilmaiskäytössä, mikä mahdollistaa niihin tutustumisen ja niillä työskentelyn pienemmillekin firmoille ja yksityishenkilöille. Suosituimpia tällaisia pelimoottoreita ovat Unity, Unreal Engine ja GameMaker. Tässä työssä käytössä oli Unreal Engine 4.

## 4.1 Unreal Engine 4

Epic Gamesin luoma Unreal Engine on tämän päivän johtavimpia ja eniten käytettyjä pelimoottoreita maailmassa. Se julkaistiin alun perin vuonna 1998 ja sitä kehitetään yhä edelleenkin. Unreal Enginellä on luotu suuriakin pelinimikkeitä, kuten Mass Effect ja Batman: Arkham -sarjojen pelit. (Game Designing 2018.)

Vaikka pelimoottori on suunniteltu ennen kaikkea pelien kehittämistä varten, sitä käytetään myös toisenlaisten visuaalisten projektien, esimerkiksi animaatioiden, tai - kuten tämän opinnäytetyön tapauksessa - hyötykäyttöön tarkoitetun interaktiivisen sovelluksen luontiin. Ottamassani kuvakaappauksessa (kuva 2) esitetään Unreal Enginen perusnäkyä. Unreal Engine on suunniteltu vaativille sovelluksille, kuten AAA-peleille, elokuvantekoon sekä fotorealistiseen visualisointiin. Virtuaalitodellisuus vaatii korkeat kuvataajuudet ja tarkkaa renderöintiä luodakseen ihmismielelle uskottavan kokemuksen. UE vastaa näihin vaatimuksiin ja tarjoaa vankan perustuksen rakentaa sisältöä kaikille VR-alustoille. (Epic Games 2018.)

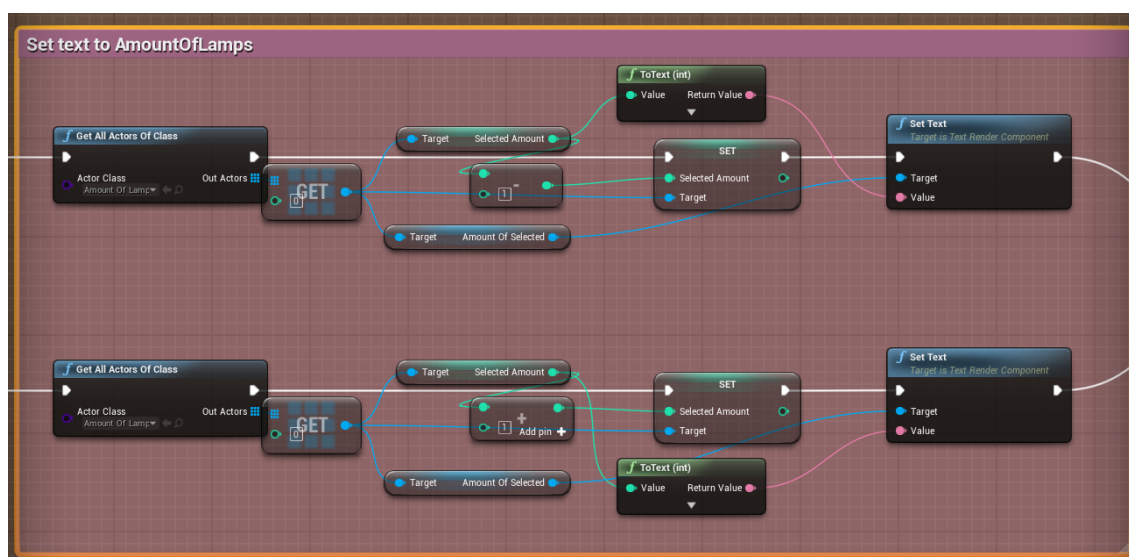


Kuva 2. Näkymä Unreal Engine 4:stä.

## 4.2 Blueprint Visual Scripting

Unreal Engine 4 tarjoaa kehittäjille visuaalisen ohjelmointijärjestelmän, nimeltään Blueprint Visual Scripting (BVS). BVS-järjestelmä Unreal Enginessä on kokonainen ohjelmointijärjestelmä, joka perustuu konseptiin solmupohjaisesta käyttöliittymästä. Kuten monissa yleisissä ohjelmointikielissä, sitä käytetään moottorissa määrittelemään olioperustaisia luokkia tai objekteja. (Epic Games 2018.)

Blueprinttien käyttö on esimerkiksi ei-ohjelmointitaustasta tulleille erinomainen vaihtoehto ohjelmoimista ajatellen ja erityisesti graafikot ovat yleisesti pitäneet mahdollisuudesta visuaalisempaan ohjelmointimuotoon. Se on monille kenties helpommin lähestyttävä tapa ohjelmoida perinteisen koodin sijasta, mikä kaventaa kuilua graafikon ja ohjelmoijan välillä sekä pienentää graafikon kynnystä ohjelmoida tai tutustua koodiin. Blueprintit käyttävät paljon värikoodeja. Esimerkiksi erityyppiset muuttujat ovat aina tietyn värisiä; kokonaislukuarvot turkooseja, tekstiarvot vaaleanpunaisia ja objektit sinisiä (kuva 3). Tämä on yksi avainominaisuuksista visuaalisemmassa ohjelmoinnissa, jossa käyttäjä osaa heti värin perusteella yhdistää ja tunnistaa koodin pätkiä. Käyttäjää muistaa, miltä tietty koodi näyttää.



Kuva 3. Blueprint Visual Scripting -ohjelmakoodia. Skriptin ympärillä on myös värikoodattu kommenttikenttä.

Blueprint Visual Scripting ei kuitenkaan kykene pelimoottorissa täysin samaan kuin C++, joka on Unreal Engine 4:n pääasiallinen ohjelmointikieli. Siksi esimerkiksi suurempien peliprojektien tekemiseen sitä ei suositella, sillä blueprintit ovat ohjelmalle raskaampia ja saattavat näin ollen hidastaa sovelluksen kulkua. Pienemmille projekteille se kuitenkin on varteenotettava vaihtoehto.

### **4.3 Renderöinti ja shaderit**

Renderöintiä varten pelimoottori vaatii hyvää grafiikkateknologiaa. Unreal Engine 4:ssä on renderöintiä varten Microsoftin DirectX 11 (DX11), joka on kokoelma sovellusohjelmointirajapintoja (Application Programming Interface, API). Kaikki 3D-ohjelmat tarvitsevat APIa ja sen yhdistäminen 3D-grafiikkakorttiin mahdollistaa reaaliaikaisen 3D-grafiikan luomisen (X3DMedia 2010). Kun DirectX 11 julkaistiin vuonna 2009, se johti vallankumouksellisiin tuloksiin tuomalla hahmojen ja objektien grafiikkaan merkityksellisen paljon syvyyttä ja yksityiskohtia, jollaisia ei pystytty aiemmilla, perinteisillä tekniikoilla saavuttamaan. Esimerkiksi hahmojen kasvoihin saatiin tämän avulla pyöreyttä ja realismia, ja ympäristöistä tuli orgaanisempia ja luonnonmukaisempia. (NVIDIA Corporation 2018.)

Peleissä ja pelimoottoreissa yksi tärkeimmistä ominaisuuksista hyvän ja realistisen grafiikan kannalta on ambient occlusion, joka tarkoittaa kahden pinnan tai objektin kohtaamisen myötä muodostamaa varjoa ja varjostamista. DX11:n myötä varjojen muodostamisesta on saatu realismia lisäksi nopeaa ja tehokasta. Varjostamisessa on esimerkiksi otettava huomioon varjoja luovan objektin etäisyys pintaan, johon varjo piirtyy. Luonnossa varjo on paljon terävämpi, kun sen luova asia on lähempänä varjon pintaa. Tätä on myös pyritty toteuttamaan Unreal Engine 4:ssä tuomaan lisää luonnollisuutta ja varjojen pehmeys on saatu DirectX 11:llä vaihtelevaan varjoa luovan objektin etäisyyden kasvaessa. (NVIDIA Corporation 2018.)

DX11 myös laskee shadereita, varjostimia, tehokkaasti ja ilman suuria odotusaikoja. Jälkikäsitellyssä varjostinten laskeminen sisältää esimerkiksi fysiikan, AI:n ja partikkelijärjestelmät, jotka vaativat runsaasti tehoja tietokoneelta. (X3DMedia 2010.)

#### **4.4 Valaistus pelimoottorissa**

Työkalussa käytetyt valot ovat Spot Light -valoja, eli kohdevaloja. Ne tuottavat valoa yhdestä pisteestä keilan muodossa ja ne sisältävät kaksi keilaa: sisä- ja ulkokeilan. Sisäkeilassa valo tuottaa täyden kirkkauden. Sen keilan reunan matka ulkokeilan reunaan tuottaa puolivarjoja ja keilojen välillä oleva alue pehmentää valoisuutta. Näiden keilojen säteen voi määrittää mielensä mukaan, kuten valokeilan pituudenkin. Ajatus kohdevalossa on siis sama kuin esimerkiksi taskulampussa tai näyttämövalossa. (Epic Games 2018.)

