

KARELIA-AMMATTIKORKEAKOULU
Tieto- ja viestintäteknikan koulutusohjelma

Mika Aaltola
Joonas Hiltunen

RAKENNUKSEN TIETOMALLIIN LIITTYVÄ VUOROVAIKUTUS
VIRTUAALITODELLISUUSJÄRJESTELMÄSSÄ

Opinnäytetyö
Joulukuu 2018



OPINNÄYTETYÖ
Joulukuu 2018
Tieto- ja viestintäteknikan koulutusohjelma

Tikkarinne 9
80200 JOENSUU
+358 13 260 600

Tekijä(t)
Mika Aaltola, Joonas Hiltunen

Nimeke
Rakennuksen tietomalliin liittyvä vuorovaikutus virtuaalitodellisuusjärjestelmässä

Toimeksiantaja
Karelia AMK

Tiivistelmä

Tässä opinnäytetyössä tutkittiin virtuaalitodellisuus pohjaisia teknologioita ja toteutettiin niiden pohjalta järjestelmä rakennuksen tietomallin kanssa vuorovaikuttamiseen. Järjestelmä toteutettiin käyttämällä Unity-pelimoottoria, jolla luotiin ohjelma, johon integroitiin BIM Track -projektipankkipalvelu käyttäen BIM Trackin omaa ohjelmointirajapintaa (API). Järjestelmää käytettiin virtuaalitodellisuuteen kykenevällä työpöytäkoneella ja Vive VR -laseilla.

Toteutusta varten saatiin ARKPII-arkkitehtitoimistolta BIM-mallin Tohmajärvelle rakennettavasta Kemien koulusta. Malli ei ollut suoraan yhteen sopiva Unityn kanssa, vaan se piti ensiksi konvertoida mallinnus ohjelmissa Unityn ymmärtämään muotoon. Unityssa malliin tehtiin korjauksia ja sen ympärille ruvettiin rakentamaan kenttää, johon lisättiin ympäristöä, valoja, VR-toiminnallisuus, IBM Watson- ja BIM Track -integraatio. BIM Track- ja Watson -integraatiot vaativat tunnuslukuja: näiden tallentamista ja vaihtamista varten luotiin erillinen aloitusvalikkokenttä, joka käynnistyy ensimmäisenä ja josta käynnistetään kenttä, jossa on koulumalli. BIM Track integraatio ohjelmointiin C#-ohjelmointikielellä käyttämällä Unityn ja BIM Trackin rajapintoja. BIM Track API:a käytetään HTTP-pohjaisilla käskyillä ja tieto siirtyy Json-formaatissa palvelimeen.

Opinnäytetyön lopputuloksena saatiin tehtyä Unityssa VR-ohjelma, johon on integroitu BIM Track -projektipankin toimintoja. Ohjelmassa voi tutkia, tehdä kehitystehtäviä, merkintöjä ja lisätä objekteja Tohmajärvelle tulevan koulun pohjalta tehtyyn 3D-malliin.

Kieli
suomi

Sivuja 72
Liitteet 0
Liitesivumäärä 0

Asiasanat
VR, Unity, BIM, HTTP, BIM Track, projektipankki



THESIS
December 2018
Degree Programme in Information and
Communications Technology

Tikkarinne 9
80200 JOENSUU FINLAND
+ 358 13 260 600

Author (s)
Mika Aaltola, Joonas Hiltunen

Title
Interaction Related to Building Information Model in a Virtual Reality System

Commissioned by
Karelia AMK

Abstract

In this thesis, virtual reality (VR) technologies were studied and applied to a system that interacts with the building information model (BIM). The system was implemented using the Unity game engine by which a program was made to integrate BIM Track project bank service utilizing BIM Track's own application programming interface (API). The system was used with VR-capable desktop computer and Vive VR glasses.

For the implementation, Architectural Office ARKPII gave a BIM model of the school of Kemie to be built in Tohmajärvi. The model was of file type RVT and had to be converted to FBX format in order to be imported to Unity. The model in Unity was slightly fixed and around it terrain, lights, VR functionality, IBM Watson and BIM Track integrations were implemented. BIM Track and Watson integrations required tokens. Separate main menu scene was created for these tokens to be saved or updated. The school scene can be launched from the main menu. BIMTrack integration was made with C # Programming Language and by using Unity and the API of BIM Track. BIM Track API is used with HTTP-based commands and the data is transmitted in Json format.

As a result of this thesis, a VR program was created with integrated BIM Track project bank's functions. The program allows one to explore, make development tasks, markings, and add objects to the 3D upcoming school model.

Language
Finnish

Pages 72
Appendices 0
Pages of Appendices 0

Keywords
VR, Unity, BIM, HTTP, BIM Track, project bank

Sisältö

Lyhenteet ja termit	6
1 Johdanto	16
2 Aloitus	16
3 VR-laitteisto	17
3.1 HTC Vive -Järjestelmä	17
3.2 Steam VR 1.0 Majakka (lighthouse)	19
3.3 HTC Vive -virtuaalitodellisuuslasit	20
3.4 HTC Vive -ohjain	22
4 Autodesk Revit -mallin vienti Unityyn	24
4.1 Ohjelmat	24
4.2 Vientiprosessi	24
4.3 Ongelmat	26
5 Mallien teko ja muokkaus	27
5.1 Ohjelmat	28
5.2 Muokkaus	28
5.2.1 3D-mallien tekoa	28
5.2.2 Päävalikon grafiikkaa	30
5.3 Koulumallin muokkaus Unityssa	31
6 Optimointi	40
6.1 Optimoinnin tarve	40
6.2 Käytetyt optimointikeinot	41
7 Ohjelman Rakenne	45
7.1 Unity Scene	45
7.2 Päävalikko	45
7.3 Koulumalli	46
8 VR-toiminnallisuuden toteuttaminen Unityssa	47
8.1 VR:n käyttöönotto	47
8.2 Teleporttaus järjestelmät	48
8.3 Valikkorakenne ja työkalunvaihtaja	50
8.4 Pelimaailman valikoiden toteuttaminen VRTK:n avulla	51
9 Vuorovaikutus Unityssa	54
9.1 Käsityökalu eli oletustyökalu	54
9.2 Kehitystehtävän lisästyökalu	56
9.3 Kehitystehtävän historiatyökalu	58

9.4	Piirtotyökalu	59
9.5	Tavaran lisäystyökalu	60
10	Bimtrack integraatio	61
10.1	Tietoa BIM:stä	61
10.2	BIM Track	61
10.3	Tunnusten luonti BIM Trackissa ja Watsonissa	62
10.4	BIM Track -API	64
10.4.1	Unityn ja API:n välinen kommunikointi	66
10.4.2	GET-metodin käyttö	66
10.4.3	POST-metodin käyttö	67
10.4.4	PATCH-metodin käyttö	67
11	Pohdinta	68
11.1	Vuorovaikutus	68
11.2	Kehitysideoita	69
11.3	Loppusanat	70
	Lähteet	72

Lyhenteet ja termit

3D-grafiikka	Tietokonegrafiikkaa, joka on mallinnettu kolmiulotteisessa tilassa.
3DS Max	Autodesk-yhtiön kehittämä 3D-mallinnusohjelma.
6-DOF	Six Degrees of Freedom, kuinka monella tapaa asia on vapaa liikkumaan 3D-ympäristössä. 6-DOF:ssa asia on vapaa liikkumaan kääntymällä (rotation) sekä siirtymällä (translation) x-, y-, z-akselien mukaan.
Ajotiedosto	Executable file, on ohjelmatiedostomuoto tietokoneohjelman tallentamiseen.
API	Application Programming Interface, ohjelmointirajapinta, jonka avulla ohjelman toimintoja voidaan käyttää.
Baking	Leipominen, Unityssa tarkoitetaan prosessia, jossa lasketaan tietoa parametrien mukaan ja tallennetaan tiedot nopeaa lukua varten.

Batch	Piirtokäskey, jossa vaihdetaan renderöintitilaa. Renderöintitilan vaihto on raskasta.
BIM	Building Information Model, rakennuksen tuotetietomalli. Rakennuksen elinkaaren ja siihen liittyvien tietojen esitys sähköisessä muodossa.
BIM Track	Kanadalaisen BIM ONE -yrityksen WWW-pohjainen projektipankkipalvelu vuorovaikutukseen BIM-mallin kanssa.
C#	Microsoftin kehittämä C-pohjainen olio-ohjelmointikieli.
Collider	Määritelty alue, joka voi toimia estäjänä, jolloin asia ei läpäise collideria tai laukaisijana (trigger), jolloin jotain tapahtuu, kun ollaan colliderin kanssa myötävaikutuksessa.
Displayport	Näyttöliitäntästandardi äänen-, kuvan- ja tiedonsiirtoon. Kirjoitetaan yleensä yhdessä versionumeron kanssa tyyliin: DisplayPort 1.4, tämänhetkinen uusin versio on 1.4.
DrawCall	Piirtokäskey, CPU antaa käskyn GPU:lle, että jotain pitää piirtää. Drawcallissa käytetään tämän hetkistä renderöintitilaa. Nykyään hyvin kevyt operaatio.

Dynamic	Tarkoitetaan ei staattisia objekteja. Dynaamisia objekteja voi liikutella
FBX	FilmBoX , autodeskin omistama tiedostotyyppi 3D-ohjelmien väliseen tiedonsiirtoon.
Fotorealismi	Pyrkimys saada valokuvantarkka lopputulos.
Fov	Field of view, näkökenttä asteina.
FPS	Ruudunpäivitystä (Frames per second), kuinka monta kertaa ruutu päivittyy sekunnissa. Ilmoitetaan joko suoraan ruutujen määränä sekunnissa tai yksittäisen ruudun päivitys nopeutetana, jolloin yksikönä ajan määre (ms).
Frostrum Culling	Optimointitekniikka, jossa piirretään vain ne 3D-mallit, jotka ovat kokonaan näkökentän (fov) ja piirtoetäisyyden sisällä.
GameObject	Perusluokka kaikille olioille Unityssa.
GIMP	GNU Image Manipulation Program, avoimeen lähdekoodin perustuva kuvanmuokkausohjelma.

GitHub	Verkkosivusta GIT-versionhallintaa käyttäville ohjelmistoprojekteille.
HDMI	High Definition Multimedia Interface, liitäntä standardi kuvan ja äänen siirtämiseen. HDMI on lisenssimaksullinen liitäntä. Kirjoitetaan yleensä yhdessä versionumeron kanssa tyyliin: HDMI 2.0, tämänhetkinen uusin versio on 2.1.
HTC Vive	Virtuaalitodellisuusjärjestelmä, johon sisältyy lasit, ohjaimet ja Lighthouse-majakat.
HTTP	Hypertext Transfer Protocol, protokolla, jota www-palvelimet ja selaimet käyttävät tiedonsiirtoon.
Hz	Lyhenne Hertsistä, taajuuden yksikkö.
IPD	Interpupillary distance, on silmien pupillien keskipisteiden välinen etäisyys millimetreinä.
Kehitystehtävä	Ticket, eräänlainen lipuke, joka kuvaa ongelman, voi myös ehdottaa ratkaisua.

Kärki (Vertex)	3D-grafiikassa pisteitä, jotka ovat särmien molemmissa päissä.
Light Probe	Valopallo, jolle lasketaan etukäteen valoisuus arvoja. Muodostaa muiden valopallojen kanssa verkon, joka määrittelee objektien valoisuuden sen perusteella missä kohdin verkkoa objekti on. Toimii sekä staattisilla, että dynaamisilla objekteilla.
Lighthouse	"majakka" on laite, joka mahdollistaa asian seuraamisen 6-DOF-tasolla, hyvän seuraamisen saamiseksi tarvitaan 2 majakkaa vastakkaisille puolille.
Lightmap	Valotekstuuri, johon etukäteen lasketaan miten valon tulisi käyttäytyä säästään huomattavasti ajon aikaisia resursseja. Voidaan käyttää vain staattisten objektien kanssa.
Liikkumiskeino	Locomotion, Järjestelmä, joka määrittelee liikkumiskeinon liikkumiseen 3D-maailmassa.
LoD	Level of Detail, optimointitekniikka, jossa mallista renderöidään heikompileatuinen versio mitä kauempana mallia havaitsija on.

Materiaali	Unityssa tarkoittaa yhtä tai useampaa tekstuuria, jolle on määritelty omat piirtoon ja näyttöön liittyvät asetukset.
Maya	Myös Autodeskin omistama 3D-mallinnusohjelma käyttää samaa 3d-moottoria 3DS Maxin kanssa. Alun perin saatu yritysostolla.
Mesh (3D)	Polygoniverkko, kappale, joka koostuu useista polygoneista. Muodostaa 3D grafiikan ruudulle.
Multipass	Unityn VR renderöintitekniikka, Multipassissa renderöitävä kuva käydään läpi kaksi kertaa, kerran kummallekin näytölle.
Näyttönohjain (GPU)	Graphics Processing Unit, tietokoneen osa, joka kiihdyttää näyttökuvien laskentaa.
Occlusion Culling	Optimointitekniikka, jossa tarkastellaan, näkeekö kamera kohteen ja piirretään vain, jos näkee.
OpenVR	Valven kehittämä avoin API, joka mahdollistaa VR-lasien käytön, tukee yleisimpiä vr-laseja.
Paint	Monien Windows-versioiden mukana tuleva helpokäyttöinen perustason kuvan muokkaus ohjelma.

Pelimoottori	Ohjelmistokehitysalusta, joka on suunniteltu pelien tekemiseen.
Piirtoetäisyys	Kuinka kauas 3D-grafiikka piirretään. Unityssa jaoteltu nearplaneen ja farplaneen, joista near kertoo piirtoetäisyyden lähtöpisteen ja far loppupisteen.
Piirtäminen	Piirtämisellä tarkoitetaan operaatiota, jossa GPU piirtää objekteja ruudulle.
Polygoni	Monikulmio, on tasokuvio, jonka reuna on suljettu murtoviiva. Kolmio on yksinkertaisin polygoni.
Projektipankki	Webpalvelu vuorovaikutukseen BIM-mallien kanssa.
Proessori (CPU)	Central Processing Unit, eli suoritin on tietokoneen osa, joka suorittaa tietokoneohjelmien konekielisiä käskyjä.
Renderöinti	Kuvan luomista 3D- tai 2D-modellista.
Revit	Autodeskin kehittämä Bim mallien luonti- ja muokaus ohjelma.

Scene	Unityssa scene tarkoittaa suoritettavaa kokonaisuutta, johon kuuluu kaikki, mitä tarvitaan Peli-tason ajamiseen.
Singlepass	Unityn VR renderöintitekniikka, Singlepassissa renderöitä kuva käydään läpi kerran mutta kaksinkertaisella leveydellä, säästään huomattavasti CPU resursseja ja hieman GPU:ta.
Skripti	Unityssa skripti on ohjelmitava komponentti, jolla voidaan käyttää Unityn api:a tai vapaasti luotavia lisäosia. Unityn skriptit ohjelmoidaan C#-ohjelmointikielellä.
Static	Tarkoitetaan unityssa staattisia objekteja. Staattisia objekteja ei voi liikuttaa.
Steam	Valven luoma maailman suosituin digitaalisten pelien-, videoiden- ja ohjelmienjakoalusta.
SteamVR	Valven oma paketointi OpenVR:aan, joka tuo lisää ominaisuuksia, kuten virtuaaliset rajat.
Särmä (Edge)	3D-grafiikassa janoja, jotka rajaavat tahkon ja joiden loppupisteet ovat kärkiä.
Tahko (Face)	3D-grafiikassa alue, jota särmät ympäröivät.

Tekstuuri	Pintakuva, kuva, jolla lisätään 3D- tai 2D-mallien laatua.
Teleportti	Välitön siirtyminen kahden paikan välillä.
Triangle	Kolmio, yksi 3D-grafiikan perusteista. Suurin osa 3D-grafiikasta muodostuu kolmioista.
UI	Käyttöliittymä, tapa, jolla käyttäjä voi olla vuorovai- kutuksessa ohjelmiston kanssa.
Unity	Tanskasta lähtöisin oleva pelimoottori, joka tukee useita eri alustoja.
Unity Asset Store	Unity kauppapaikka, jossa voidaan jakaa ilmaiseksi tai myydä asioita Unityyn.
USB	Universal Serial Bus, liitännästandardi, joka määrit- telee kaapelit, protokollat ja liittimet tiedon- ja ener- giansiirtoa varten.
Valve	Yhdysvaltalainen yritys, joka toimii videopelien - kehittäjänä, -julkaisijana ja -digitaalisenajakelijana.

VR	Virtual Reality, virtuaalitodellisuus. Virtuaalitodellisuudessa jäljitellään käyttäjän ja oikean elämän vuorovaikutusta.
VRTK	Virtual Reality Toolkit, kokoelma hyödyllisiä skriptejä ja esimerkki toteutuksia vr-ohjelmien tekoon.
XML	Extensible Markup Language, tekstimuotoinen merkintäkieli. Yleensä UTF-8-koodattua.

1 Johdanto

Opinnäytetyön aiheena on rakennuksen tietomalliin liittyvä vuorovaikutus virtuaaliodellisuusjärjestelmässä (VR). Työssä käydään läpi prosessi, jossa Revit -tyyppisen arkkitehtuurimallin pohjalta tehdään Unity-sovellus, joka sisältää 3D-mallin ympäristöineen, VR-toiminnallisuuksia ja tuen BIM Track -projektipankki palvelun käyttöön suoraan VR:ssä.

Opinnäytetyömme kantavana ajatuksena oli tehdä ohjelma, joka esittelisi mahdollisuuksia VR:n käyttöön rakennusteknisen mallin liittyvässä työkulussa. Se näyttäisi miten VR voisi tukea perinteistä rakennusten suunnittelua ja olisi visualisointi apu sekä ammattilaiselle että arkkitehtuuripiirroksista ymmärtämättömille maalikoille. Siinä voisi myös suoraan viestiä projektipankki-palvelun kanssa ja luoda korjaus ehdotuksia.

Opinnäytetyö tehtiin yhteistyössä Karelia AMK:n kanssa. Aloitimme opinnäytetyön teon toukokuussa 2018. Vain hetki ennen opinnäytetyön aloitusta oppilaitoksemme oli hankkinut HTC Vive -virtuaaliodellisuuslaseja ja työasemia niiden kanssa työskentelemiseen. Näistä saimme käyttööme opinnäytetyön teon ajaksi järjestelmän, jolla koko opinnäytetyö toteutettaisiin.

2 Aloitus

Aloitimme opinnäytetyön valitsemalla pelimoottorin, joka olisi perustus koko VR-ohjelmallemme. Valitsimme ilman sen suurempia valinta/testi -prosessia Unity-pelimoottorin käytettäväksi alustaksi. Valintaan vaikutti positiivisesti Unityn tutuus, sillä meistä toisella (Mika) oli ollut kursseja, joissa Unityä oli käytetty, lisäksi nopealla hakukoneiden käytöllä löytyi paljon alottelijaystävällisiä ohjeita VR:n käyttöön Unitylla. Unity versio, jota käytimme, oli 2017.4.2f2.

Opinnäytetyön toteutusta varten saimme ARKPII-arkkitehtitoimistolta Revit-muodossa olevan arkkitehtuurimallin. Tämä malli piti muuttaa kuitenkin Unityn kanssa yhteensopivaksi. Avasimme mallin 3DS Maxissa ja konvertoimme Unityn ymmärtämään muotoon. Tähän konvertoituun malliin ei kuitenkaan tullut mukaan kaikki tekstuurit, joten meidän piti asettaa ne Unityssä paikoilleen. VR-sovelluskehitystä varten haimme Unityn Asset Storesta SteamVR- ja VRTK-työkalut.

Aloitimme VR-sovelluskehityksen harjoittelun RayWenderlich-sivustolta [1, HTC Vive Tutorial for Unity] löydetyn tutoriaalimateriaalin mukaan. Tutoriaalissa käyttäjä oppii saamaan pelimaailman näkymään VR laseissa, teleporttaamaan eri puolille kenttää sekä tarttumaan valmiiksi määriteltyihin esineisiin. Kun olimme sisäistäneet perusasiat, aloitimme uuden Unity-projektin, jota kehittäisimme eteenpäin valmiiksi opinnäytetyöksi asti.

3 VR-laitteisto

3.1 HTC Vive -Järjestelmä

Kehitimme ohjelman HTC:n Vive -laitteiston (Kuva 1) avulla. Vive koostuu kahdesta Lighthouse "majakasta", VR-laseista ja kahdesta ohjaimesta. Majakat asetetaan vastakkain toisiaan nähden ja luovat käyttäjällä alueen, jossa VR-laseja ja ohjaimia seurataan 6-DOF tasolla. Ohjaimet ovat samanlaiset kumpaankin käteen ja toimivat sekä vasen että oikeakätisille. SteamVR API yhdessä majakoiden kanssa osaa ohjelmallisesti arvioida kummassa kädessä ohjainta tällä hetkellä pidetään.



Kuva 1. HTC Vive -Virtuaalitodellisuusjärjestelmä vasemmalta oikealle ohjain, majakat ja VR-lasit.



Kuva 2. Viven ja PC:n välinen linkkilaatikko.

Vive vaatii kytkennän tietokoneeseen toimiakseen, tämä tapahtuu erillisen linkkilaatikon (Kuva 2) avulla. Linkkilaatikkoon liitetään tietokoneelta näyttökaapeli, joko HDMI- tai DisplayPort tyyppisenä ja USB-kaapeli. Lisäksi linkkilaatikkoon yhdistetään verkkovirtaan yhdistetyn muuntajan virtajohto. Linkkilaatikosta ulospäin lähtee kaapelit Vive VR -laseille. Vivessä oleva kaapeli on yhdistelmäkaapeli, joka sisältää HDMI-, USB- ja virtakaapelit.

On suositeltavaa, että tietokone, jolla VR-laitteita käytetään ylittää vähintäänkin viralliset minimi järjestelmävaatimukset (Taulukko 1). Koulun Virtuaalitodellisuus työasema, jota käytimme koko opinnäytetyön tekemiseen, on kaikilla osa-alueilla selvästi virallisia HTC Viven järjestelmäsuositusten yläpuolella.

Taulukko 1. Suositus- ja minimivaatimukset sekä Koulun Vive-työasema. [2, What are system requirements].

Komponentti	Suosittelut järjestelmävaatimukset	Vähimmäis järjestelmävaatimukset	Koulun Vive-työasema
Proessori (CPU)	Intel Core i5-4590/AMD FX 8350 vastaava tai parempi	Intel Core i5-4590/AMD FX 8350 samantasoinen tai parempi	AMD Ryzen 1600X
Näytönohjain (GPU)	NVIDIA GeForce GTX 1060, AMD Radeon RX 480 vastaava tai parempi	NVIDIA GeForce GTX 970, AMD Radeon R9 290 vastaava tai parempi	NVIDIA 1080 GTX
Muisti	4 GB RAM tai enemmän	4 GB RAM tai enemmän	16 GB
Näyttöliitännät	HDMI 1.4, DisplayPort 1.2 tai uudempi	HDMI 1.4, DisplayPort 1.2 tai uudempi	HDMI 2.0, Displayport1.4, DVI
USB portti	1x USB 2.0 tai uudempi	1x USB 2.0 tai uudempi	4x USB 3.1 Gen1 2x USB 2
Käyttöjärjestelmä	Windows 7 SP1, Windows 8.1 tai myöhempi, Windows 10	Windows 7 SP1, Windows 8.1 tai myöhempi, Windows 10	Windows 10

3.2 Steam VR 1.0 Majakka (lighthouse)

SteamVR lighthouse on Valven kehittämä seuraustekniikka. Sillä voidaan luoda kahden majakan (Kuva 3) avulla (Kuva 4) tila, jossa laitteita voidaan seurata erittäin tarkasti 6-DOF tasolla. Majakoiden suositeltu sijoitus on maksimissaan viiden metrin päässä toisistaan nurkasta nurkkaan, tällöin seuranta alueen kooksi tulisi 3,54mx3,54m eli 12,50 neliometriä. Omissa testeissämme majakat ovat toimineet moitteettomasti jopa seitsemän metrin päässä toisistaan.



Kuva 3. Päällä oleva lighthouse majakka.



Kuva 4. Vivellä luotu pelialue.

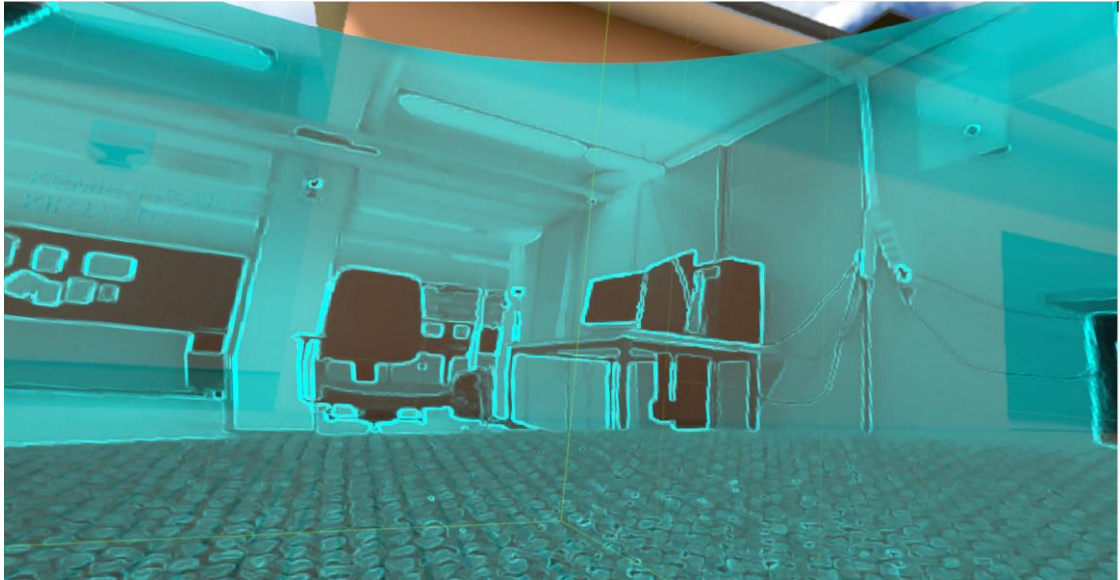
3.3 HTC Vive -virtuaalitodellisuuslasit

HTC Vive -virtuaalitodellisuuslaseissa (Kuva 5) on kaksi 90:n hertzin taajuudella toimivaa resoluution 1080x1200 näyttöä, yksi kummallekin silmälle. Näyttöjen edessä on linssit, jotka mahdollistavat näyttöjen tarkan näkemisen.

Ne ikään kuin saavat näytöt näyttämään olevansa kauempana kuin oikeasti ovat. Linssien etäisyyttä toisistaan voi säätää lasien oikealla sivulla olevasta pyöriteltävästä säätimestä. HTC Viven IPD:n säätöväli on 60,3-74,1. Lasien vasemmalta löytyy LED-valo, joka kertoo lasien tämän hetkisestä tilasta ja painike, joka avaa Steamvalikon. Laseissa on myös tarranauhat sivuilla ja ylhäällä, joita voi kiristää ja löysentää käyttäjän päähän paremmin sopivaksi. Lasien kuopat sisältävät sensoreita, jotka yhdessä majakoiden kanssa mahdollistavat Viven tarkan seurannan 6-DOF tasolla. Vivessä etupuolella keskellä alhaalta löytyy kamera. Kameraa hyödynnetään ainoastaan kuvan läpivientiin lasien näytölle. Sen avulla voi nopeasti katsoa onko esteitä tai (Kuva 6) vaikka kotieläimiä lähistöllä.



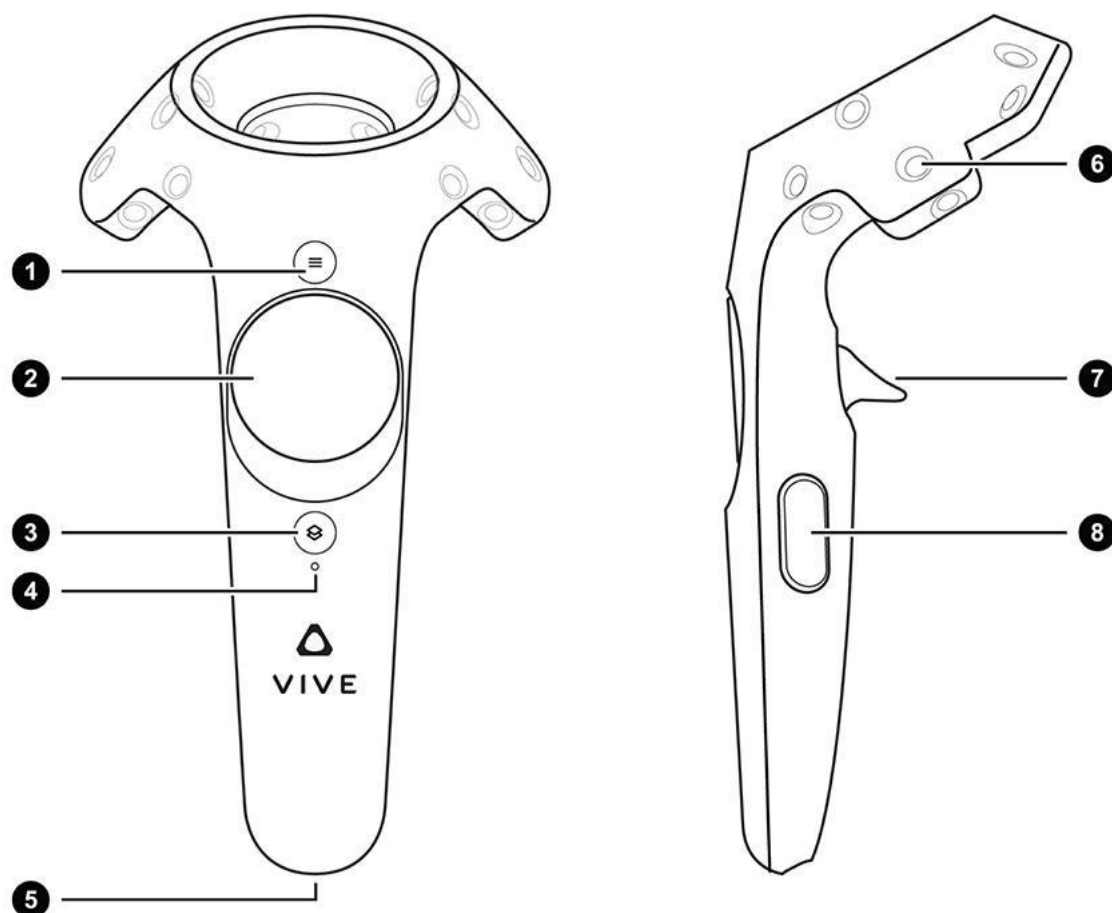
Kuva 5. Vive-virtuaaliodellisuuslasit, kuvassa vasemmalla on linssien säätörulla ja keskellä kamera.



Kuva 6. Kuvaa Viven kameran kautta.

3.4 HTC Vive -ohjain

Vive ohjain on pitkulainen kapula, jota pidetään kädessä kaukosäädinmäisesti. Ohjain toimii langattomasti, ja saa virtansa akusta.



Kuva 7. Ohjaimen eri toiminnallisuudet numeroituna.