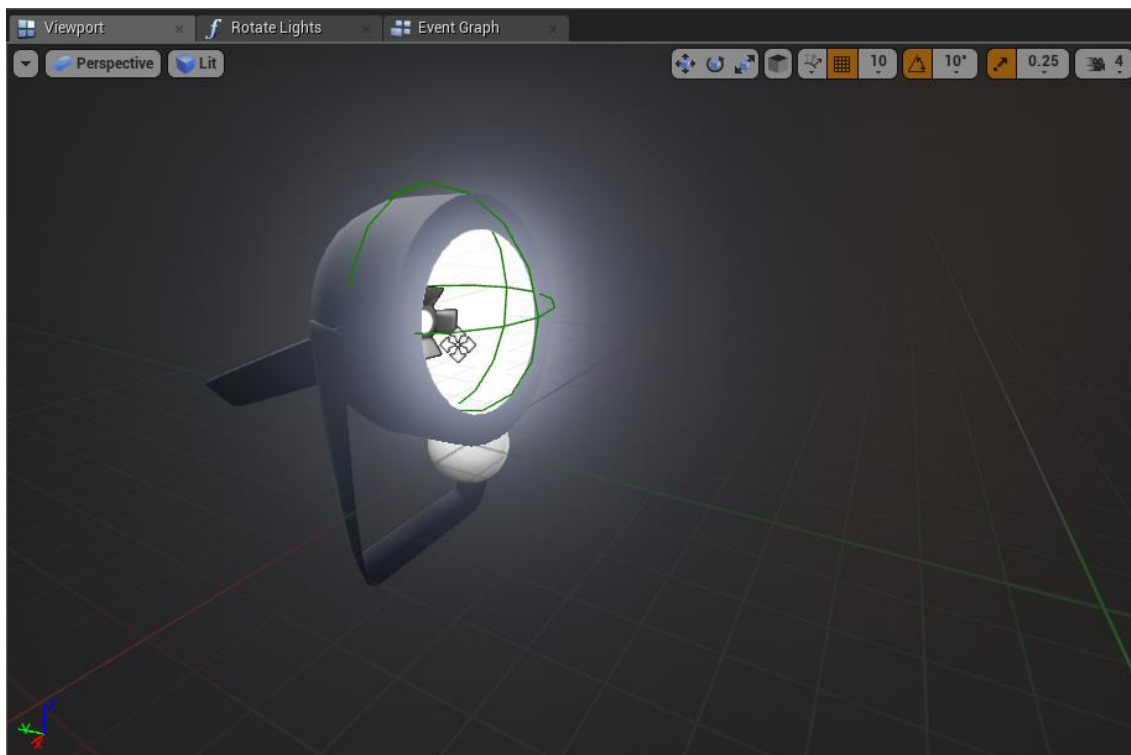
Unreal Engineissä valoja voi asettaa kolmella eri tavalla, jotka vaikuttavat pelimoottorin valon ja varjojen piirtoon. Nämä tavat ovat staattinen (static), kiinteä (stationary) ja siirrettävä (moveable). Staattisella tarkoitetaan UE:n tapauksessa sitä, ettei valo ja sen määrä vaihdu laisinkaan sovelluksen ollessa käynnissä, eikä valo liiku tai sen ominaisuudet muutu. Se on näin ollen moottorille myös nopein ja kevyin näistä kolmesta. Kiinteää valoa ei voi siirtää, eikä se siis piirrä uusia varjoja laisinkaan, mutta valon väriä ja kirkkautta voi muuttaa. Siirrettävä valo taas on täydellisesti muokattavissa, ollen siis dynaaminen valo. Dynaaminen valo on renderöinnin kannalta hitain. (Epic Games 2018.)

Valaistustyökalussa valoja tulee saada muokattua ja liikuteltua täysin käyttäjän mielen mukaan, joten nämä valot ovat siirrettäviä.

## 5 Lähtötilanne

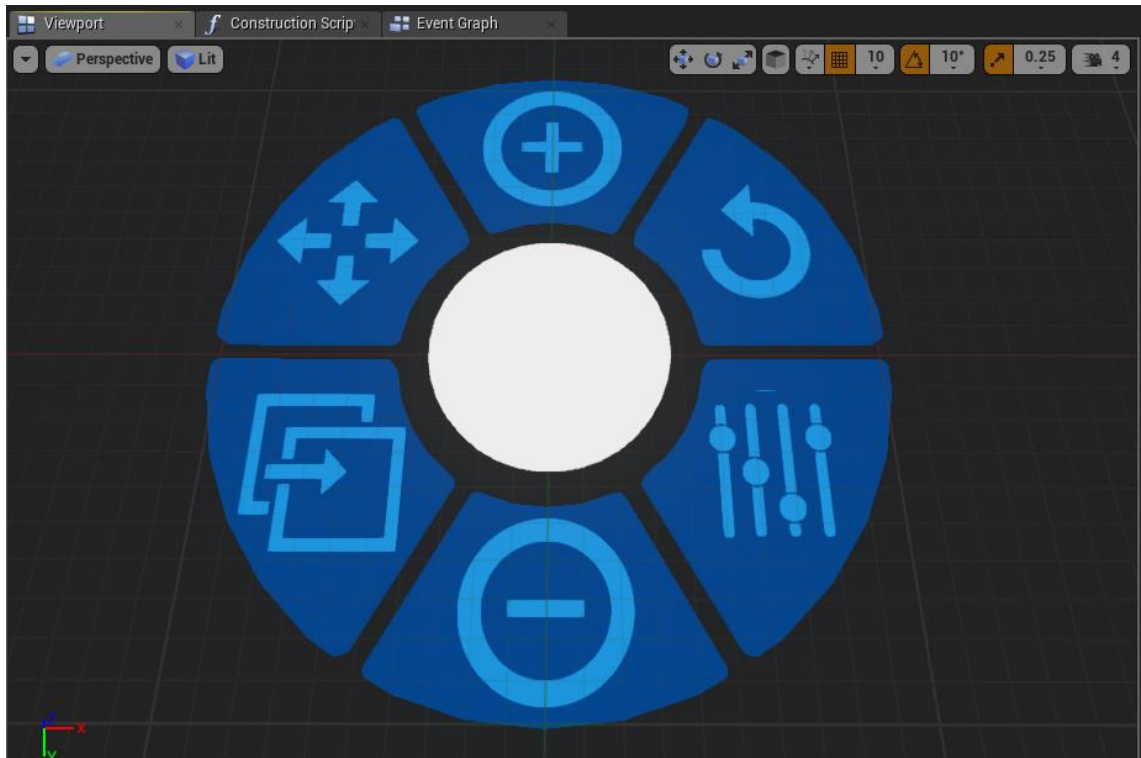
Työ alkoi luonnollisesti suunnittelulla. Projektiin tuli miettiä millaisia ominaisuuksia valoille tarvittaisiin ja mitä niillä voisi tehdä, mutta myös se, kuinka näihin toiminnallisuuksiin pääsisi käsiksi. Lisäksi VR-laitteisiin ja niiden käyttöön oli tutustuttava ennen kuin saattoi alkaa kuvitella mahdollisuuksia erilaisille toiminnallisuuksille ja miten niitä lähteä toteuttamaan.

Pelimoottoriin oli luotava uusi projekti ja mappi, johon alue rakennettiin ja jossa työkalua testattiin. Koska valaistustyökalu oli osa suurempaa projektia, graafikot ja mallintajat rakensivat projektissa Joensuun asemanseudun aluetta. Työkalu integroitiin projektiin ja se itsessään vaati myös osakseen kevyttä mallinnusta: graafikot loivat valo-objektille mallin aitoa valaisinta mukaillen (kuva 4) ja lisäksi muovasivat palat työkalun valikkoon, sekä loi niille yksilölliset, tunnistettavat 2D-ikonit.



Kuva 4. Mallintajien luoma valo-objekti, johon on liitetty spotlight-komponentti.

Valaistustyökaluun valikoituivat seuraavat toiminnallisuudet: luominen, poistaminen, siirtäminen, kääntäminen, kopioiminen ja mukauttaminen. Näiden lisäksi sovellukseen toteutettiin Undo-Redo -toiminto, eli kumoa- ja tee uudestaan -toiminnot. Valikossa on erikseen painikkeet jokaiselle toiminnallisuudelle (kuva 5). Ainoastaan Undo-Redo -toiminnot ohjelmoitiin toimimaan suoraan ohjaimen painikkeista niiden helpon saatavuuden vuoksi.



Kuva 5. Valikko valaisimien toimintoihin.

Lisäämistöiminnallisuuteen pääsee +-merkillä kuvatusta painikkeesta, sen oikealla puolella oleva kiertävä nuoli tarkoittaa valaisimen kääntämistä. Kääntöpainikkeen alla oleva painike kuvaa valaisimen ominaisuuksien mukauttamista, ja miinusmerkillä kuvataan valaisimien vähentämis- eli poistominaisuutta. Poistopainikkeen vasemmalla puolella oleva kuvake kertoo valaisimien kopiointi-toimintoa, jonka yllä puolestaan on siirtämistoimintoa edustava painike.

Painamalla jotakin näistä painikkeista, ei tapahdu mitään konkreettista, vaan sovellus on tällöin tietystä tilassa, esimerkiksi kopiointitilassa. Ohjaimessa oleva liipaisin toimii niin sanottuna toiminnallisuuspainikkeena. Tämä tarkoittaa siis sitä,

että kun valikosta on painettu vaikkapa juuri kopiointipainiketta, sovellus on kopiointitilassa ja se näin ollen tietää, että jos nyt käyttäjä painaa liipaisinta, sen on tarkoitus kopioida halutut valaisimet.

## 6 Toteutus

Virtuaalitodellisuusmaailman on mahdollista täyttää niin monilla valaisimilla kuin käyttäjä itse haluaa. Koska näin on, rakennettiin sovellukseen aiemmin mainittujen toiminnallisuuksien lisäksi mahdollisuus valita valaisimia jo maailmaan luoduista valo-objekteista. Näin niiden muokkaaminen on helpompaa, sillä jokaista ei välttämättä tarvitse ryhtyä mukauttamaan, tai esimerkiksi siirtämään yksilöllisesti. Valitsemistoiminnon avulla valaisimista voi ottaa ryhmän, jotta tehdyt muutokset välittyvät niille kaikille vain yhden sijaan.

Tätä valitsemiskäytäntöä voi hyödyntää jokaisessa toiminnossa. Valo-objektit muuttavat väriään mustasta valkoiseksi, ja tämä väri toimii käyttäjälle indikaattorina siitä, että valaisin on valittuna, ja seuraavat muutokset vaikuttavat tähän valittuun valaisimeen. VR-näkymässä on kaiken aikaa näkyvillä laskuri, jossa on nähtävissä valittujen valaisimien määrä suhteessa valaisimien määrään maailmassa kaiken kaikkiaan. Näin käyttäjä on alituisesti selvillä siitä, kuinka monta valaisinta hän on maailmaan luonut ja kuinka moni niistä on valittu.

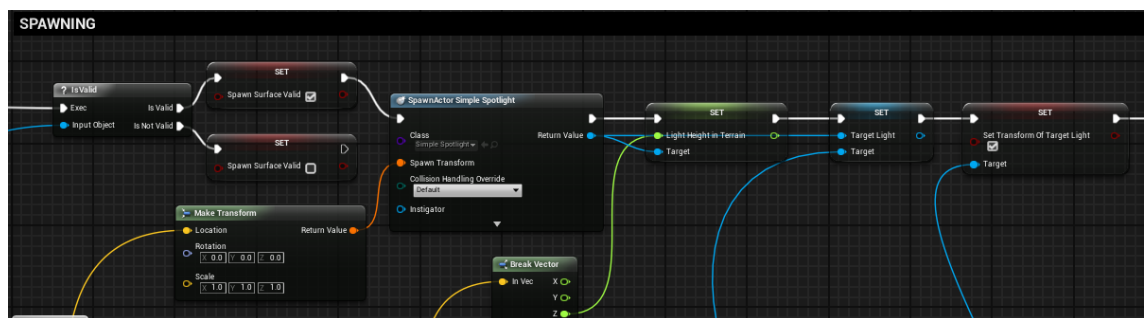
Yksittäisen valitsemisen lisäksi käytössä on myös Valitse kaikki- ja Poista valinnat -painikkeet. Valitse kaikki -toiminnolla sovellus asettaa jokaisen maailmassa olevan valaisimen valituksi, jolloin muutokset vaikuttavat jokaiseen olemassa olevaan valaisimeen. Poista valinnat -toiminnolla taas vastaavasti poistetaan kaikki valinnat, jolloin yksikään valaisin ei ole valittuna, eivätkä muutokset siis vaikuta mihinkään.

Nämä ovat sovellukseen lisättyjä perustoimintoja, jotka helpottavat ja nopeuttavat suunnittelutyötä. Seuraavaksi kerrotaan tarkemmin itse työkalun toiminnoista.

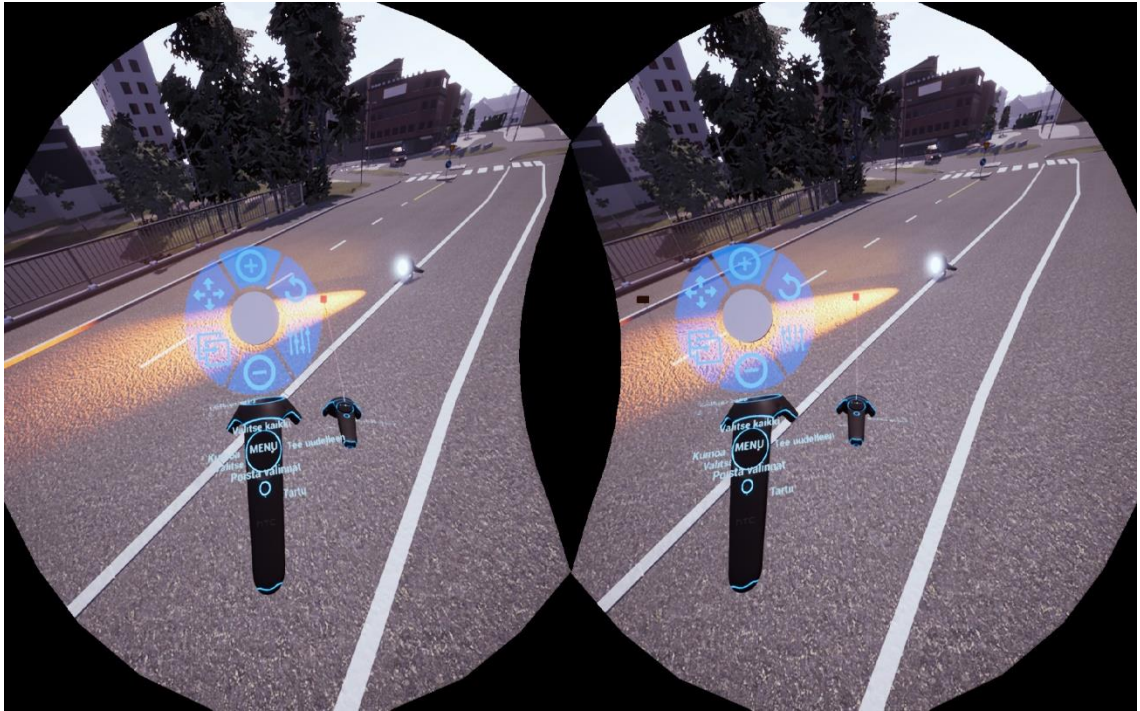
## 6.1 Luominen

Ensimmäinen toteutettu toiminnallisuus oli valaisimien lisääminen maailmaan. VR-ohjaimen ominaisuuksiin lisättiin laserosoitin, joka pystyi osoittamaan eri kohteisiin maailmassa ja tunnistamaan ne. Tämä ominaisuus rajattiin niin, että kohteet, joihin valaisimen pystyisi asettamaan, ryhmiteltiin erikseen estämään se, ettei valaisimia voinut sijoittaa esimerkiksi veteen tai puuhun. Näin ollen vain konkreettisen ja tarkoituksenomaisen pinnan päälle oli mahdollista sijoittaa valaisin.

Laserin osoittaessa esimerkiksi tienpintaan ja toimintanäppäintä painettaessa valaisin-objekti syntyi juuri siihen kohtaan, missä laser osuu yhteen kohteen pinnan kanssa. Toiminnallisuuden ohjelmoinnista tärkein osuus esiintyy alla, kuvassa 6. Ohjelmakoodi tarkistaa ensin, että laserin osoittama kohde on validi, eli kuuluu aiemmin mainitsemaan ryhmään sopivista alustakohteista. Jos näin on, valaisin-objekti luodaan ja ohjelmakoodi hakee laserin ja kohteen XYZ-osumakohdan valaisimen sijainniksi maailmassa sijoittaakseen sen siihen. Valaisin asetetaan lisäksi Target Light -objektiksi, sillä se on juuri sillä hetkellä muokkauksen kohteena.



Kuva 6. Osa luomisen ohjelmakoodia.



Kuva 7. Valaisimen luonti VR-näkymässä.

Kuva 7 esittää, miltä näkymä näyttää virtuaalitodellisuudessa, kun valaisin on luotu. Kuvassa näkyy myös ohjaimen laserosoitin, jonka avulla valaisimen sijainti määräytyi.

Toiminnallisuutta testatessa heräsi myös ajatus suoraviivaisemmasta sijoittelusta rotaation puolesta, josta pääsemme seuraavaan ala-aiheeseen.

## 6.2 Poistaminen

Luomistoiminnon vastakohtana toimii valaisimien poistamistoiminto. Kuten jokaisessa luomistyökalussa, on poistaminen aina tarvittava vaihtoehto toiminnoissa. Ylimääräisiä, myöhemmin turhiksi osoittautuvia tai testikappaleita tulee aina tehtyä ja jotta ne eivät jäisi lojumaan maailmaan, tulee valaisimia voida myös poistaa.

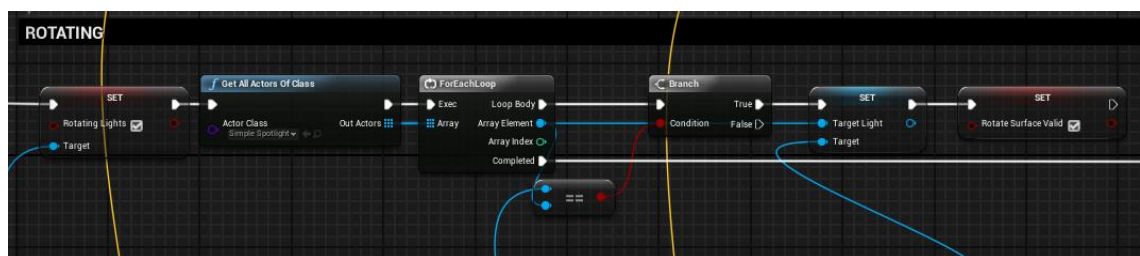
Valaisimien poistaminen onnistuu ainoastaan valojen valitsemisen avulla. Poistamispainiketta painettaessa ohjelma poistaa kaikki valitut valaisimet maailmasta. Tai piilotettuna toimintona itse asiassa vain asettaa ne

näkymättömiksi Undo-Redo -toiminnon mahdollisuuden vuoksi, mutta näennäisesti kyseessä on poistaminen. Aiemmin tekstissä mainittiin VR-näkymässä olevasta valaisimien laskurista. Tämä laskuri ehkäisee myös sitä, ettei käyttäjä huomaamattaan poista tarpeellisia valaisimia turhien joukossa, jos ei esimerkiksi muista jonkin lampun olleen niin ikään valittuna.

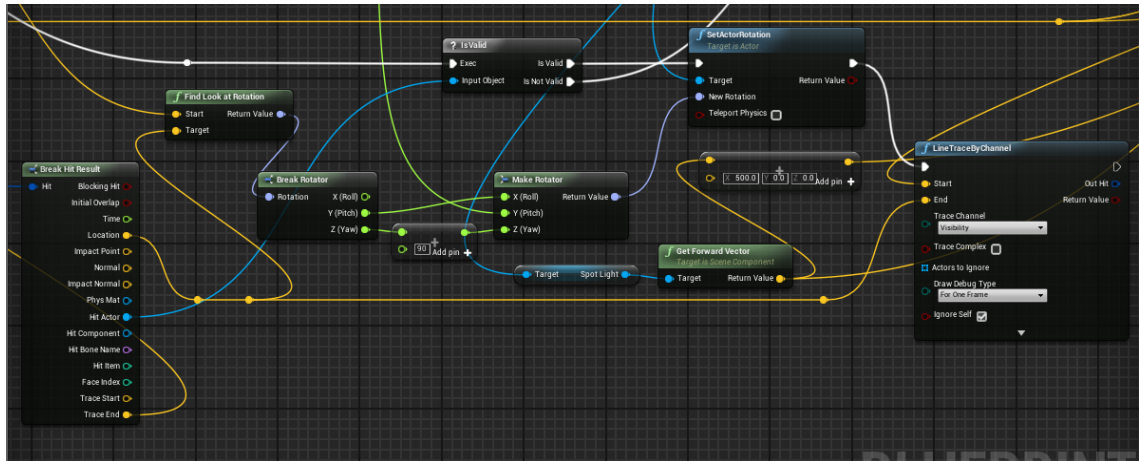
### 6.3 Kääntäminen

Kääntämistoiminnallisuus lienee melko selvä konsepti; ajatuksena on voida kääntää valaisinta haluttuun suuntaan. Kääntämistilassa ollessa käytetään niin ikään ohjaimen laseria apuna. Toiminnallisuuspainiketta painettaessa haluttua valaisinta kohden pysyy valaisin paikallaan, mutta kääntyy seuraten laserosoittimen loppupistettä. Tämä tarkoittaa sitä, että jos ohjaimella osoitetaan esimerkiksi puun runkoa tai rakennuksen seinämää, valaisin kohdentaa valonsa sitä kohden valaisten sen. Valo on helppo saada osoittamaan haluttuun suuntaan ja korkeuteen.