Ohjaimen toiminnot kuvassa (Kuva 7) ensimmäinen kohta viittaa valikko nappiin (menu button). Kakkoskohdan nappia kutsutaan ohjauslevyksi (trackpad). Kosketuslevy on kaksiosainen, itse kosketuslevy joka on hyvin samantyyppinen kuin kannettavissa tietokoneissa oleva ja painonappi. Kosketuslevy tunnistaa koko alalta, jos sormi on levyn päällä ja se voidaan ohjelmoida tekemään haluttua asiaa riippuen sormen sijainnista. Kun painaa levyn pohjaan, painautuu nappi pohjaan ja tällöin ohjelmoidut funktiot voivat olla täysin erilaisia. Kolmas kohta on järjestelmäpainike (system button). Järjestelmä painikkeen toiminta on ennalta määritelty ja sitä ei voi ohjelmoida itse. Pitkä järjestelmänapin painallus joko sammuttaa tai käynnistää ohjaimen sen tilasta riippuen. Kun ohjaimet ovat päällä ja painaa hetkellisesti nappia, avautuu SteamVR-valikko. Neljäs kohta on LED-valo, joka antaa tietoa ohjaimien tämänhetkisistä tiloista. Viides kohta on micro USB -liitin USB-johtoa varten. Tätä kautta tapahtuu ohjainten akun lataaminen ja ohjelmiston päivittäminen. Kuudes kohta on SteamVR 1.0 sensori, jo-

ka majakan avulla havaitsee ohjaimen sijainnin. Jokainen pieni kuoppa ohjaimessa on yksi sensori. Seitsemäs kohta on liipaisin. Liipaisin on kaksiosainen, jossa ensisijainen painallus on analoginen osa, jolle voidaan ohjelmoida lukuisia eri tiloja riippuen, kuinka paljon liipaisinta on painettu pohjaan. Toinen osa on perinteinen painonappi, joka aktivoituu, kun liipaisin painetaan täysin pohjaan. Kahdeksas kohta on otenäppäin (grip), joka löytyy kummaltakin puolelta ohjainta. Riippumatta kummalta puolelta nappia painetaan, sama toiminto tapahtuu.

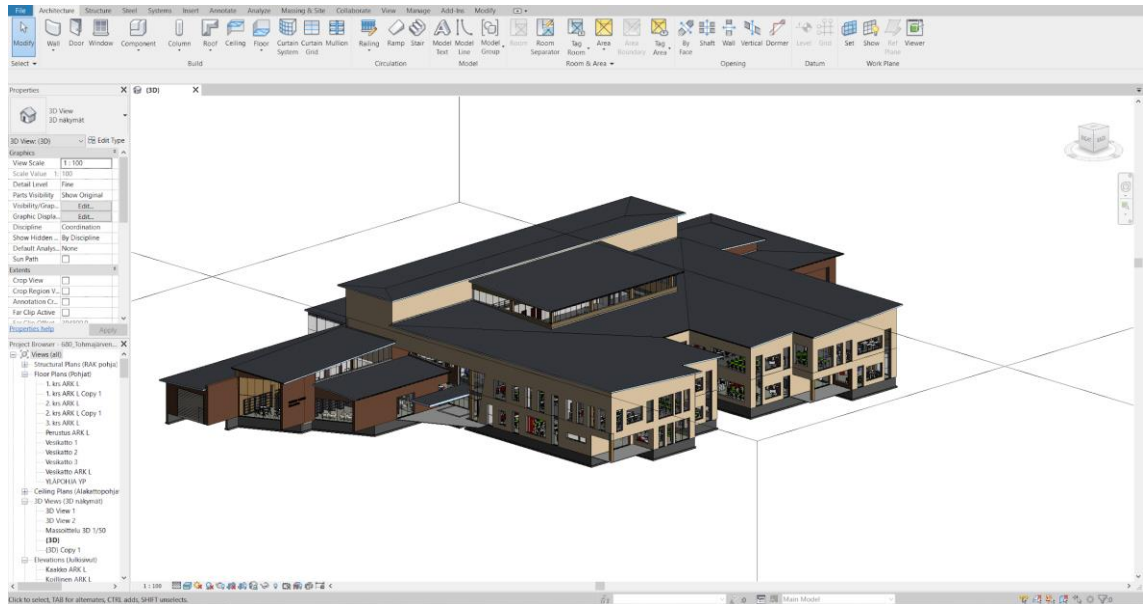
4 Autodesk Revit -mallin vienti Unityyn

4.1 Ohjelmat

Revit-mallin käsittelyssä käytettiin kahta eri ohjelmaa; Autodesk Revit ja 3DS Max. Kumpikin ohjelma ovat Autodeskin kehittämiä. Kumpikin ohjelma on normaalisti maksullinen, mutta saatavilla opiskelijoille kolmen vuoden ilmaisella lisenssillä.

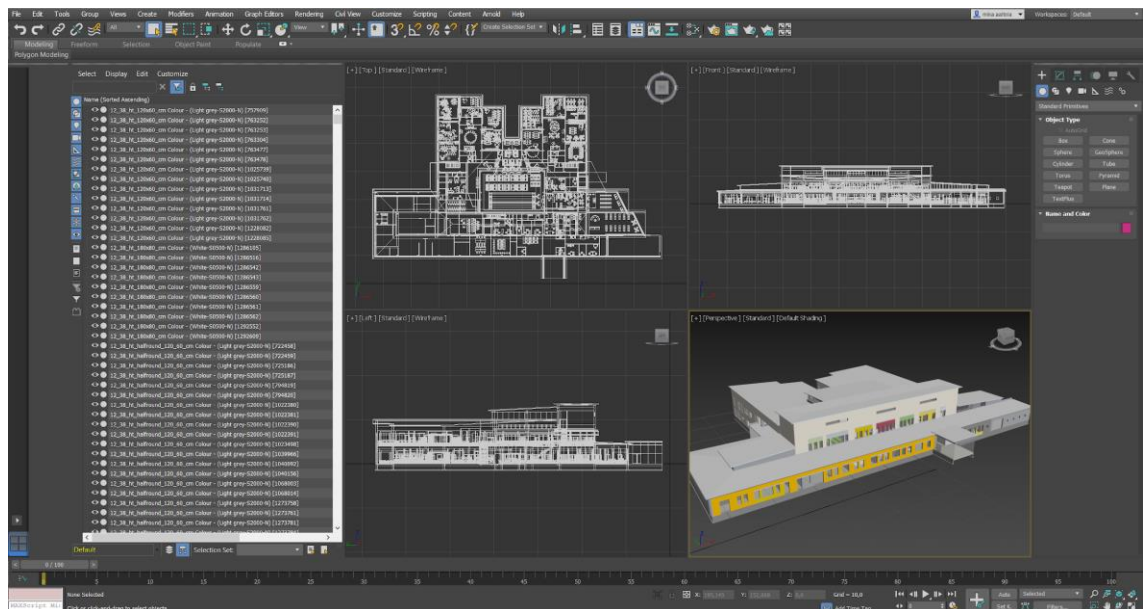
4.2 Vientiprosessi

Meillä oli käytettävissä arkkitehtitoimisto ARKPII:n RVT-tiedostomuotoinen malli Tohmajärvelle tulevasta Kemien koulusta. Mallin avasimme ensimmäisenä Revit ohjelmassa (Kuva 8), jossa analysoimme mallia ja katsoimme, että siitä löytyy 3D-view kamera. Sen jälkeen konvertoimme mallin FBX-muotoon, joka on yleisesti käytössä oleva siirtotiedostotyyppi eri 3D-ohjelmien väliseen tiedonsiirtoon ja yhteensopiva 3D-mallinnusohjelmien ja Unityn kanssa.



Kuva 8. Koulumalli Autodesk Revit -ohjelmassa.

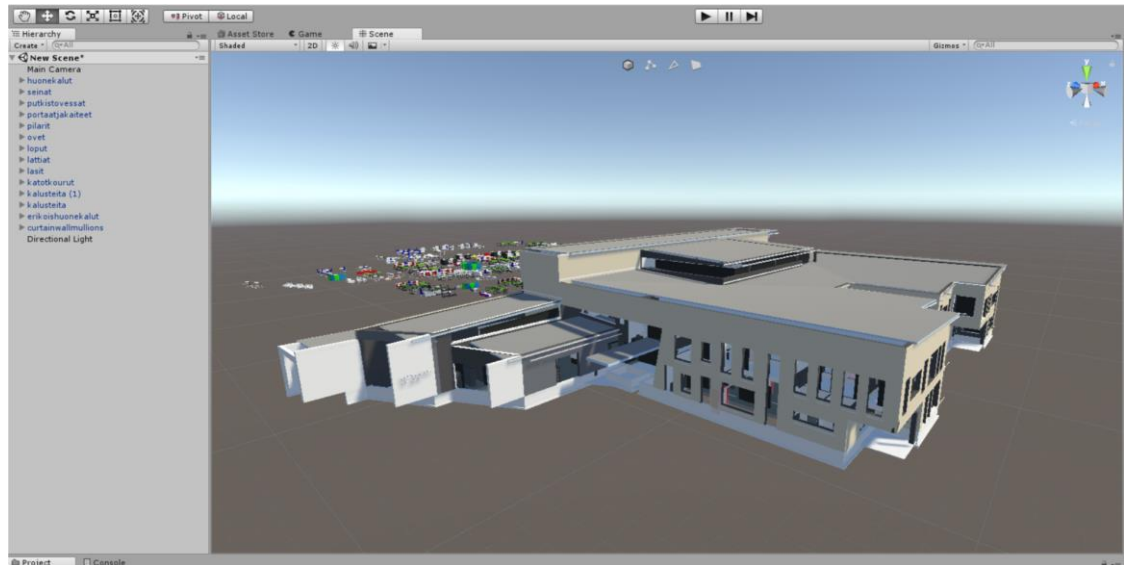
FBX muotoisena avasimme tiedoston 3DS Max:ssa (Kuva 9). Mallissa oli noin 3400 osaa, joka on aika massiivinen määrä, joten helpompaa vientiä ja muokkausta varten jaoinme mallin 11:ta osaan osien tyyppin perusteella.



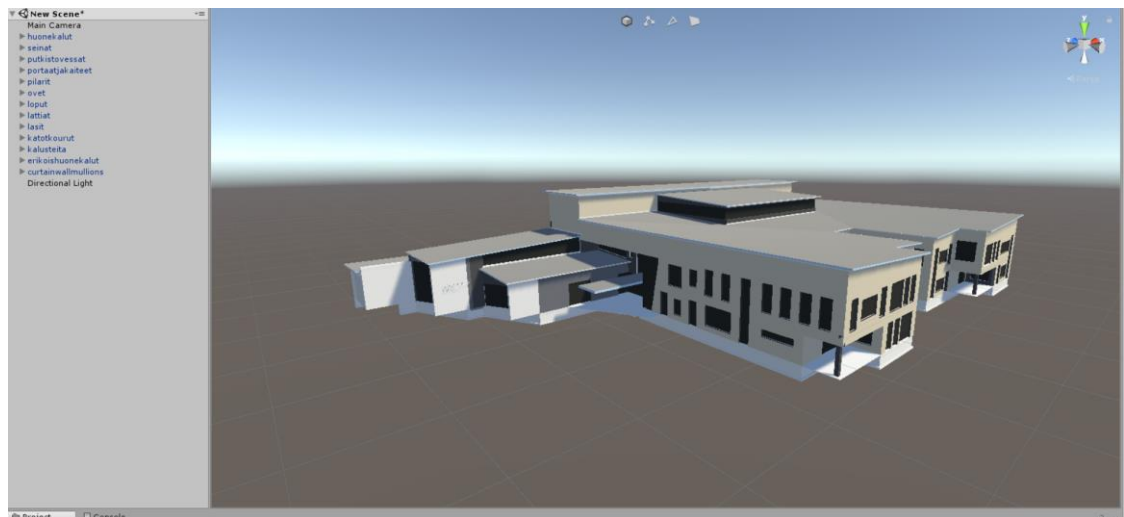
Kuva 9. Koulumalli 3DS Max:ssa.

Unityyn toimme 11 osaa yksitellen käyttäen Unityn omaa tuonti (import) toimintoa samalla katsoen, miten hyvin jokainen joukon tuonti onnistui. Osat olivat aluksi väärissä kohdissa (Kuva 10) toisiinsa nähden. Tämä ongelma saatiin rat-

kaistua laittamalla kaikki osajoukot manuaalisesti samaan koordinaattipisteeseen, jolloin havaitsimme, että joukoista oli muodostunut kokonainen 3D-malli (Kuva 11), jossa osat olivat oikeissa paikoissa.



Kuva 10. Koulumalli Unityssa suoraan 3DS Maxista.



Kuva 11. Koulumalli Unityssa korjatuilla sijoituksilla.

4.3 Ongelmat

Ongelmia tuli vientiprosessin aikana. Ensimmäinen ongelma oli, että materiaalit häviävät, jos yrittää suoraan viedä mallin Revitistä Unityyn. Ongelmaan ratkaisua etsiessämme vastaan tuli maksullisia lisäosia, jotka mainospuheiden mu-

kaan olisivat toimineet. Testasimme yhtä näistä maksullisista lisäosista nimeltä Export to Unity. Export to Unity on suoraan Revit:iin asennettava lisäosa, joka lupaa yhdellä klikkauksella viedä mallin Revitistä Unityyn kokonaisena materiaaleineen. Testeissämme lopputulos ei ollut aivan näin ruusuinen, vaan käytettäessä iso osa materiaaleista siirtyi Revitistä Unityyn ilman ongelmaa, mutta ei kaikki.

Päätimme kuitenkin olla käyttämättä lisäosaa ja käyttää 3DS Max:a, tällöin suurin osa materiaaleista siirtyi ongelmitta ja 3DS Max:ssa pystyimme jaottelemaan mallin eri osat joukkoihin, joka helpotti muokkausta Unityn puolella. Unityssa lisätyt loput materiaalit teimme joko itse tekijänoikeus vapaista kuvista [3, Terms and Conditions] GIMP:a käyttämällä, tai latasimme valmiiksi tehdyn materiaalin ilmaisia materiaaleja jakavalta sivulta [4, f.a.q.].

Toinen ongelma oli huomattavasti vähäpätöisempi, joissain viedyissä joukoissa kierto oli väärin. Kierron korjaus oli yksinkertaista tehdä Unityssa. Ei tarvitse kuin valita vain kaikki objektit, joissa kierto on väärin ja kääntää ne oikein päin. Kaikki kierto ongelmat ratkesivat, kun x-akselin kierrot laittoi -90 asteeseen. Tämän korjauksen olisi voinut myös tehdä 3DS Max:ssa.

Kolmannen ongelman havaitsimme, kun kaikki 3D-malli joukot oli lisätty Unityn Sceneen. Havaitsimme, että osa koulumallin osista menee päällekkäin toistensa kanssa ja niiden ulkopinnat ovat täsmälleen samassa kohdin. Tästä aiheutuu, että kummatkin osat vilkkuvat rumasti. Sen lisäksi, että vilkkuminen näytti huonolta ulkonäöllisesti, se aiheutti pahaa oloa ohjelmaa käytettäessä VR-laseilla. Vilkkumiset saatiin korjattua, kun toista osaa siirrettiin hyvin pieni määrä koordinaattiakselilla, jolloin osien ulkopinnat eivät olleet enää täsmälleen samoissa kohdissa ja Unityllä ei ollut enää ongelmia päättää kumpaa pintaa näyttää.

5 Mallien teko ja muokkaus

5.1 Ohjelmat

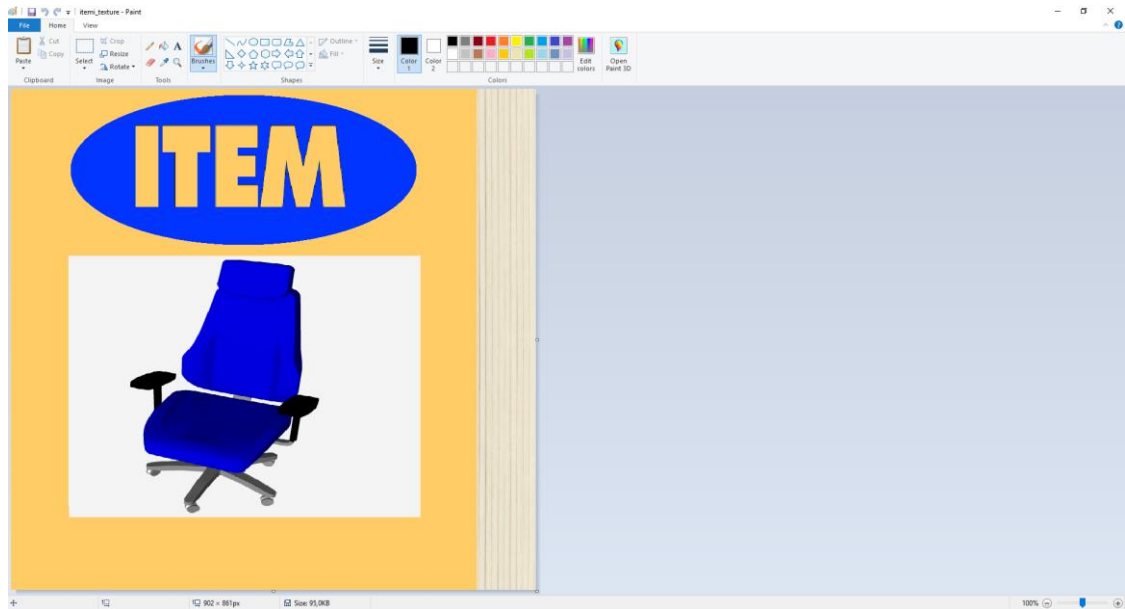
Mallien luomiseen ja muokkaamiseen käytettiin Autodesk Mayaa. Mayan saa haltuun opiskelijalisenssillä kolmeksi vuodeksi ja sopii siten tähän tarkoitukseen hyvin. Mayan käyttämiseen vaikutti se, että toisella meistä (Joonas) oli ennestään kokemusta Mayan käyttämisestä. Muokkaukset olisi voinut suorittaa myös muissa 3D-muokkaus ohjelmissa, kuten 3DS Max.

Mallien tekstuureja varten käytettiin Paintia yksinkertaiseen kuvanmuokkaukseen ja GIMP:ä monimutkaisempaan muokkaamiseen.

5.2 Muokkaus

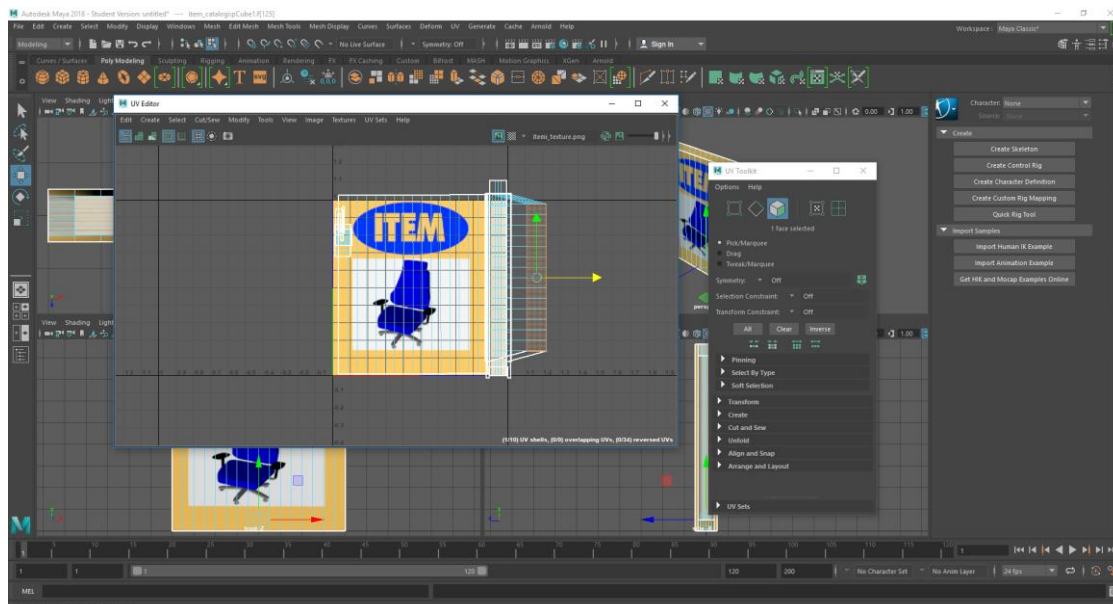
5.2.1 3D-mallien tekoa

Paintin helppokäyttöisyys sopi hyvin yksinkertaisten tekstuurien tekemiseen. Esimerkiksi tavaraluontityökalua esittävä tavarakatalogin tekstuuri oli tehty Paintilla kuvan (Kuva 12) osoittamalla tavalla.



Kuva 12. Tavaranolisäystyökalun grafiikkaa Paint-ohjelmassa.

Grafiikka tavaranolisäystyökaluun (Kuva 12) luotiin ottamalla Unity scenestä kuvankaappaus sinisestä tuolista ja poistamalla kaikki muu paitsi sininen tietokonehuone. Tuolin ympärille piirrettiin paintin työkaluja käyttäen loput katalogin grafiikasta. Soikionmuotoisen sinisen logopallon sai tehtyä soikiotyökalulla. Teksti logoon onnistui tekstityökalulla. Katalogin sivut saatiin tehtyä ottamalla kuva oikeasta kirjasta ja leikkaamalla sivukohta kuvasta katalogikuvaan.

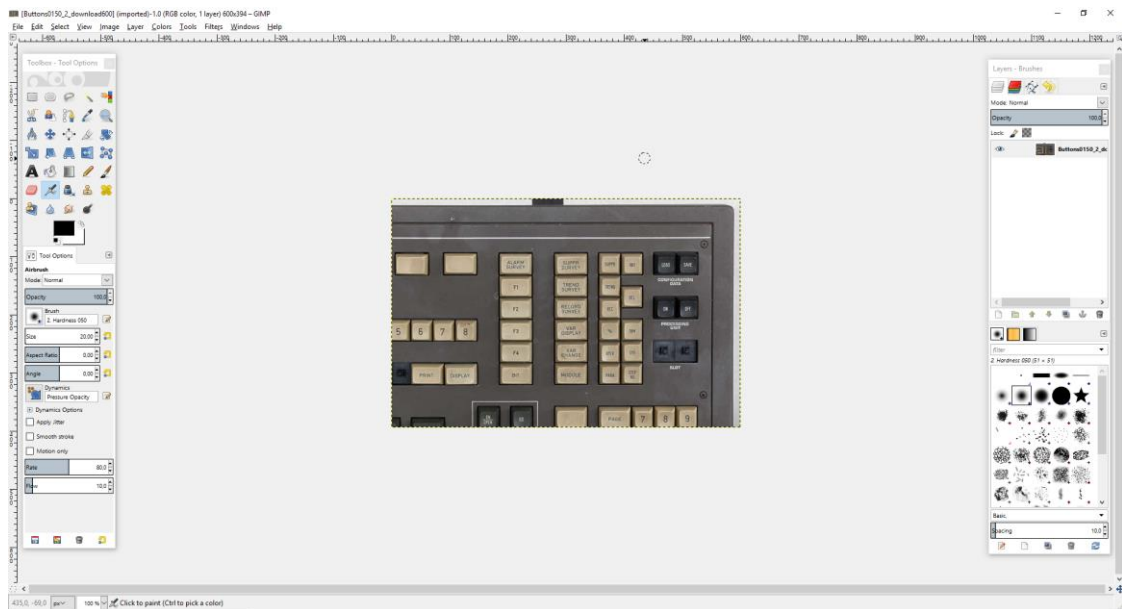


Kuva 13. Tavarallisäys työkaluun tulevaa grafiikka Autodesk Maya ohjelmassa.

Kun grafiikka oli tehty Paintilla, Autodesk Mayalla (Kuva 13) tehtiin 3D malli ensin luomalla yksi polygon cube ja lisäämällä siihen subdivisiooneja. Sitten pinnanvalintatyökalulla valittiin eri pintoja ja pintoja liikutettiin niin, että malli alkoi muodostua tavarakuvaston muotoiseksi. Tämän jälkeen luotiin uusi materiaali, joka pohjautuu Paintilla tehtyyn tavarakatalogiin. Materiaalin tekstuuri oli tosin väärissä kohdissa 3D-mallin pintoja ja tekstuuri piti asettaa paremmin, jotta malli näyttäisi oikealta tavarakuvastolta. 3D-mallin pintoja pystyi liikuttamaan kaksiulotteisesti oikeisiin kohtiin tekstuuria ja tekstuurin kohdat piirtyivät oikeisiin kohtiin (Kuva 13).

5.2.2 Päävalikon grafiikkaa

Otimme kuvan vanhasta näppäimistöistä internetistä Textures-sivustolta [5, Buttons0150], ja muokkasimme GIMP:llä (Kuva 14) siitä näppäimet pois käyttäen kloonaustyökalua.



Kuva 14. Näppäimistö ennen muokkausta GIMP-ohjelmassa.

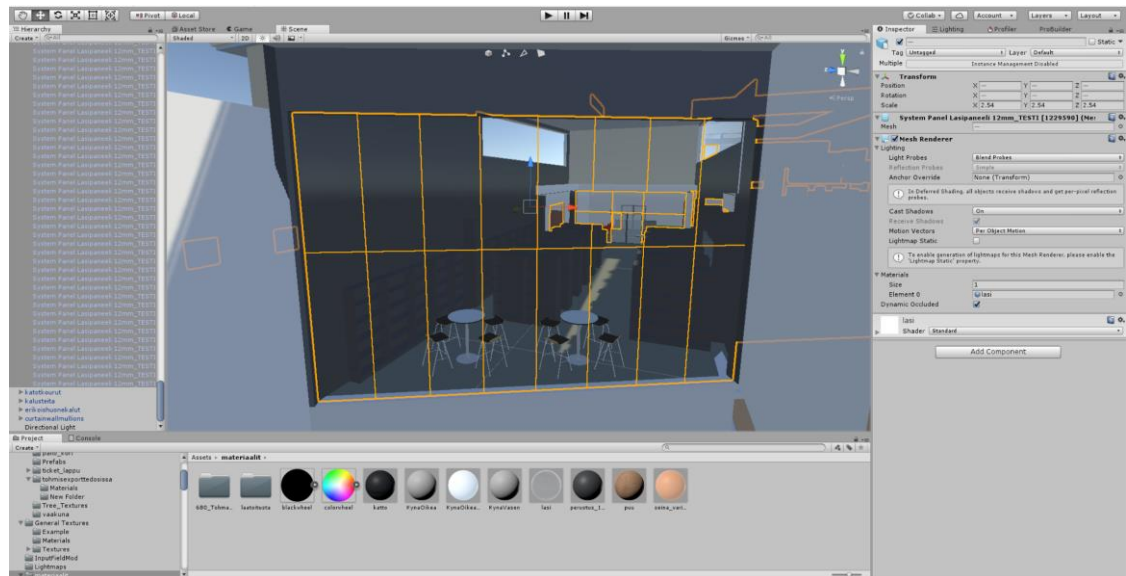
Kloonauustyökälulla siis kopioitiin näppäimistön reunakohdat ja siirrettiin ne näppäinten päälle. Vaikka mitään näppäintä ei enää ollut kuvassa, teimme yhdestä näppäimestä oman kuvansa. Kuvaa käytettiin päävalikossa näppäinkuvakkeena. Näppäimetöntä näppäimistötaustaa käytettiin päävalikon taustakuvana. Tämä sai päävalikon näyttämään näppäimistöltä.

5.3 Koulumallin muokkaus Unityssa

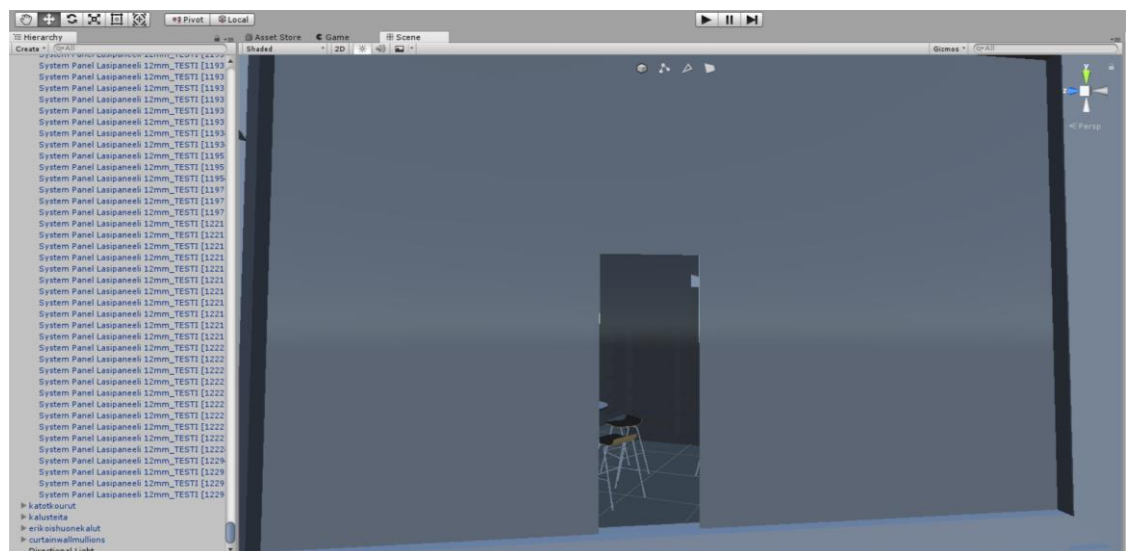
Unityssa malli lisättiin Sceneen ja siihen ruvettiin lisäämään valoja, materiaaleja ja korjattiin 3D-mallin virheitä. Lisäksi rakennettiin tyhjistä hiukan ympäristöä koulun ympärille.

Kun koulun 3D-malli tuotiin Unityyn, siitä puuttui paljon materiaaleja ja esimerkiksi lasit olivat tummanharmaita ja läpinäkymättömiä. Materiaaleja lähdettiin lisäämään käyttämällä Unity-Editoria. Editorissa materiaali joko raahataan Scene-näkymässä halutun 3D-mallin päälle, tai lisätään Inspector-välilehdessä. Materiaalin (Kuva 15) lisäystavoista Inspector-välilehdessä lisääminen on huomattavasti tehokkaampaa, jos tarvitsee lisätä useampaan malliin materiaali, sillä

materiaali lisätään kaikkiin valittuihin 3D-malleihin, kun taas raahaamalla (Kuva 16) saadaan tarkemmin määriteltä mihin osaan materiaali tulee.