Valaisimen kääntämisen lisäsimme myös ominaisuudeksi suoraan luomistoiminnon perään, mikä nopeuttaa useamman valon luomisprosessia. Valon luotuaan käyttäjä voi pitää toiminnallisuuspainiketta painautuneena, jolloin valo siirtyy saman tien kääntötoimintoon. Käytännössä tämä tarkoittaa sitä, että kun uusi valaisin luodaan, sitä voi myös kääntää haluttuun suuntaan saman tien ilman vaatimusta siirtyä valaisimen kääntötilaan erikseen. Näin uusi valaisin saadaan saman tien niin haluttuun sijaan, kuin myös osoittamaan haluttuun suuntaan. Jälkeenpäin kääntäminen tapahtuu kuitenkin vain kääntötilassa.



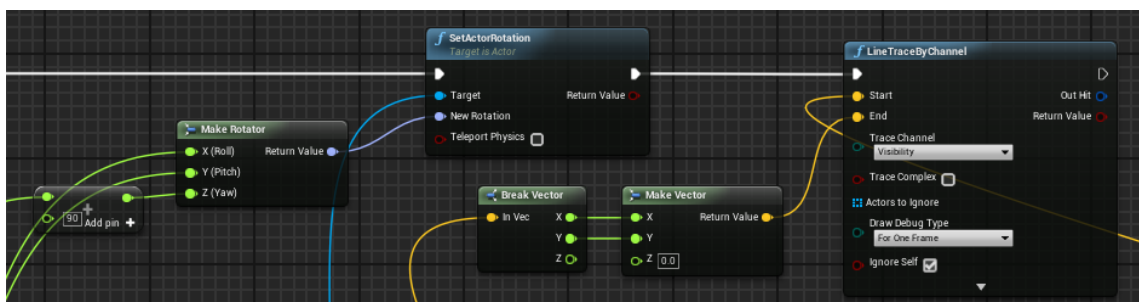
Kuva 8. Osa kääntämisen ohjelmakoodista.



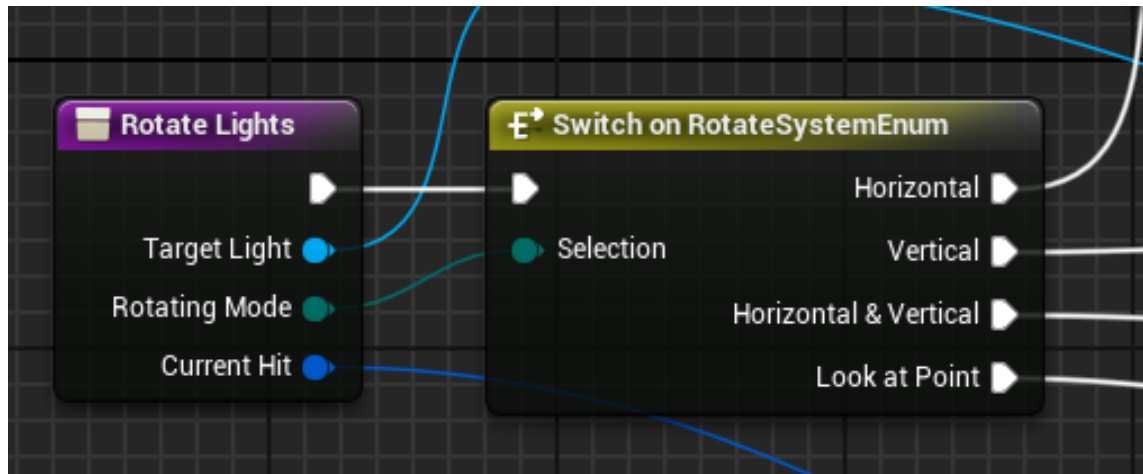
Kuva 9. Osa kääntämisen ohjelmakoodista.

Kuvassa 8 näkyvässä koodinpätkässä ohjelma tarkastaa, mikä maailmassa olevista valaisimista on käyttäjän kohteena ja asettaa sen Target Lightiksi. Kuvassa 9 taas on ohjelmakoodia valaisimen omasta luokasta, joka lähtee pyörimään jokaisessa valitussa valaisimessa kääntämisprosessin käynnistyttyä. Valaisin hakee ensin laserosoitimen osoittaman kohteen, ja mikäli se on validi, eli kohde löytyy, SetActorRotation hakee kääntämistä varten XYZ-arvot. FindLookAtRotation-solmuun asetetaan input-arvoiksi Start-kohtaan valaisimen sijainti ja Target-kohtaan laserosoitimen osumakohta, jolloin valo kohdistuu tähän pisteeseen.

Tällä tapaa valaisin kohdistaa valon niin pysty- kuin vaakasuunnassakin. Kääntäminen onnistuu myös vain horisontaalisessa tai vertikaalisessa suunnassa, mikäli käyttäjä haluaa lukita toisen suunnan arvon (kuva 10). Tällöin ohjelmakoodi hakee valaisimen omat arvot ja muuttaa vain yhtä akselia Target-pointin mukaisesti.



Kuva 10. Kääntäminen horisontaalisesti.

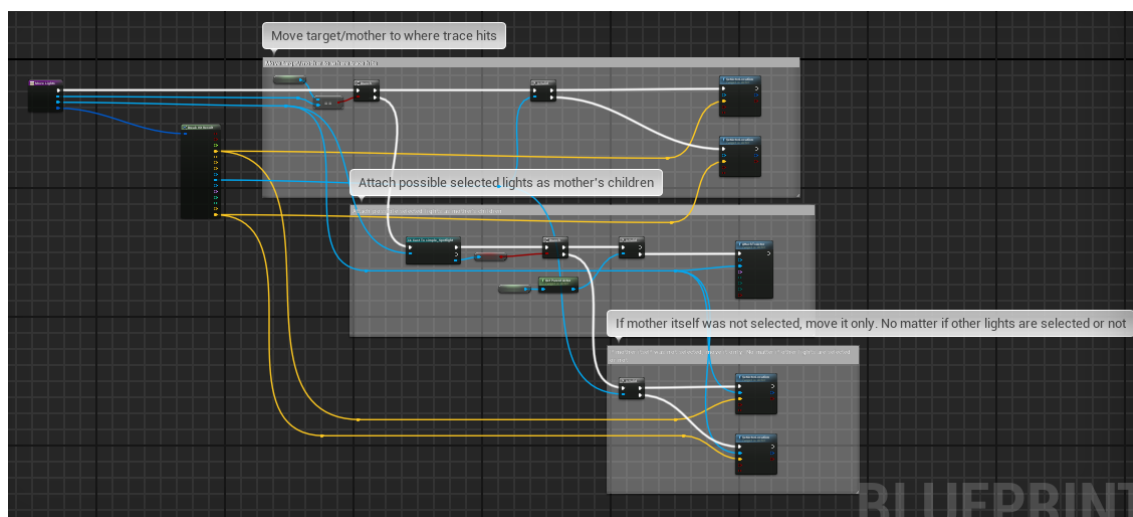


Kuva 11. Rotate Lights -olion alku, joka ensimmäisenä katsoo valitun kääntämisoperaation ja siirtyy sen mukaiseen toimintoon.

Kuvissa 10 esitetään tarkemmin ohjelmakoodia, kun kyseessä on horisontaalinen kääntäminen. Ohjelmassa yksi akseli pidetään staattisena, jotta valaisinobjekti pysyy vertikaalisessa suunnassa kohdillaan. Kahden muun akselin arvot ohjelma hakee sen mukaan, mihin käyttäjä laserosoitimellaan osoittaa. Ohjelma on ensimmäiseksi hakenut, millaisen kääntämisoperaation käyttäjä on valinnut ja siirtyy sen mukaan suorittamaan haluttua oikeaa toimintoa (kuva 11).

#### 6.4 Siirtäminen

Siirtotoiminto on ajatuksena hyvin samanlainen kääntötoiminnon kanssa. Kun käyttäjä on siirtotilassa ja valitsee halutun valaisimen, toiminnallisuuspainiketta painaessa valaisin siirtyy laserosoitimen osoittamaan sijaintiin. Valaisin ei suinkaan seuraa osoittimen päätepistettä, vaan sen ja tason pintojen törmäämispistettä. Valaisimen sijoittelussa toimii sama periaate kuin luonnissakin, eli sitä ei tule voida asettaa noin vain mihin sattuu, vaan kohteen on aina oltava sopiva valaisimen sijainnille.

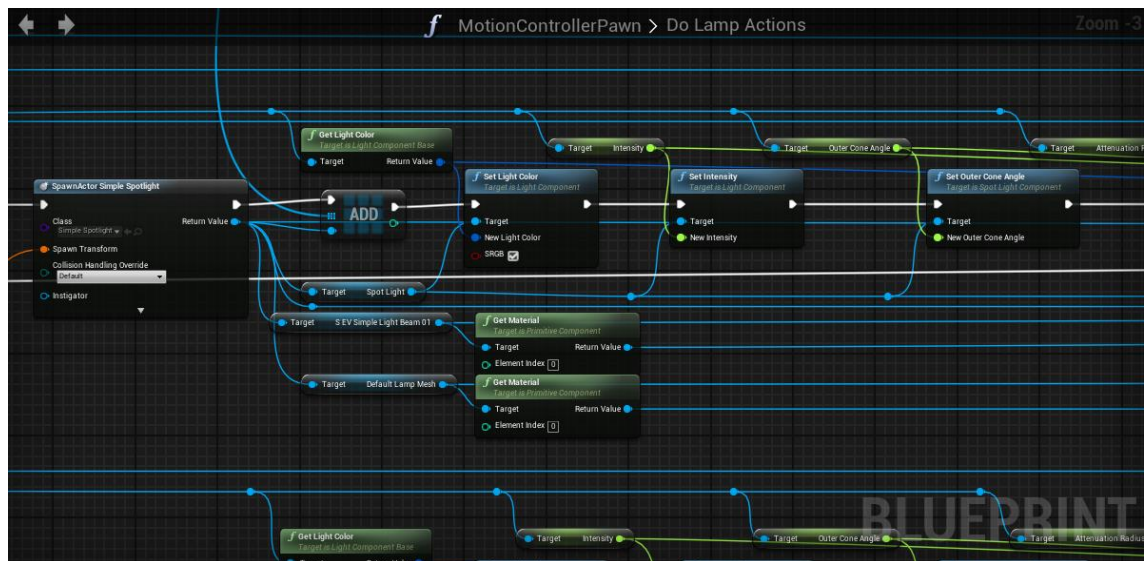


Kuva 12. Siirtämisen ohjelmakoodi.