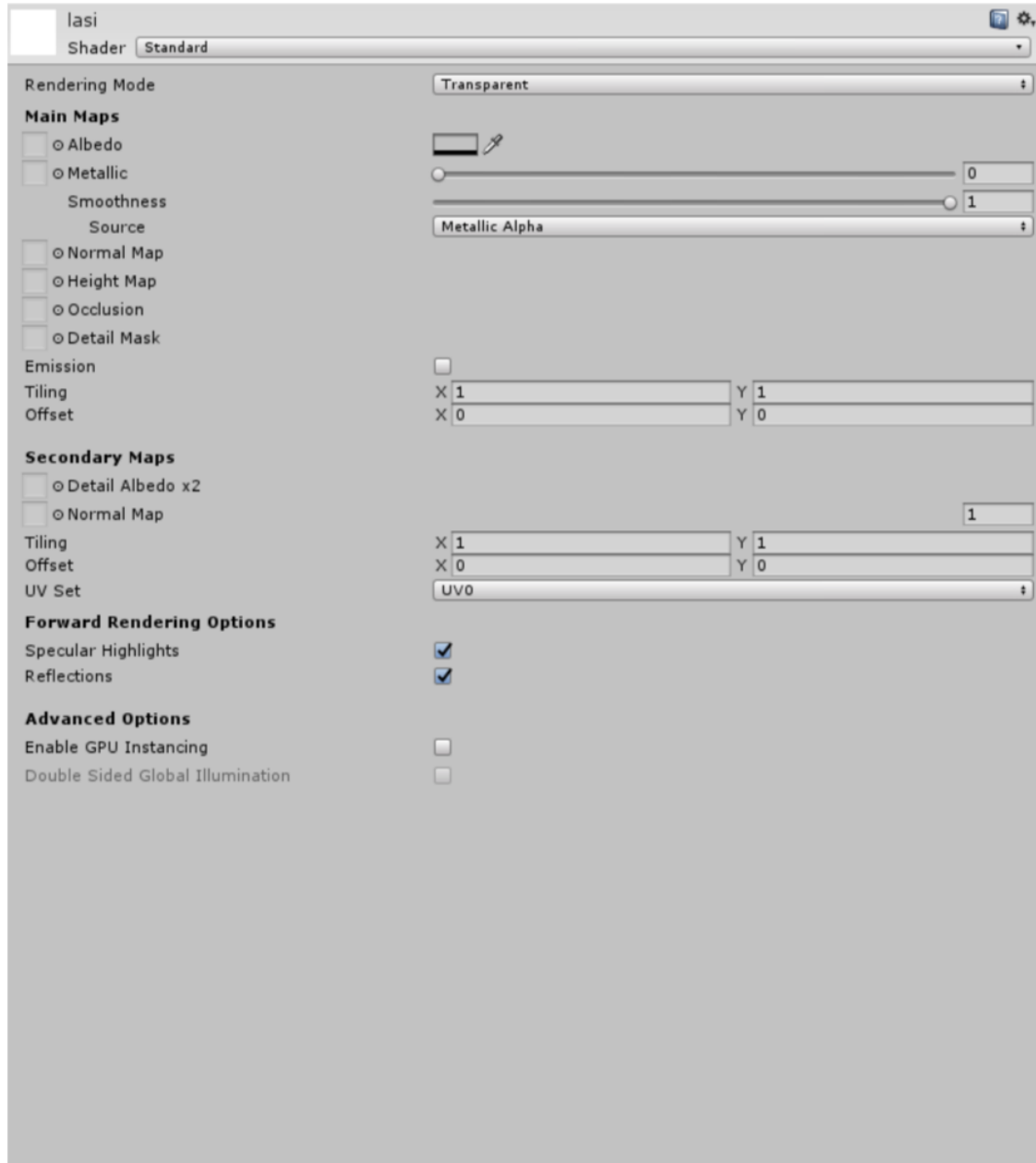


Kuva 15. Lasi-materiaali lisättyä Inspectorin avulla kaikkiin valittuihin 3D-malleihin. Valitut mallit näkyvät keskikuvassa keltaisilla ääri viivoilla ja vasemalla listassa tummennettuina.

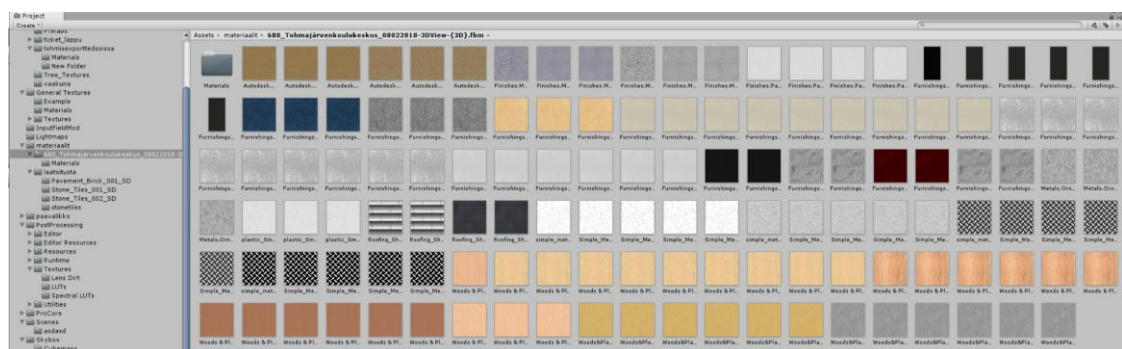


Kuva 16. Lasi materiaali lisättyä raahaa materiaali 3D-malliin tavalla.

Lisätyt materiaalit tehtiin joko itse, käytettiin Autodeskin valmiita tai ladattiin netistä. Itse tehdyt materiaalit tehtiin käyttäen (Kuva 17) Unityn omaa materiaali luontia. Sillä tehtiin esimerkiksi lasi-materiaali ja oranssi seinä.

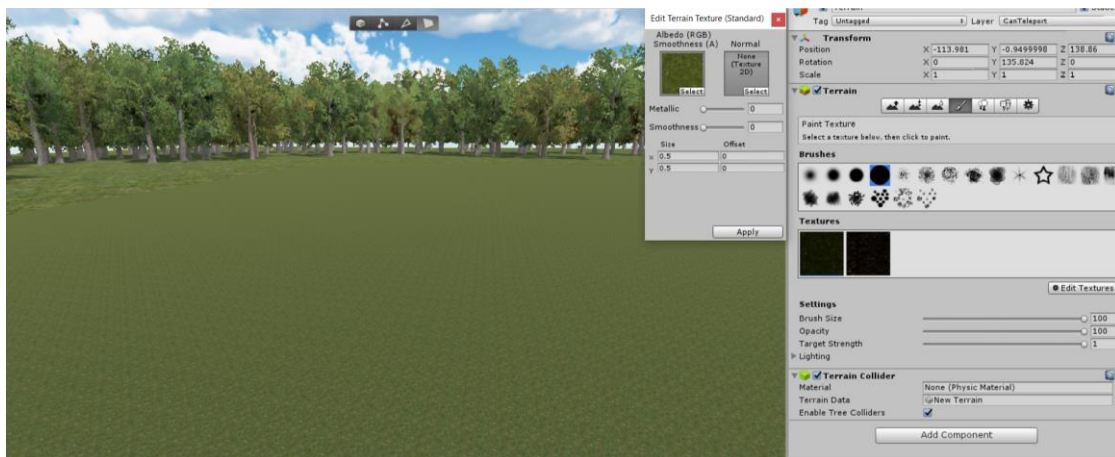


Kuva 17. Kuva Unity Materiaali luonnista, kuvassa lasimateriaali.



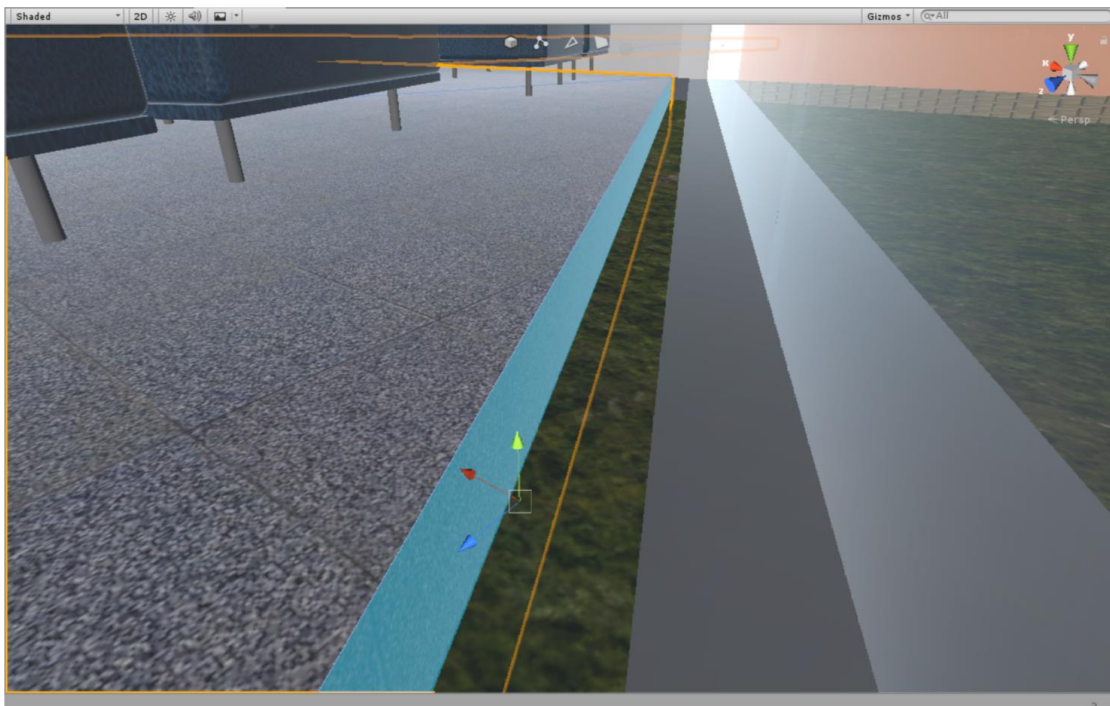
Kuva 18. FBX-mallin mukana tulleet Autodesk tekstuurit.

Vaikka osa Autodeskin materiaaleista ei automaattisesti mennyt 3D-malleihin, sisältyi ne silti FBX-tiedostoon. FBX-tiedosto piti manuaalisesti purkaa, jolloin saimme ison kasan tekstuureja. Tämän jälkeen tekstuureista piti tehdä materiaaleja lisäämällä tekstuuri albedo (Kuva 18) kohtaan Unityn materiaali luonnissa. Internetistä ladattiin 3D TEXTURES-sivustolta [6, Pavement Brick 001] ilmainen tekstuuri koulumallia varten, josta tehtiin materiaali ulkolaatoituksiin.



Kuva 19. Terrain, johon on lisätty Standard Assets -paketista löytyvä ruohotekstuuri ja puita.

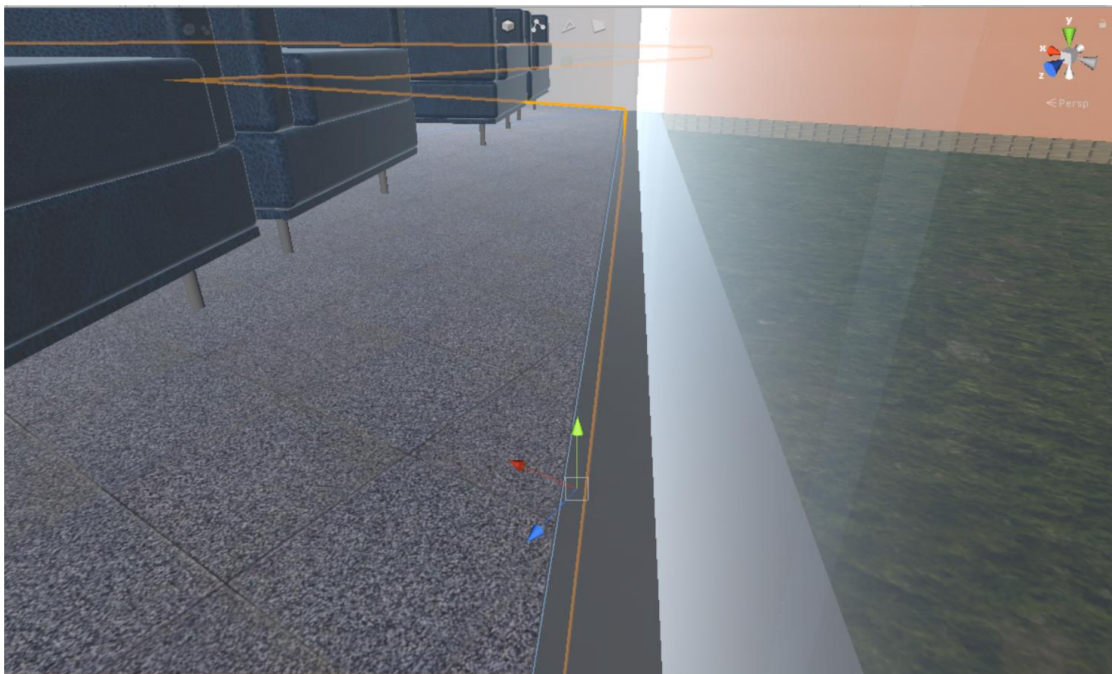
Ympäristöä lähdettiin toteuttamaan luomalla terrain 3D-Objekti. Terrain on Unityn GameObject, joka on tarkoitettu ulkomaailman luontiin. Terrainin muotoa voi vapaasti muokata ja siihen voi helposti lisätä kasvillisuutta ja muita yksityiskoh-
 tia. Terrainiin saatiin ruoho-tekstuuri ja puita lataamalla Unityn Asset Storesta ilmainen Standard Assets -paketti. Standard Assets -paketti on Unity Oy:n kehittämä paketti, joka sisältää lukuisia apuvälineitä helpottamaan Unityllä työskentelyn aloittamista. Terrainiin (Kuva 19) lisättiin ruoho-tekstuuri ja alueen laitamille puita.



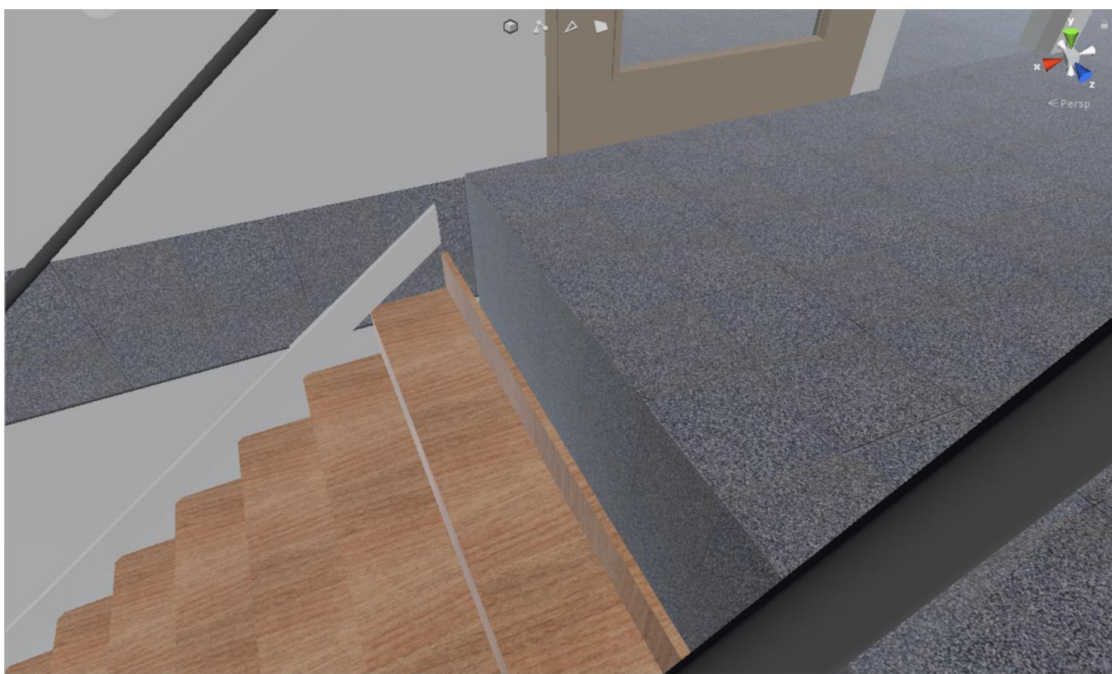
Kuva 20. Vajaaksi jäänyt sivu valittuna ProBuilderin tahkovalinnalla.