Siirtäminen on toimintona yksinkertaisin ohjelmoinnin puolesta. Kuvassa 12 esitetään, kuinka siirtäminen ohjelmassa rakentuu. Se tarkastaa, oliko käyttäjän valitsema valo jo ennestään valittuihin valoihin kuuluva. Mikäli kyllä, kyseinen valaisin asetetaan Mother Lightiksi ja muut valitut valot sen lapsiksi. Näin koko valittujen valaisimien ryhmä siirtyy tämän valon mukana. Mikäli taas ei, tämä valaisin siirtyy yksinään eivätkä muut valot seuraa sitä. Tämä ehkäisee valitsemisessa turhaa työtä. Ohjelma hakee laserosoittimen ja kohteen pinnan osumakohdan XYZ-arvot ja asettaa valon siihen sijaintiin. Mahdolliset lapsivalot sijoittuvat suhteessa yhtä kauas Mother Lightista kuin ne olivat ennen siirtoakin, jolloin valoryhmän keskinäiset etäisyydet eivät muutu laisinkaan.

## 6.5 Kopioiminen

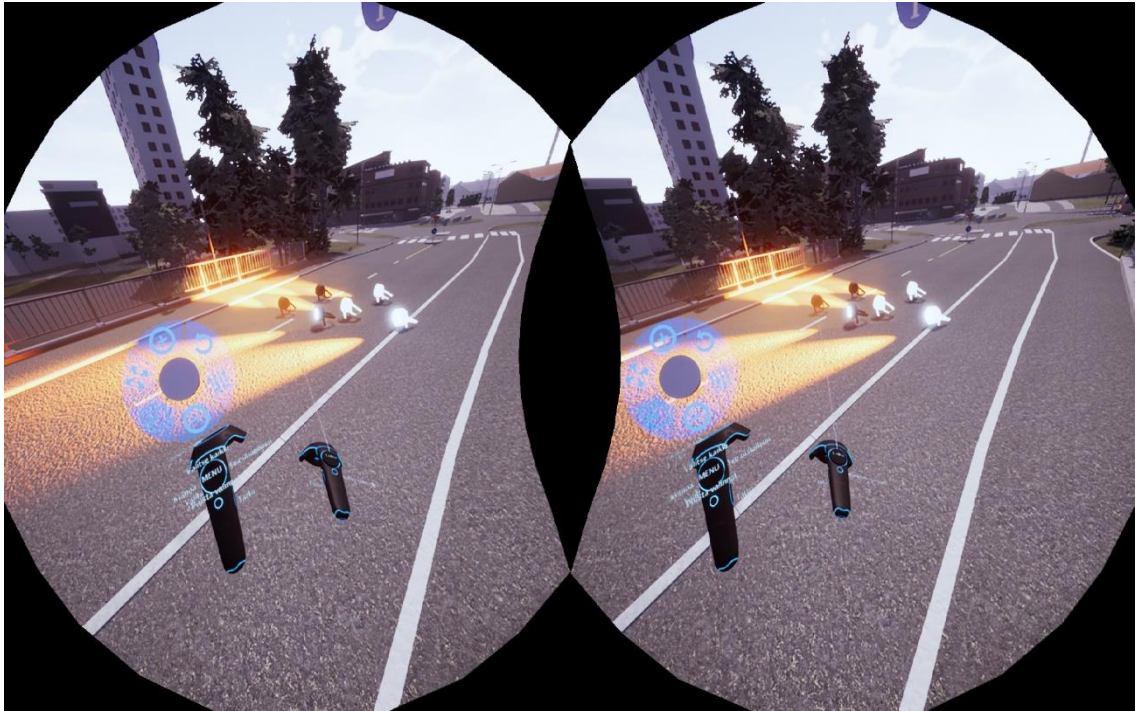
Kopioimistoiminnon periaate on synnyttää jo maailmassa olevan valaisimen kanssa identtisiä valaisimia. Kopioitu valaisin perii alkuperäiseltä vastineeltaan kaiken: suunnan mihin se osoittaa ja kaikki valon ominaisuudet, eli värin, kirkkouden ja keilan kantavuuden. Ainostaan sijaintia se ei ota muistiin, sillä sijainti määräytyy jälleen laserosoittimen mukaan; kopioidun valaisimen voi saman tien siirtää uuteen paikkaan, esimerkiksi alkuperäisen viereen, eikä se siis synny sen päälle. Myös valaisinryhmiä voi kopioida, jolloin valaisimien luonti nopeutuu, jos samankaltaisia ryhmittymiä haluaa useampaan kohteeseen.



Kuva 13. Osa kopioimisen ohjelmakoodista.

Kuvassa 13 esitetään sitä, mitä kopioimisen ohjelmakoodissa tapahtuu. Kopioimisessa luodaan aina uusi valaisin, mutta sen default-arvot muutetaan luomisen jälkeen vastaamaan alkuperäisen valaisimen omia arvoja. GetLightColor-solmu hakee alkuperäisen valaisimen valon värin ja SetLightColor puolestaan asettaa tämän väriarvon uuden, kopioidun lampun värin arvoksi. Tällä tapaa ohjelmakoodi käy läpi kaikki tarvitsemansa arvot, hakien ne ensin alkuperäiseltä valaisimelta ja asettaa nämä arvot uuden valaisimen arvoiksi.

Kuvassa 14 nähdään kopioimista VR-näkymässä. Valkoisena loistavat valo-objektit ovat kopioitavia valoja ja mustat niiden identtisiä kopioita, joita käyttäjä on laserosoitimen avulla asettamassa paikalleen.

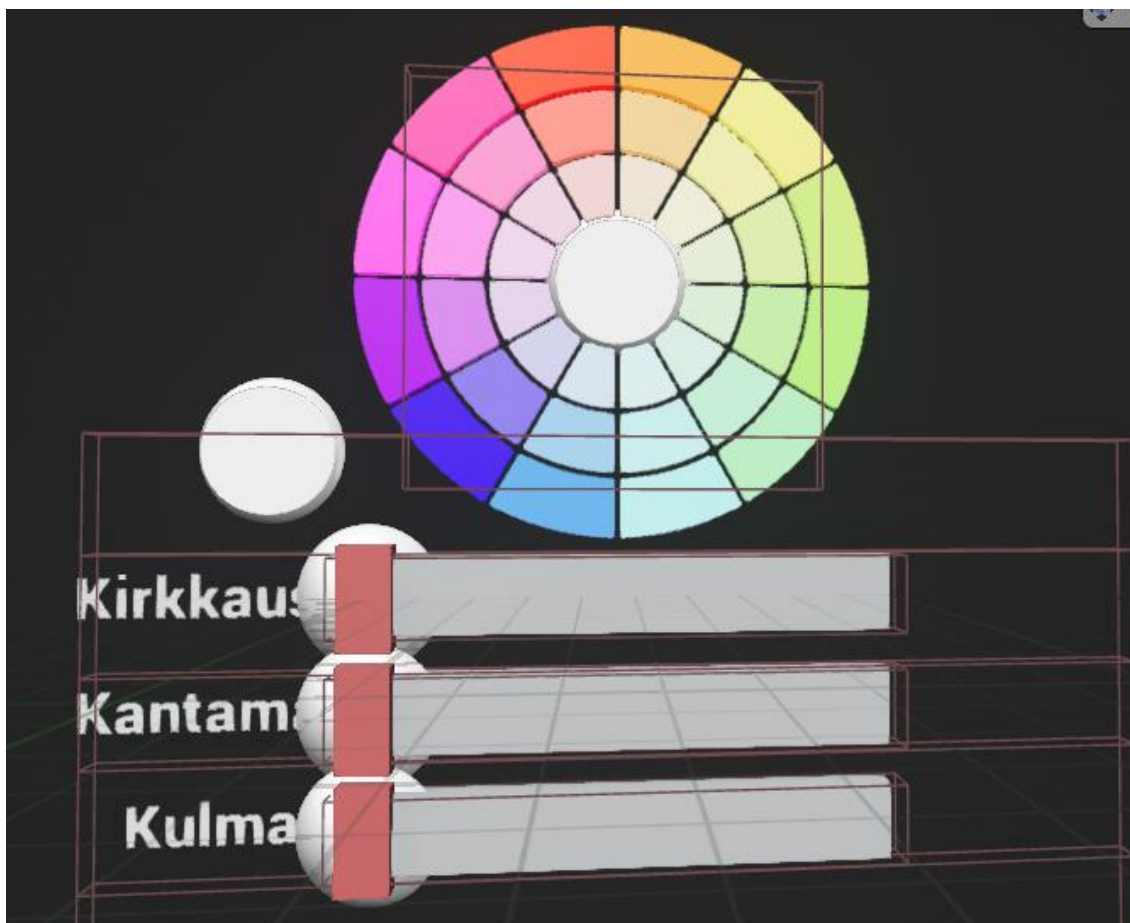


Kuva 14. Valittujen valaisimien ryhmä kopioituna niiden viereen.

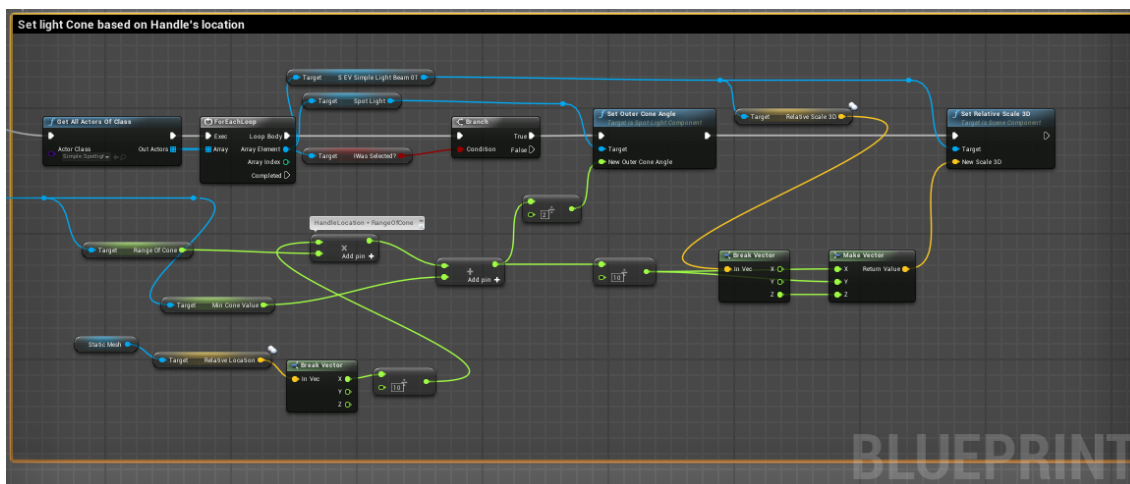
## 6.6 Mukauttaminen

Mukauttaminen on luodun valaisimen valon ominaisuuksien muokkaamista. Valon väriä, kirkkautta sekä valokeilan pituutta ja kulman suuruutta voi säädellä tiettyjen rajoitusten sisällä.

Väriä voi valita 37:stä värilaatasta valitsemalla. Kirkkauden sekä valokeilan pituuden, eli kantaman, ja kulman voi määrittellä liukuvien säädinten avulla ja niiden asettamien rajoitusten mukaan. Säädinten rajoitukset on asetettu reaalisten arvojen mukaisesti, jotta arvoja ei voi asettaa aivan mielivaltaisen suuriksi tai pieniksi. Nämä ovat esitettynä kuvassa 15.



Kuva 15. Mukauttamista varten luodut 3D-väriympyrä ja -säätimet.

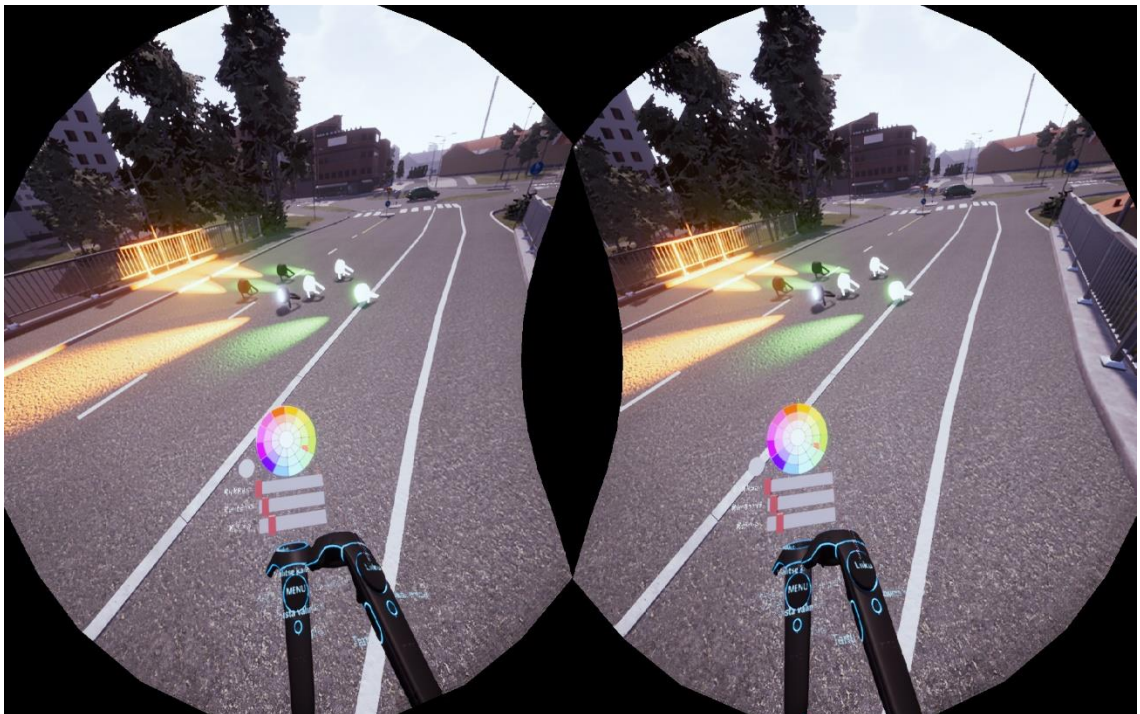


Kuva 16. Osa mukauttamisen ohjelmakoodista.

Kuvassa 16 esitetään pieni pätkä mukauttamista varten rakennetusta ohjelmakoodista. Pätkässä asetetaan valokeilan koko sen mukaan, mihin kulman säätimen kahva on asetettu suhteessa säätimen pituuteen. Sovellus hakee ensin kaikki valitut valaisimet ja laskee näille valaisimille niiden valokeilan kulman

arvon. Arvo lasketaan kertomalla ensin keskenään kulman raja-arvojen erotus ja kahvan sijainnin arvo suhteessa säätimen pituuteen. Tuloon lisätään valokeilan kulman minimiarvoksi asetettu luku ja summa jaetaan kahdella.

Tähän samaan tyyliin tapahtuu kirkkauden ja kantaman arvon vaihtuminen, eli luku lasketaan aina kahvan sijainnista säätimen pituuteen nähden. Väriarvo sen sijaan on yksinkertainen toiminto; käyttäjä painaa haluamaansa väripainiketta laserosoitimella, jolloin sovellus vaihtaa jokaisen valitun valaisimen väriarvoksi kyseisen väripainikkeen värin.



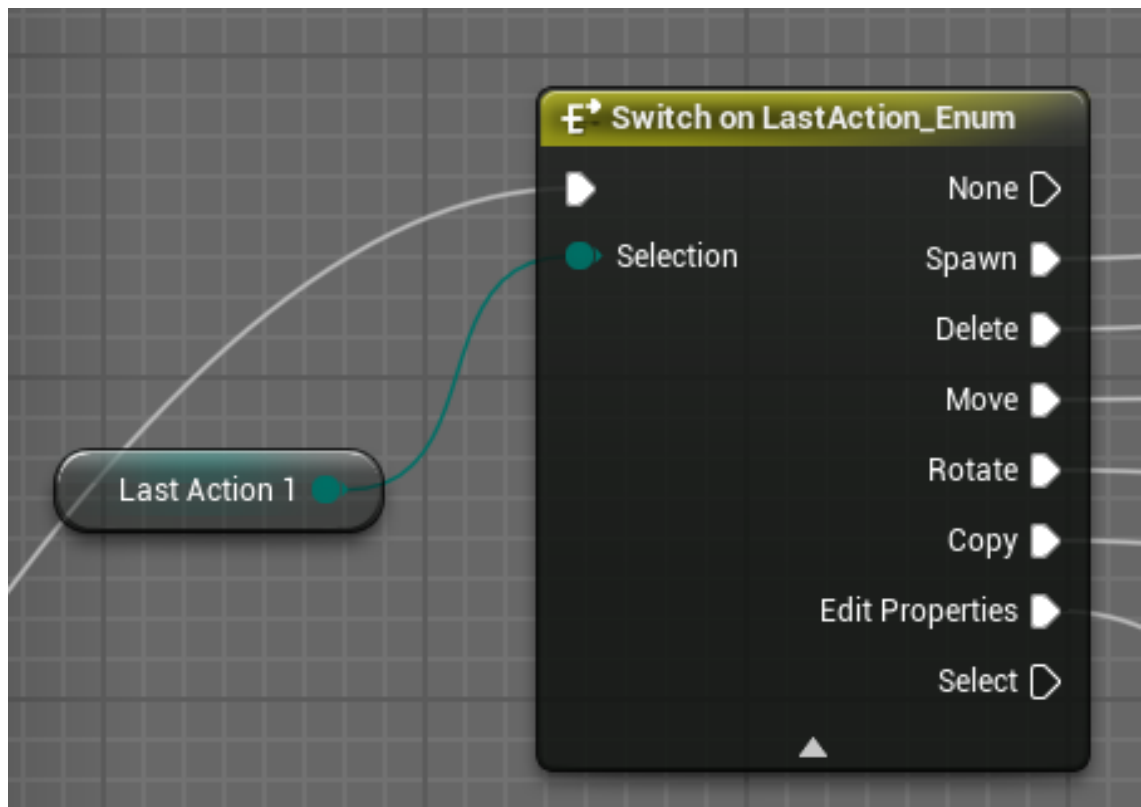
Kuva 17. Mukauttaminen VR-tilassa.

Yllä olevassa kuvassa 17 nähdään mukauttamista konkreettisesti. Kuvassa on esillä mukauttamisen valikko, jonka arvoja on säädely. Kuten maailmaan asetetuista valaisimista voi huomata, on valaisimilla eri arvot. Valituilla, eli valkoisena loistavilla valoilla, on valon värinä vihreä ja niiden valon kantomatka on huomattavasti lyhyempi kuin oranssia valoa heittäville valaisimilla. Nämä muutokset on saatu aikaan valitsemalla oikeanpuoleinen kolmen valaisimen ryhmä ja mukauttamalla näiden valaisimien arvoja. Toiset kolme valoa eivät arvojaan muuta, vaan ne pysyvät entisellään.

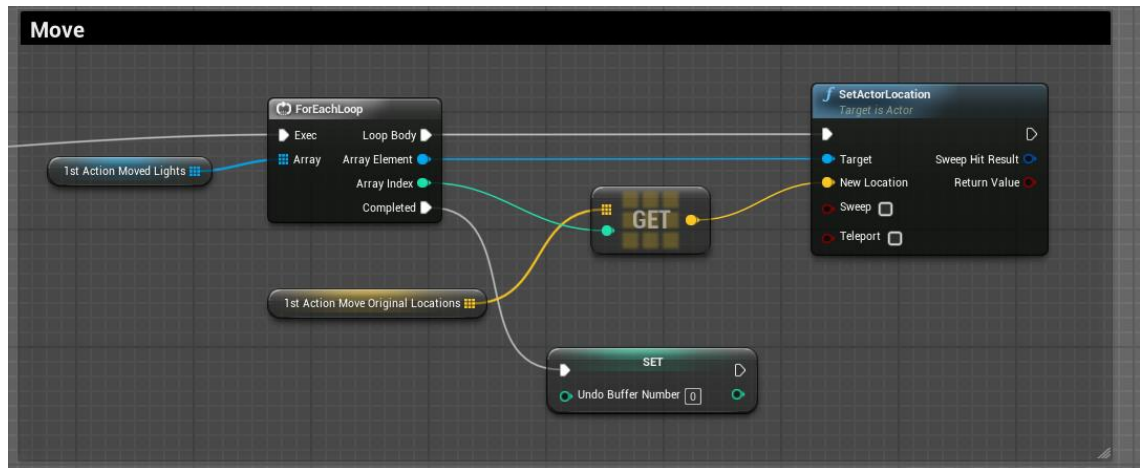
## 6.7 Undo ja Redo

Kumoaminen on yleinen toiminto hyvin suuressa osassa sovelluksia ja tarpeellinen etenkin juuri luomistyökaluissa. Tässäkin sovelluksessa valaisimien kanssa voi tehdä virheitä, jotka on hyvä saada kumotuksi tai tehdyksi uudestaan, jos käyttäjä haluaa verrata vaihtoehtoja.