Unityssa 3D-mallia tarkastellessa huomasimme, että mallissa oli pieni virheitä. Virheet todennäköisesti tapahtuivat, kun Revit-malli konvertoitiin FBX-muotoiseksi, sillä niitä ei näkynyt, kun Revitillä katsottiin alkuperäistä mallia. Unityssa konvertointi virheiden korjausta helpottamaan asennettiin ProBuilder lisäosa. ProBuilderilla pystyy monipuolisesti muokkaamaan 3D-malleja suoraan Unity-editorissa. ProBuilder on uudemmissa Unity-versioissa sisään rakennettu ominaisuus, mutta meidän käyttämässä se piti vielä ladata Unity Asset Storesta. Yksi yleinen virhe on (Kuva 20) lattian vajaaksi jääminen, jolloin raosta selkeästi näkyy maana toimiva terrain-objekti. Vajaaksi jääneet 3D-mallit saatiin näppärästi korjattua valitsemalla vajaan sivun tahko ProBuilderin tahkovalintaa käyttämällä ja vetämällä (Kuva 21) tarvittava matka editorissa. Näitä vajaaksi jääneitä kohtia oli useammalla sivulla ja kaikki sai korjattua samalla menetelmällä.

Toinen virhe koski kaikkia portaita (Kuva 22). Portaat eivät aivan tulleet ylös asti toisen kerroksen tasanteelle. Portaat saatiin korjattua liikuttamalla niitä hieman ylöspäin ja hiukan skaalausta säätämällä.



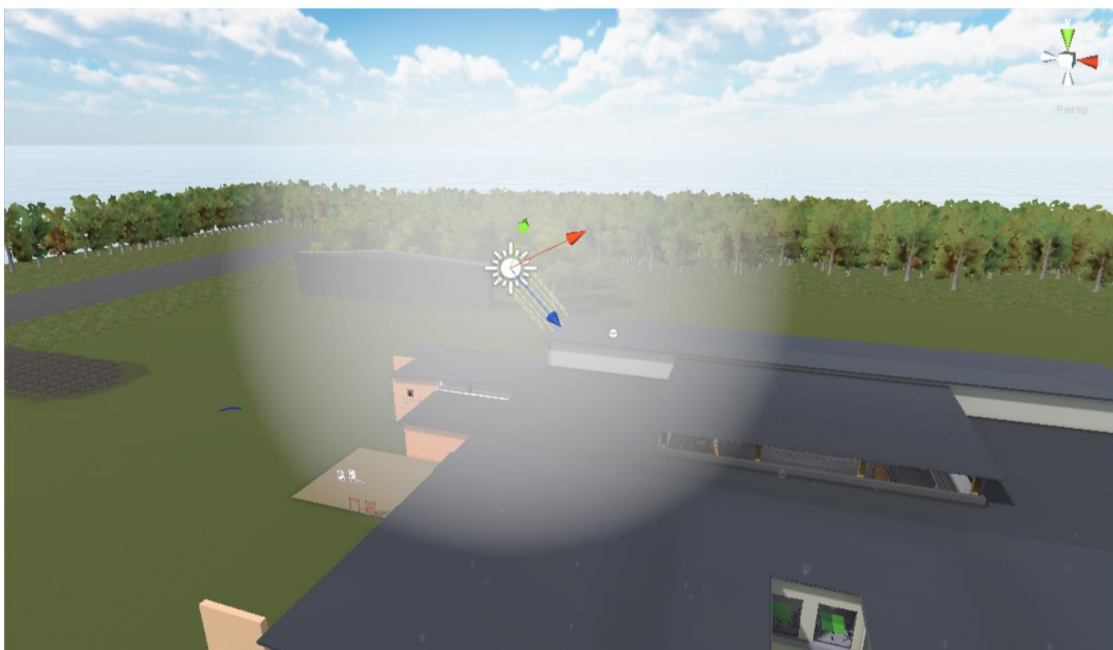
Kuva 21. Vajaaksi jäänyt sivu korjattuna.



Kuva 22. Vajaa porras.

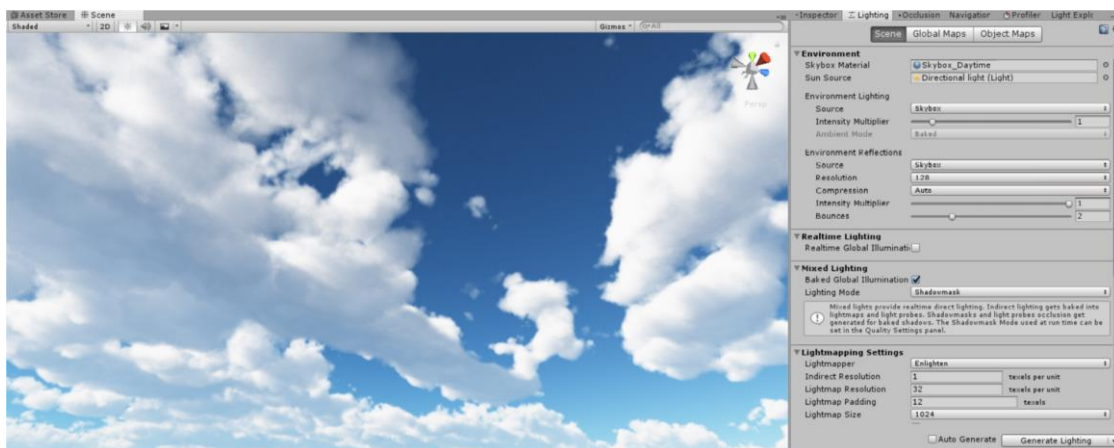
Unityssa on kuusi erilaista tapaa tehdä valoja, Point Light, Spot Light, Directional Light, Area Light, emissiiviset materiaalit ja Ambient Light. Näistä käytettiin kaikkia muita paitsi Area Lightia. [7, Types of light].

Directional Light kuvastaa valoa, joka on äärettömän kaukana, sitä yleensä käytetään auringonvalon kuvaamiseen. Directional Lightin sijoituspaikalla ei ole väliä, ainoastaan missä kulmassa valo osoittaa objektiin, vaikuttaa lopputulokseen.



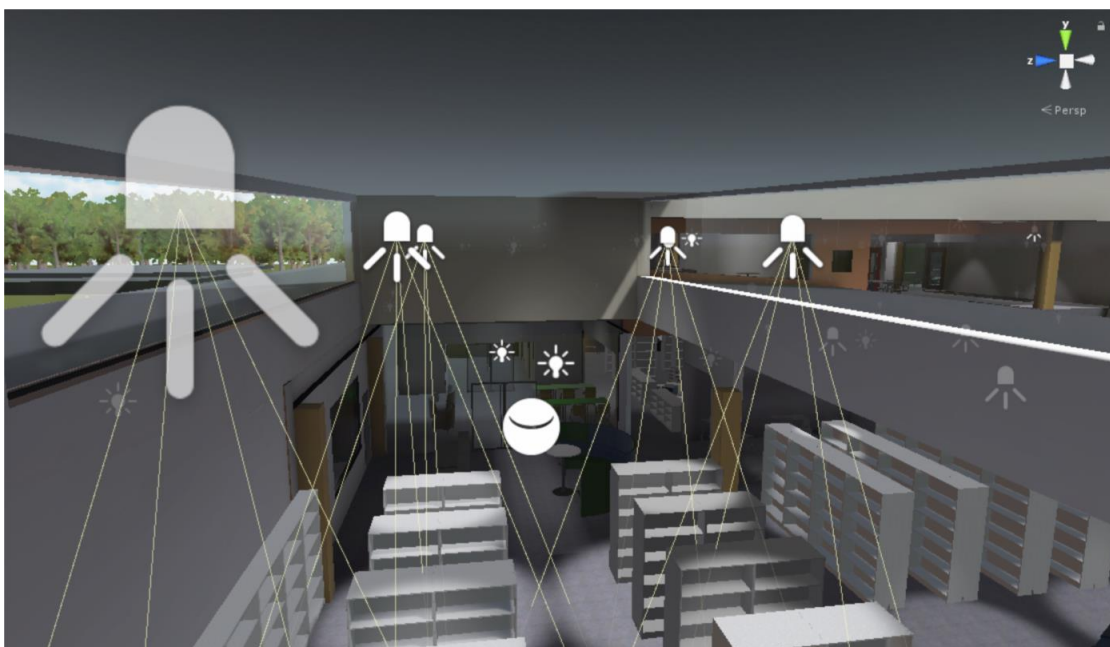
Kuva 23. Direction Light halo-efektillä.

Työssämme Directional Lightia käytettiin kuvastamaan aurinkoa (Kuva 23), joka oli asetettu kulmaan, jossa se valaisisi Sceneä mahdollisimman hyvin, sillä halusimme mahdollisimman paljon päivänvaloa Sceneemme.



Kuva 24. Taivaslaatikko (engl skybox) ja Ambient Light asetukset.

Taivaslaatikkoa varten lataimme Unity Asset Storesta PROASSETS käyttäjän Free HDR Sky -paketin. Paketti sisälsi kaksi taivaslaatikkoa, joista toisessa on yötaivas ja toisessa päivätaivas. Näistä otimme käyttöön (Kuva 24) päivätaivaan. Unityssa taivaan ulkonäkö määrittelee kuinka paljon se tuo Ambient-valoa Sceneen.

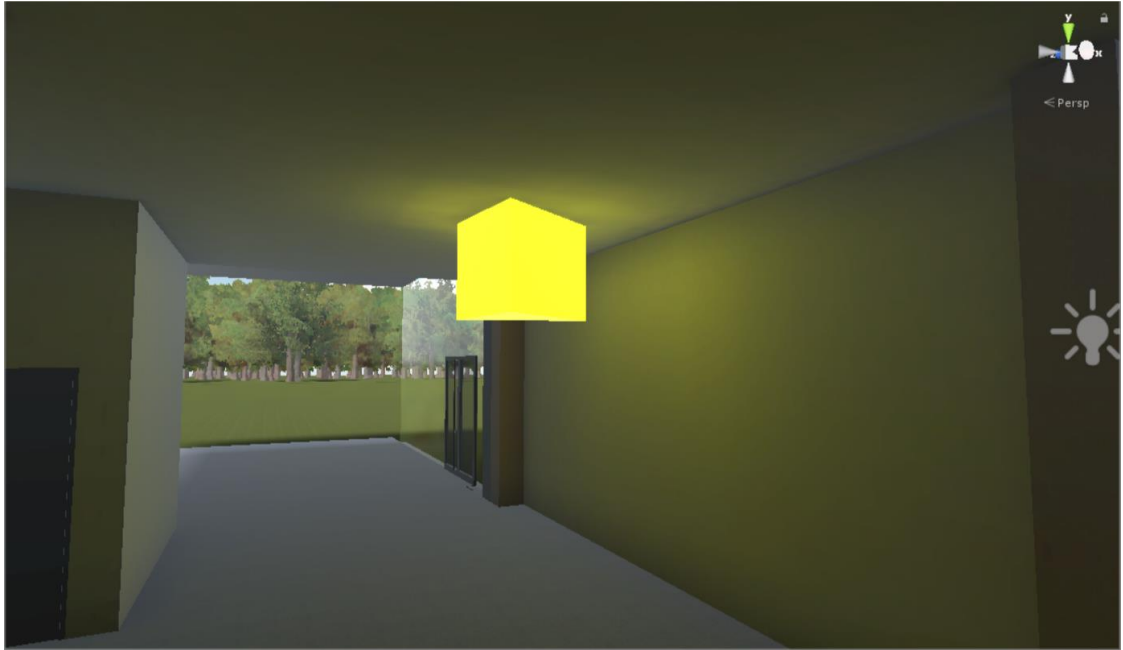


Kuva 25. Kuvassa olevat härpäkkeet kolmella viivalla kuvaavat Spot Lighteja ja ääri viivat näyttävät valaistavan alueen.



Kuva 26. Hehkulamppu kuvake kuvaa Point Lightia ja ääriviivat näyttävät valaistavan alueen.

Point- ja Spot Lighteja lisättiin Sceneen useita ja niiden tehtävä on toimia normaaleina sisätilan lamppuina. Point Light on valo (Kuva 25) joka valaisee joka suuntaan ympärilleen tasaisesti, mitä kauempana valosta sitä himmeämpi ja rajojen (Kuva 25) ulkopuolella se ei valaise yhtään. Point Lighteja lisäsimme Sceneen selkeästi eniten. Lähes jokaiseen huoneeseen ja käytävään lisättiin oma Point Light. Spot Light (Kuva 26) on valo, joka valaisee kartion muotoisella alueella. Spot Lighteja lisäsimme kohtiin, joihin ne voisivat sopia esimerkiksi kirjastoon näyttämään kirjahyllyjen välit.



Kuva 27. Itse tehty lamppu keltaisella emissiivisellä materiaalilla.

Lisäsimme (Kuva 27) myös testimielessä emissiivisellä materiaalilla toimivan valon. Kyseisen valon voisi korvata Point Lightilla ja saada suunnilleen saman lopputuloksen.

6 Optimointi

6.1 Optimoinnin tarve

Opinnäytetyön edetessä emme oikeastaan missään vaiheessa ajatelleet tarvitsemamme optimointia, meillä oli tehokas työasema käytettävissä eikä tavoite ollut saada fotorealistista grafiikkaa. Näistä seikoista huolimatta suorituskyky alkoi heiketä. Viven näyttöjen yhteen laskettu resoluutio on 2160x1200 ja renderöinti resoluutio on 3024x1680, koska VR:n kanssa renderöidään aina yli näytön resoluution kuvanlaadun ja paremman pään seurannan takaamiseksi. SteamVR:aa käyttävien VR-lasien renderöinti resoluutio löytyy SteamVR asetuksista video välilehdeltä. Näyttöjen virkistystaajuus on 90Hz ja on suositeltua, että VR-ohjelmat toimisivat vähintään 90 FPS parhaan kuvan ja sulavuuden

saamiseksi. Jos ohjelma ei kykene pyörimään 90 FPS se ohjelmallisesti tiputetaan 45 FPS ja ruudut tuplataan algoritmin avulla. Tämä aiheuttaa haamukuvaa ja muita kuvaongelmia.

Lopulta optimoinnin tarve syntyi monen asian myötävaikutuksesta, resoluutio yhdessä 90:n FPS suosituksen kanssa, revit-mallista exportatut 3D-mallit eivät olleet kovinkaan hyvin optimoituja ja niitä oli paljon eikä valojenkaan kanssa säästely. [8, Performance Optimization].