Toimintoa varten rakennettiin kolmen tilan järjestelmä. Tämä tarkoittaa sitä, että sovelluksella on kaiken aikaa muistissa kolmen viimeisimmän muutoksen arvot, joiden välillä käyttäjä voi vaihdella. Viimeisimpänä toteutettu toiminto tai muutos ottaa aina ensimmäisen tilan tässä järjestelmässä ja siirtää kahta edellistä toimintoa tai muutosta yhdellä seuraavaan tilaan. Kolmannessa tilassa olleet arvot taas poistuvat kokonaan muistista uusien tieltä, eikä niihin arvoihin voi enää palata kumoamalla.



Kuva 18. Osa Undo-Redo -ohjelmakoodista.



Kuva 19. Osa Undo-Redo -ohjelmakoodista.

Kuvissa 18 ja 19 esitetään kumoamisen toiminnon ohjelmakoodia. Aina uuden toiminnon toteuttamisessa tämä toiminto tallennetaan Last Action 1 -muuttujaan, jonka mukaan sovellus siirtyy suorittamaan tiettyä osaa tuotetusta ohjelmakoodista. Esimerkiksi jos käyttäjä siirtää valaisinta, tämä toiminto tallentuu. Sovellus asettaa Last Action 1 -muuttujan arvoksi Move, jolloin ohjelmakoodi tietää, että viimeksi tapahtunut toiminto on siirto. Näin ollen, jos käyttäjä päätyy kumoamaan viimeksi tekemänsä toiminnon, ohjelma tarkastaa Last Action 1 -muuttujan arvon ja siirtyy suorittamaan kuvassa 19 näkyvää koodinpätkää. Sovellus hakee kaikki viimeisimmässä toiminnossa siirretyt valoobjektit ja asettaa näiden kaikkien sijainniksi ne XYZ-arvot, jotka on tallennettu 1st Action Move Original Locations -muuttujan taulukkoon. Näin valaisimet siirtyvät edellisiin sijainteihinsa.

Käyttäjän painaessa Redo-painiketta on periaate täysin samanlainen, mutta arvot on tallennettu Undo-toimintoa tehdessä. Valaisimet siirtyvät siis uusiin sijainteihinsa, niihin joihin käyttäjä valot asetti niitä siirtäessään.

## 7 Lopputulos

Valmiissa valaistuksensuunnittelutyökalussa on siis kuusi toimintoa ja lisäksi Undo-Redo -toiminto. Kaikki toiminnot ovat toimintakunnossa ja näin ollen valaistuksensuunnittelutyökalu vastaa pääasiallisia tavoitteitaan. Se on toimiva työkalu, jolla voi luoda 3D-valaisinobjekteja virtuaalitodellisuusmaailmaan ja muokata näiden valaisinobjektien ominaisuuksia, sijaintia ja suuntaa.

Jo kerrottujen ominaisuuksien ja toiminnallisuuksien lisäksi sovellukseen luotiin tallennus- ja lataamistoiminnot. Suunnittelutyö ei aina ole helppo tai nopea prosessi, vaan vaatii usein enemmän kuin yhden session valmistumiseen. Käyttäjä voi tallentaa sen aikaisen sessionsa, jotta kaikki, mitä hän on työkalulla tehnyt, tallentuu tietokoneen muistiin. Tallennustiloja on kolme, eli käyttäjä voi pitää yhtä aikaa tallessa kolme eri suunnittelutyötä, mikäli kokee sen tarpeelliseksi. Tallentaminen ja myös lataaminen onnistuu valikon kautta, mutta lisäksi sovellus lataa automaattisesti viimeisimmän tallennuskohteen tiedot käyttäjän käynnistäessä sovelluksen. Hän voi siis tallentaa keskeneräisen työnsä, sammuttaa sovelluksen ja jatkaa samasta luomistyöstä esimerkiksi seuraavana päivänä uudestaan.

Toimintojen toteutuksen aikana suurimmat bugit ja virheet korjattiin, mutta aivan kaikista ei ehditty päästä eroon. Toteutuksessa on näin ollen hieman virheitä, mutta ei niin sanotusti sovellusta rikkovia tai työtä liikaa häiritseviä. Bugit ovat pääasiassa kauneusvirheitä, sovelluksen toimivuus ei niistä kärsi.

Sovellusta testattiin paljon toteutuksen aikana ja sille toteutettiin myös niin kutsuttu stressitesti. Sovelluksessa luotiin ensin 50 valaisinobjektia ja sen jälkeen 100. Toimivuus hidastui, mutta sovellus ei kaatunut. Kaikki luodut valo-objektit valittiin aktiivisiksi valoiksi ja niiden tuottaman valon kirkkautta ja kantamaa mukautettiin. Vasta kun valokeilat asetettiin äärimmilleen, ei tietokone enää kestänyt risteävien valojen tuottamaa visuaalista taakkaa, vaan sovellus kaatui. Tällainen äärimmäisyys ja ylitsevuotava valon määrä ei tietystikään ole suunnittelutyössä tarkoitus, joten kaatumisesta huolimatta testi koettiin

onnistuneeksi, sillä valoja pystyi asettamaan suuria määriä tarkoituksenmukaisilla asetuksilla.



Kuva 20. Karelia AMK:n opiskelija Tuisku Saarelainen käyttämässä HTC Vive -laitetta ja valaistuksensuunnittelutyökalua.

## 8 Pohdinta

Valaistuksensuunnittelutyökalu oli työläs prosessi erityisesti sen vuoksi, että sen tekoon ryhtyessäni vastaan tuli todella paljon uusia asioita, joihin tuli tutustua ennen varsinaiseen työhön ryhtymistä. Esimerkiksi VR-laitteistoa en ollut koskaan edes läheltä nähnyt, Unreal Engine ei ollut laisinkaan ennestään tuttu ohjelma ja virtuaalitodellisuuden käyttöön ja soveltamiseen UE-projektissa meni

oma hetkensä. Tästä kaikesta huolimatta työkalun toteutusvaihe alkoi ja sujui hyvin. Matkan varrella jotkin suunnitellut asiat muuttuivat ja uusia ideoita syntyi kesken kaiken. Nämä tietysti pitkittivät toteutuksen aikataulua, mutta tekivät työkalusta aina aiempaa paremman ja käytännöllisemmän. Jotkin ideat päätyivät lopulliseen tuotokseen, kun taas jotkin jäivät jatkokehitysideoiksi.

Haasteista huolimatta projektin työstäminen osoittautui mukavaksi ja erityisesti opettavaiseksi. Kyseessä on yksi mielekkäimmistä projekteistani ja olen varsin tyytyväinen itse lopputulokseen, vaikka huomaankin siinä epäkohtia ja kehittämisen mahdollisuuksia. Näiden huomaaminen voi olla hyväkin asia, sillä se osoittanee työkalun mallin kehittyneen mielessäni, kuten myös kykyni ja tapani ohjelmoida. Työkaluun päätyi ohjelmakoodia, jonka tiedän osaavani tämän kirjoittamisen hetkellä paremmin ja optimaalisemmin. Oppia työkalun parissa sainkin runsaasti.

## **8.1 Vertailu tavoitteisiin**

Työn tavoitteena oli toteuttaa virtuaalitodellisuudessa toimiva työkalu, jonka avulla luoda ja mukauttaa valoja. Nämä valot mukailevat oikean elämän valotaiteessa käytettyjä valoelementtejä, jotta työkalun avulla olisi mahdollista suunnitella valotaiteeseen tarkoitettua valaistusta.

Tämä tavoite toteutui. Valmistuneen tuotteen avulla käyttäjä on kykeneväinen tekemään jo mainittuja toimintoja virtuaalitodellisuudessa ja se luonnistuu näin ollen valaistuksen suunnitteluun. Tavoitteita lähdettiin toteuttamaan niin, että tärkeimpänä olisi saada kaikki toiminnallisuudet toimintaan ja jotakin kautta aktiivisiksi. Helppokäyttöisyys oli toisarvoinen tavoite, mutta mielestäni työkalu on myös sitä. Se on ehkä hieman raskas käyttää pitemmän aikaa, sillä toimintojen aktivoiminen on hitaanlaista. Valikon käyttö ja käyttäjän oma VR-käyttäytyminen toki nopeutuvat suhteessa aikaan, mitä hän työkalun parissa käyttää.

Aikataulullisestikin tavoite saatiin toteutettua. Työkalu valmistui oman aikataulun puitteissa, eikä sen valmistuttua tarvinnut enää ryhtyä jatkotyöstämään tai

korjailemaan toimivaa sovellusta. Tavoitteisiin nähden olen erittäin tyytyväinen lopputulokseen.

## 8.2 Opinnäytetyön haasteet ja ammatillinen kehitys

Varsinaisen opinnäytetyön aiheen keksiminen ei tuottanut vaikeuksia, sillä jo työharjoitteluni alettua olin melko varma, että tahdon tehdä opinnäytetyön, joka liittyy siihen, mitä siellä pääsin tekemään. Varsinaiset haasteet tulivat vastaan työn alkupuolella ja raporttia kirjoittaessa.

Kuten todettua, lähtökohdat opinnäytetyön tekovaiheessa olivat hieman haasteelliset, sillä siinä oli paljon osia, jotka eivät olleet minulle ennestään tuttuja. Opin virtuaalitodellisuudesta ja Unreal Engine 4:stä nopeasti ja työnteko alkoi sujua kaiken aikaa helpommin, nopeammin ja rutiininomaisemmin. Opinnäytetyön tietoperustan kirjoitusvaihe tuotti niin ikään hienoisia ongelmia. Joistakin aiheista oli hieman hankala löytää hyviä viitetietoja. Tutkin paljon Karelia-ammattikorkeakoulun Finna-tietokantaa, sekä Itä-Suomen yliopiston UEF-finnaa, mutta esimerkiksi Unreal Enginestä tai pelimoottoreista yleensä löytyi todella vähän kirjatieta. Jouduin näin ollen turvautumaan paljon internetiin, josta luotettavaa tietoa löytyi loppujen lopuksi hyvinkin helposti ja vaivattomasti. Sain paljon uutta tietoa myös jo ennalta tuttuihin asioihin ja vaikka koin tietoperustan kirjoittamisen aluksi hyvin hankalaksi ja ongelmalliseksi, alkoi se lopulta käydä sujuvaksi. Toteutusvaiheen kirjoitus kävi joka tapauksessa huomattavasti helpommin. Tekovaiheet oli helppo kuvata ja kertoa, mitä milloinkin oli tarkoitus tapahtua ja kuinka se toteutettiin.