6.2 Käytetyt optimointikeinot

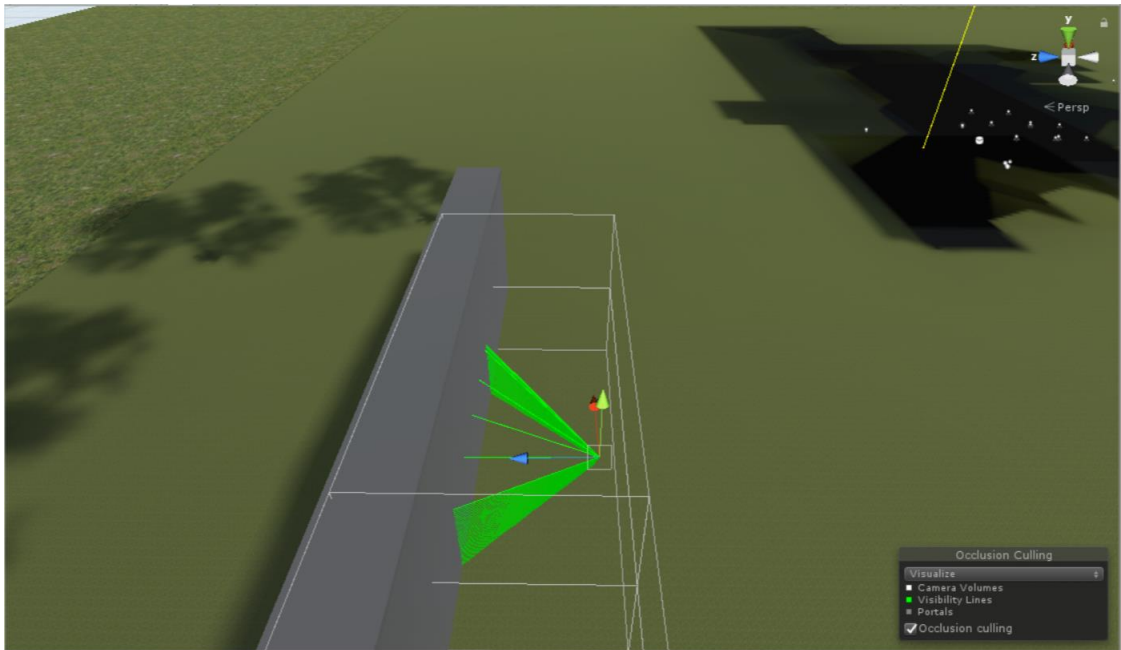
Unityssa on lukuisia keinoja optimoida peliä pyörimään paremmin. Tavoitteemme oli saada peli toimimaan aina minimissään 90fps. Tämän onnistuimme saavuttamaan melko pienellä panostuksella optimointiin.

Cullingilla tarkoitetaan asioiden renderöimättä jättämistä. Käytimme kolmea eri tyyppistä cullingia, Occulsion-, Frostrum- ja Manualculling. [8, Performance Optimization].



Kuva 28. Kuvassa kamera katsoo puita kohden. Taustalla oleva musta läntti on varjot koulumallista, jota ei renderöidä koska Frostrum Culling on päällä.

Frostrum Culling (Kuva 28) on vakiona Unityssa päällä. Sillä tarkoitetaan sitä, että ei renderöidä kokonaan näkökentän ulkopuolella olevia kohteita ollenkaan.

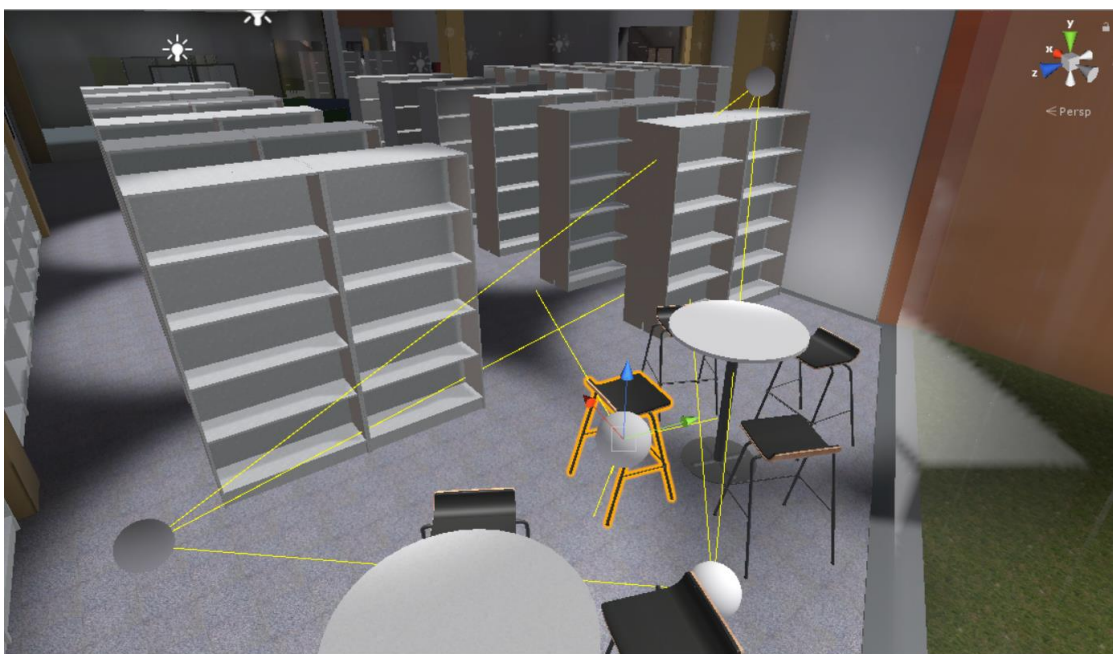


Kuva 29. Kun sekä Occlusion Culling että Frostrum Culling on päällä, ei puita eikä koulumallia renderöidä.

Occlusion Culling (Kuva 29) on tekniikka, jossa leivotaan etukäteen näköalueita. Näitä alueita sitten luetaan pelin suorituksen aikana ja jätetään renderöimättä ne objektit, joita ei nähdä. Manual Cullingilla tarkoitetaan, että eri objekteille on asetettu arvot, kuinka kaukana pelaajasta niiden pitää olla, että ne renderöidään.



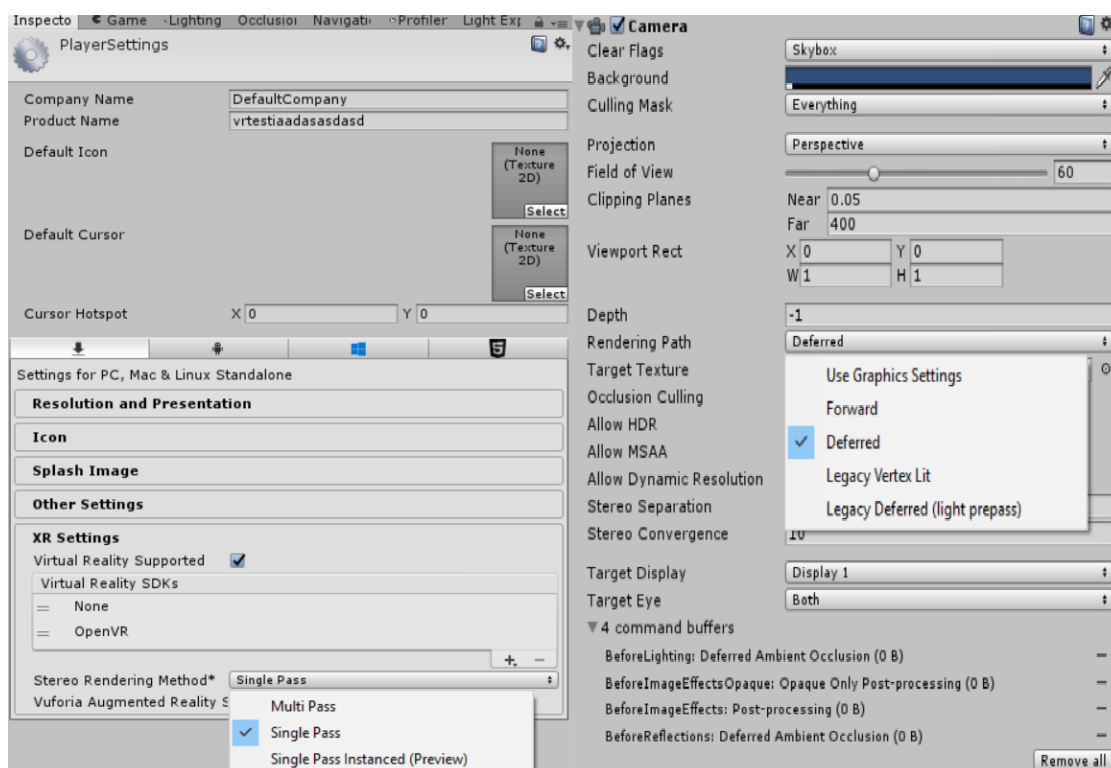
Kuva 30. Light Probeja ja niiden muodostamia valaistus alueita.



Kuva 31. Tuoli ja sen valoarvon määrittävät Ligh Probet.

Aluksi, kun lisäilimme valoja koulumalli Sceneen, alkoi suorituskyky laskea nopeasti. Tämä johtui siitä, että kaikki valot olivat real time eli niitä laskettiin reaaliajassa, jolloin ne loivat varjoja huonokosti optimoituihin 3D-malleihin. Valoja lähdettiin korjaamaan laittamalla kaikki valot mixed tilaan, jolloin paljon suorituskykyä käyttävä epäsuora valo lasketaan etukäteen joko Light Probeen (Kuvat

30 ja 31) tai Light Mappiin, tällöin valoisuus luetaan valmiiksi laskettuna käytön aikana säästään huomattavasti resursseja. Aluksi kaikki valot loivat varjoja, tämä aiheutti suuria suorituskyky ongelmia, sillä suoraan Revitistä konvertoituneet 3D-mallit eivät olleet hyvin optimoituja. Päädyimme lopulta poistamaan varjojen luonnin kaikista muista valoista paitsi Directional Light:sta. Tämä paransi suorituskykyä huomattavasti eikä juurikaan vaikuttanut ulkonäköön millään tavalla.



Kuva 32. Single Pass -rendering valinta player settings välilehdessä ja Deferred rendering -path valinta kameran ominaisuuksissa.

Meidän kannalta helppoja suorituskykyä nostavia keinoja olivat rendering pathin ja rendering method:in vaihtaminen (Kuva 32) Deferred:ksi ja Single Pass:ksi. Nämä saatiin aktivoitua menemällä playersettings ja kameran ominaisuudet valikkoihin ja valitsemalla ne alasetoalvikosta.

Näillä optimointi keinoilla saatiin vakaasti suurempi kuin 90 FPS, joka oli meille täysin riittävä. Unitystä ja yleensäkin pelien kehittämisessä on lukuisia keinoja optimoida peli toimimaan paremmin. Tässä kappaleessa käytyt keinot eivät

vielä edes raapaisseet pintaakaan Unityn optimointi potentiaalista. [8, Performance Optimization].

7 Ohjelman Rakenne

7.1 Unity Scene

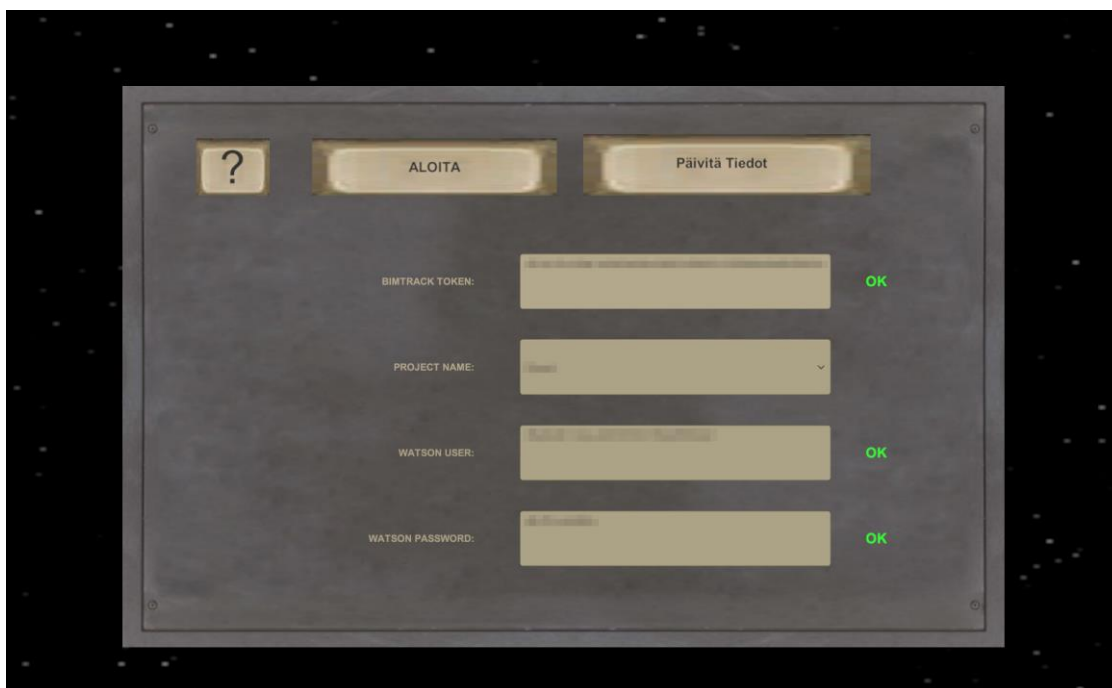
Unityssä Scene sisältää ympäristöjä ja valikkoja. Scene sisältää kaikki tarvittavat tiedot yhden kentän pyörittämiseen. Esimerkiksi ohjelman alkaessa vain päävalikko Scene avautuu. Vasta ohjelman ensimmäisen tason alkaessa se latautuu tietokoneen käyttömuistiin ja päävalikon käyttämä muisti vapautuu [9, Unity Manual].

7.2 Päävalikko

Päävalikon toiminta tässä ohjelmassa on tarkistaa BIM- ja Watson-tunnisteet. Ohjelman alkaessa näkyy läjä tekstikenttiä, joihin käyttäjä voi syöttää BIM-tunnisteen, Watson-käyttäjätunnisteen ja salasanan. Yksi kentistä on alasvetovalikko, josta valitaan BIM Track -projekti. Jos tunnisteet ovat olemassa, luetaan ne käyttäjän kovalevyllä sijaitsevasta XML-tiedostosta valmiiksi tekstikenttiin.

Jos käyttäjä syöttää tietoja ensimmäistä kertaa, ohjelma luo XML-tiedoston ohjelman ajotiedoston kansioon. Tietojen päivitys avaa olemassa olevan XML-tiedoston, päivittää tiedot ja sulkee XML-tiedoston.

Kun tietoja luodaan, tai päivitetään, tarkistaa ohjelma BIM ja Watson tunnisteet. Tekstikenttien viereen ilmestyy vihreä OK, tai punainen X riippuen tekstikenttien totuudenmukaisuudesta. Kun kaikki tiedot ovat ok, näyttää päävalikko alla olevan kuvan mukaiselta.



Kuva 33. Oikeat tiedot syötettynä tekstikenttiin.

Vihreä OK merkki tarkoittaa tiedon olevan käyttökelpoinen. Punainen X tarkoittaa tiedon olevan käyttökeltoton. Jos tiedot eivät ole kentissä, ohjelman voi silti ajaa, mutta kaikki ohjelman ominaisuudet eivät tosin toimi. Painamalla aloita nappia alkaa Koulumalli Scene.

7.3 Koulumalli

Päävalikon jälkeen käyttäjän eteen ilmestyy koulumalli vuorovaikutusympäristö. Koulumalli (Kuva 34) on saatu käyttöön opinnäytetyötä varten ARKPII-arkkitehtitoimistolta. Koulun 3D-malli pohjautuu suunnitelmiin Tohmajärvellä sijaitsevasta Kemien koulun uudelleenrakennuksesta. Unityssä on lisätty koulun ympärille maastoa, puita ja hiukan interaktiivista ympäristön kanssa. Koulumalli on tärkein Scene koko ohjelmassa, koska se sisältää kaiken toiminnan koulumallin kanssa. Koulumalli Scene sisältää myös BIM Track -integraation, joka on yksi tämän opinnäytetyön ydinasioista. Käyttäjä aloittaa VR-osuuden koulumallin etuoven läheltä.



Kuva 34. Koulumalli Unitylla luodussa Scenessä.

Käyttäjällä on käytössä useita työkaluja vuorovaikutuksen mahdollistamiseksi. Käyttäjä aloittaa liikkumistyökalulla. Liikkuminen tapahtuu painamalla kosketusnäppäintä. Liikkumistyökalu toimii myös tarttumistyökaluna ja käyttäjä voi tarttua joihinkin esineisiin kouluympäristössä.