Koen kehittyneeni tämän työkalun työstämisen aikana valtavasti. Koulussa kärsin jatkuvasti itsetunto-ongelmista ohjelmoijana, mutta työn parissa tunsin oppivani kaiken aikaa, rento ja mukava työilmapiiri edesauttoivat sitä. Blueprintit osoittautuivat minulle sopivaksi ohjelmointityyliksi ja huomasin pitäväni niiden avulla ohjelmoinnista huomattavasti. Jo koulussa opittuja asioita aloin ymmärtää paremmin työtä tekemällä ja blueprinttien kautta niihin syventymällä. Edistyessä törmäsi jatkuvasti uusiin ongelmiin, joihin piti etsiä ratkaisuja internetistä, mutta

pikkuhiljaa tämäkin väheni vähenemistään, kun alkoi osata pohtia ratkaisuja paremmin itse jatkuvan avun etsinnän sijaan. Unreal Engine 4 kävi myös ohjelmana hyvin tutuksi ja opin myös erittäin paljon sen käytöstä muutenkin kuin ohjelmoinnin kannalta. Koska työkalu oli osa suurempaa visuaalista projektia, opin myös kokonaisen projektin tuotosta paljon varsinaisen työni ohella. Työkalun teon jälkeen koen olevani huomattavasti parempi ohjelmoija. Tiedän myös selviytyväni paremmin ohjelmointiin liittyvistä ongelmista ja laaja kokemus olio-ohjelmoinnista sai minut ymmärtämään sitä syvemmin kuin koulun tunneilla. ”Tekemällä oppii” osoittautui tässä tapauksessa todenmukaiseksi sanonnaksi, sillä koen oppineeni työn parissa paljon enemmän kuin koulussa, vaikka koulussa hyvät alkeet saikin.

Sekä työkalun teko-, että opinnäytetyön raportinkirjoitusvaihe osoittivat aikataulutuksen tärkeyttä. Työkalun teossa käytössä oli Kanban-taulu, jossa työtehtävät olivat selkeästi näkyvillä ja jota käytiin viikoittain lävitse. Aikataulutus sujui sen myötä paremmin kuin ennen Kanbanin käyttöönottoa. Kirjoitusvaiheessa en varsinaista aikataulua itselleni luonut, mikä varmasti onkin osatekijänä siihen, että kirjoitus lykkääntyi lykkääntymistään. Ryhdyin aikatauluttamaan kirjoitustyötäni vasta myöhemmin, jolloin se alkoi myös edistyä huomattavasti paremmin ja sutjakkaammin, järjestelmällisemmin. Aikataulutuksesta oli suuresti hyötyä myös henkisellä puolella. Kun teki juuri sen verran asioita kuin oli itselleen määritellyt päivän tavoitteeksi, pystyi hyvillä mielin tekemään muita juttuja loppupäivän ajan, eikä tarvinnut potea omantunnontuskia.

Aikataulutuksen tärkeyden pyrin huomioimaan tarpeen tullen jatkossa paremmin ja aion myös suunnitella sen heti alusta alkaen tulevissa projekteissani, mikäli tarpeellista.

### 8.3 Kehitysideat

Vaikka toteutus täytti asetetut tavoitteet ja lopputuloksena on toimiva työkalu, on valmis tuotos herättänyt muutaman ajatuksen, joiden mukaan kehittää sitä entisestään. Ensimmäisenä mainittakoon kääntämistoiminto. Kääntäminen tapahtui siis laserosoittimella osoittamalla, jolloin valaisin kääntyy kohti osoittimen päätepistettä. Tämä ei kuitenkaan aina ole kovin tarkka tapa, sillä mikäli osoitin ei osu mihinkään, esimerkiksi puuhun tai rakennukseen, se jatkuu kovin pitkän matkaa. Tällöin valaisimen kohdistamisesta tulee epätarkkaa. Vielä epätarkemman kääntämisestä tekee, jos kohteena on useampi valaisinobjekti. Jokainen valituista valaisimista kohdistuu osoittimen päätepisteeseen, mutta käyttäjä ei välttämättä tätä tahto. Työkaluun voisi siis lisätä vielä yhden kääntämistavan, joka saisi valaisimet osoittamaan samaan suuntaan, mutta omassa pisteessään. Tarkoittaen siis sitä, että jokaisesta valaisimesta lähtisi suora viiva esimerkiksi itää kohti ja jokainen niistä kääntyisi täysin samaan suuntaan. Suorassa rivissä olevat valaisimet kääntyisivät siis tismalleen samoin päin, sen sijaan että ne katsoisivat samaa tiettyä pistettä kohti. Tämä lisäisi kääntämistoimintoon variaatiota ja auttaisi käyttäjää saamaan valaisimet juuri haluttuun asentoon.

Undo-Redo -toiminto kaipailisi myös hieman jatkokehitystä. Kolmen tilan kumoaminen on erittäin suppea, eikä loppujen lopuksi tuo käyttäjälle paljonkaan. Vaikka toiminto ei ole niin sanottu avaintoiminnallisuus toteutetussa työkalussa, voisi kumoamiseen tehdä lisätiloja, joihin palata. Ainakin viisi olisi hyvä alku, mutta kumoamisessa on aina käyttäjän puolesta parempi mitä pidemmälle taaksepäin voi palata. Useimmissa tietokonesovelluksissa kumoamisen määrän raja tulee harvoin vastaan.

Yksi kehitysidea on itse asiassa jo toteutuksen aikana kokeiltu ajatus siitä, että varsinaisen valikon lisäksi käytössä olisivat pikanäppäimet toimintojen saamiseksi aktiiviseksi. Nykypäivänä pikanäppäimet ovat tärkeä osa mitä tahansa sovellusta. Lähestulkoon jokaiselle tietokoneen käyttäjälle esimerkiksi CTRL-S on tuttu näppäinyhdistelmä tiedoston tallentamiselle. Valikko toimisi ensisijaisena käytävänä valaisimien toimintojen löytämiseen, mutta käyttäjä voisi

oppia myös tiettyjä painikkeita ohjaimesta painamalla pääsemään käsiksi eri valaisinten ominaisuuksiin. Tämä olisi omiaan nopeuttamaan valaisinobjektien mukauttamista erityisesti suurkäyttäjillä.

Kaiken kaikkiaan työkalussa on pientä parantamisen varaa ja viimeistelyä, jotta työkalusta saisi parhaiten irti kaiken vaadittavan.

## Lähteet

- Bates-Brkljac, N. 2012. Computer Science, Technology and Applications: Virtual Reality. New York: Nova Science Publishers.
- Epic Games. 2018. Leading the Way in Virtual Reality. <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>. 4.6.2018.
- Epic Games. 2018. Unreal Engine Documentation: Blueprints Visual Scripting. <https://docs.unrealengine.com/en-us/Engine/Blueprints>. 5.6.2018.
- Epic Games. 2018. Unreal Engine Documentation: Spot Light. <https://docs.unrealengine.com/en-US/engine/rendering/lightingandshadows/lighttypes/spot>. 11.8.2018.
- Eva, K. 2017. University Of Leeds Conducts Virtual Reality Tests On Young Children. VRFocus. <https://www.vrfocus.com/2017/10/leeds-university-conducts-virtual-reality-tests-on-young-children/>. 7.8.2018.
- Freeman, D. 2008. Studying and Treating Schizophrenia Using Virtual Reality: A New Paradigm. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2486455/>. 15.6.2018.
- Game Designing. 2018. The Top 10 Video Game Engines. <https://www.gamedesigning.org/career/video-game-engines>. 8.6.2018.
- Hintikka, K. A., Kojo, I. & Metsämäki, M. 1998. Virtuaaliympäristöjen suunnitteluopas. Helsinki: Edita.
- Holly, R. 2016. Everything you need to know about HTC Vive. VRHeads. <https://www.vrheads.com/everything-you-need-know-about-htc-vive>. 7.8.2018.
- HTC Corporation. 2018. Vive VR System. <https://www.vive.com/us/product/vive-virtual-reality-system/>. 8.8.2018.
- Jay. 2016. Field of View for Virtual Reality Headsets Explained. VR LENS LAB. <https://vr-lens-lab.com/field-of-view-for-virtual-reality-headsets/>. 7.8.2018.
- NVIDIA Corporation. 2018. DirectX 11 Technology. <https://www.geforce.com/hardware/technology/dx11/technology>. 12.8.2018.

- Salovaara, J. 2007. Valotaiteesta: Valoinstallaatio 7 hengen porttia suhteessa valotaiteen historiaan. Tampereen ammattikorkeakoulu. Viestinnän koulutusohjelma. Tutkintotyö.  
<https://www.theseus.fi/bitstream/handle/10024/10563/Salovaara%20%20Jenni.pdf?sequence=2&isAllowed=y>. 15.5.2018.
- Sony Interactive Entertainment. 2016. PlayStationVR Käyttöohje.  
[https://www.playstation.com/fi-fi/content/dam/support/manuals/scee/web-manuals/ps-vr/PSVR\\_Instruction\\_Manual\\_FIN\\_\(SCA\)\\_Web.pdf/](https://www.playstation.com/fi-fi/content/dam/support/manuals/scee/web-manuals/ps-vr/PSVR_Instruction_Manual_FIN_(SCA)_Web.pdf/). 7.8.2018.
- Tapanainen, N. 2017. Vivemoose, virtuaalinen ampumaratasimulaattori. Tietotekniikka. Opinnäytetyö.  
[https://www.theseus.fi/bitstream/handle/10024/125411/Tapanainen\\_Nikke.pdf?sequence=1&isAllowed=y](https://www.theseus.fi/bitstream/handle/10024/125411/Tapanainen_Nikke.pdf?sequence=1&isAllowed=y). 13.6.2018.
- The Franklin Institute. 2018. History of Virtual Reality. <https://www.fi.edu/virtual-reality/history-of-virtual-reality>. 14.6.2018
- Valoparta Oy. 2017. Luminous Finland 100.  
<http://www.valoparta.com/luminousfinland100/>. 15.5.2018.
- X3DMedia. 2010. What is DirectX 11.  
<https://www.develop3d.com/features/what-is-directx-11>. 12.8.2018.