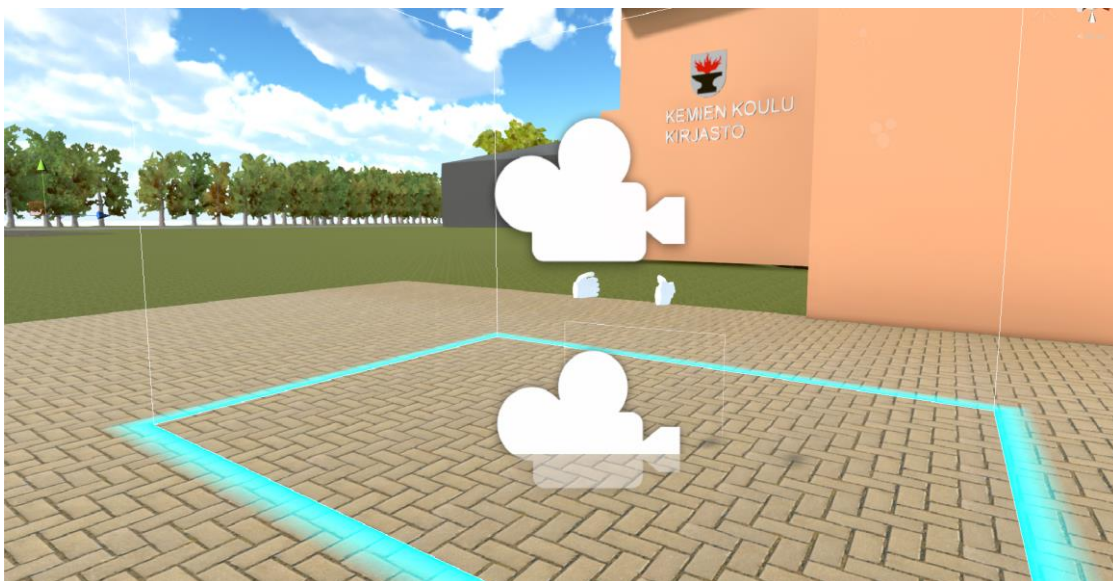
8 VR-toiminnallisuuden toteuttaminen Unityssa

8.1 VR:n käyttöönotto

VR-toiminnallisuus toteutettiin käyttämällä SteamVR API:a. SteamVR API toiminnallisuus saatiin käyttöön Unityssa lataamalla Asset Storesta SteamVR Plugin, joka oli versioltaan 1.2.3.

Perus VR-toiminnallisuus saatiin lisäämällä Unity sceneen SteamVR pluginin mukana oleva CameraRig-komponentti (Kuva 35). CameraRig tuo sceneen tuen VR-laseille ja ohjaimille. Sen avulla laseja ja ohjaimia voidaan seurata tarkasti 6-DOF tasolla 3D-avaruudessa. CameraRigin mukana ei tule minkäänlai-

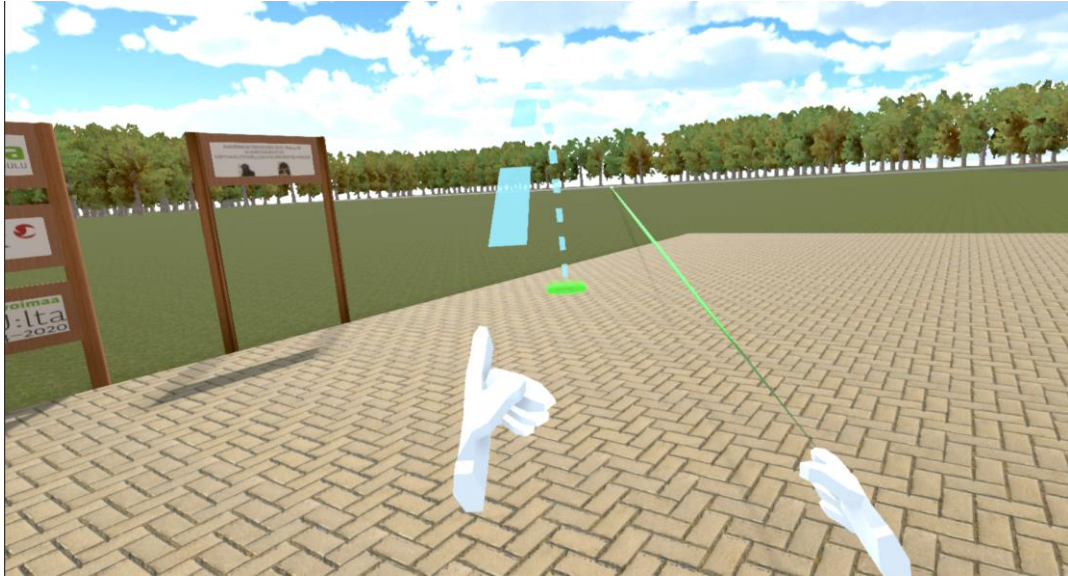
sia liikkumiskeinoja tai muuta toiminnallisuutta, vaan se on lähtökohta omien järjestelmien rakentamiseen.



Kuva 35. Unity CameraRig-komponentti, ylempi videokamera-ikoni on käyttäjän päätä seuraava kamera ja kädet ovat liikeohjattavia ohjaimia.

8.2 Teleporttaus järjestelmät

Koska virtuaalimaailma on lähes aina suurempi kuin fyysinen maailma tarvitsee VR-ohjelmat jonkinlaisen liikkumiskeinon liikkumista varten. Päädyimme teleportti liikkumiseen koska sen avulla voidaan nopeasti ja vaivattomasti liikkua suuriakin matkoja. Työhömmme implementoimme kaksi erilaista (Kuva 36) teleporttijärjestelmää, lähelle tarkkaa liikkumista varten ViveTeleporter-teleportti [10, HTC Vive Teleportation System with Arc Pointer] ja kauas RayWenderlich (ks. luku 2) ohjeesta jatkojalostettu teleportti.



Kuva 36. Vasemmalla ViveTeleporter ja oikealla RayWenderlich tutoriaalin pohjalta tehty teleporttijärjestelmä.

ViveTeleporter otettiin käyttöön lataamalla ViveTeleporter-paketti Githubista. Paketti sisältää valmiiksi määriteltyjä skripteja ja GameObjecteja, jotka lisäämällä sceneen teleportaus funktio saatiin toimimaan. ViveTeleporter käyttää Unityn Navmeshia (Kuva 37) määrittelemään minne pystyy liikkumaan ja minne ei. Toinen teleportteri toteutettiin RayWenderlich tutoriaalin pohjalta, sen toimintaperiaate on, että se lähettää näkymättömänsäteen erittäin kauas osoitus suuntaan. Jos säde törmää alueeseen, joka on määritelty, että siihen voi teleportata, silloin luodaan kädestä törmäys paikkaan (Kuva 36 vihreä säde) näkyvässä, joka viittaa siihen, että teleportaatio paikkaan on mahdollista.



Kuva 37. Sininen alue on Unity Navmesh, joka määrittelee minne ViveTeleporter:lla voi liikkua.

8.3 Valikkorakenne ja työkalunvaihtaja

Käytimme työssämme työkaluja erilaisten toiminnallisuuden toteuttamiseen. Työkalujen vaihtoa varten meidän tarvitsi rakentaa työkalunvaihtaja toiminnallisuus ja valikko, jolla sitä käytettiin.

Työkaluvalikko (Kuva 37) avautuu painamalla otenäppäintä (ks. luku 3.4) ja käynnistää valikon GameObjectit (Kuva 38 siniset). Se pysyy auki niin kauan kuin otenäppäintä pidetään pohjassa. Valikossa on neljä 3D-mallia, jotka kuvaavat työkaluja, jotka niistä avautuvat. Työkalunvalinta prosessi tapahtuu niin, että käyttäjä pitää otenäppäintä pohjassa ja liikuttaa ohjaimen halutun 3D-mallin sisään, tällöin valikko lähettää tiedon työkalunvaihtajalle mikä työkalu (Kuva 38 punaiset) pitää laittaa päälle ja mikä pois, jolloin ohjaimen 3D-malli muuttuu kuvastamaan valittua työkalua.



Kuva 38. Unityn hierarkia välilehdellä valikko- ja työkalu -GameObjectit. Siniset ovat valikon objektit ja punaiset työkaluja.

8.4 Pelimaailman valikoiden toteuttaminen VRTK:n avulla

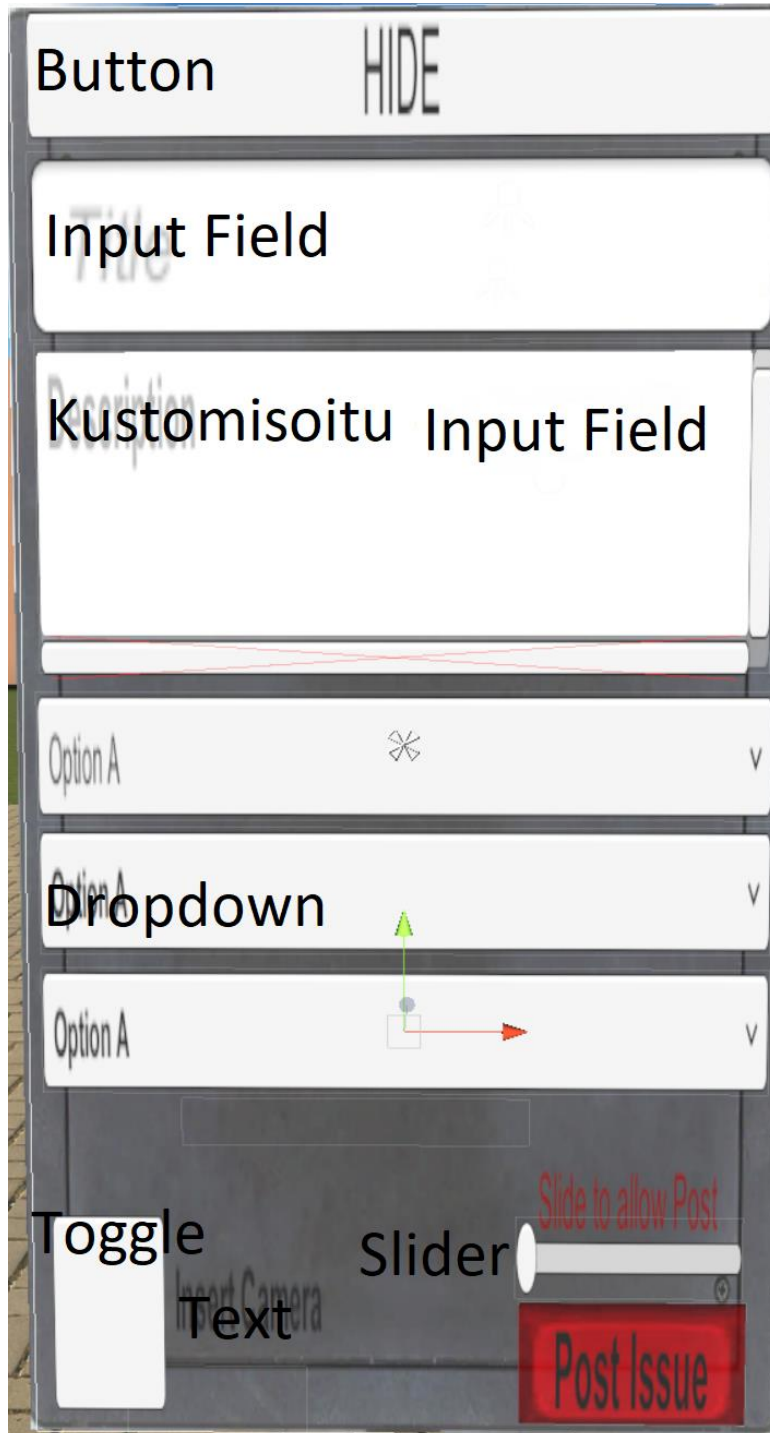
Teimme kolme Unity UI -järjestelmää käyttävää valikkoa, joiden avulla vuorovaikutettiin BIM Track -projektipankkipalvelun kanssa VR:n sisällä. Unity UI -järjestelmä on Unityyn sisäänrakennettu apuväline käyttöliittymän suunnitteluun. Se sisältää valmiita GameObjekteja yleisimpiin käyttöliittymän tarkoituksiin. Unity UI:sta (Kuva 39) käytimme Button-, Dropdown-, Input Field-, Text-, Slider- ja Toggle -GameObjekteja. [11, UI].

Button-GameObjekteja valikkoihin tuli eniten, sillä näppäimistön jokainen kirjain on yksi Button. Buttonin toimintaperiaate on yksinkertainen, se on nappi, kun

painat sitä, jotain tapahtuu. Näppäimistöissä napin painallus syöttää syötteeseen painettua nappia vastaavan kirjaimen. [11, UI.]

Työssämme Input Field-, Text- ja Dropdown -Gameobjektit sisältävät kaiken tiedon, jota käyttäjälle näytettiin. Input Field- ja Text -GameObjektit luovat kummatkin tekstilaatikon, johon voidaan kirjoittaa tekstiä. Ero näiden kahden välillä on siinä, että Input Field on suunniteltu, että siihen syötetään jotain tekstiä ohjelman aikana, kun taas Text:ssa teksti on valmiiksi syötettyä etukäteen. Yksi Input Field käyttää kustomisoitua koodia, jonka avulla Input Field saatiin hyvin toimivaan, kun syötettyä tekstiä oli paljon [13, illad3d]. Dropdown on alavetovalikko, siihen tallennettiin erilaisia ennalta määritetty arvoja käyttäjälle valittavaksi. [11, UI.]

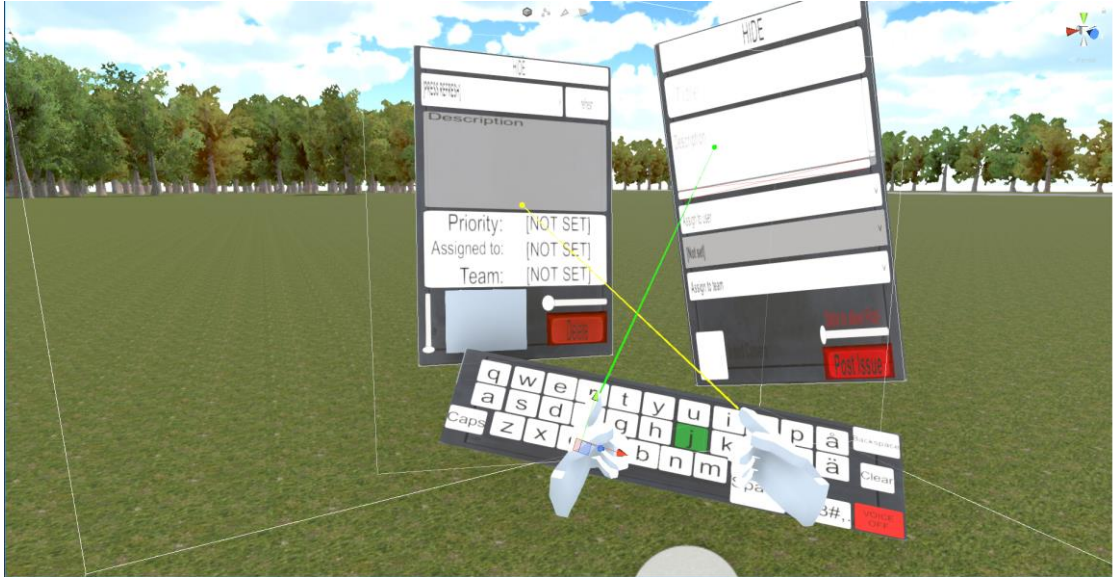
Slider- ja Toggle -GameObjekteja käytettiin muutamaan kertaan, näistä slidereita käytettiin varmistuksena, että ei tapahtuisi vahinko painalluksia. Togglella avattiin valikko, jossa on lisäominaisuuksia. [11, UI.]



Kuva 39. Unity UI -GameObjektit.

Jotta Unity UI -valikkojen kanssa vuorovaikutus VR:ssa saatiin toteutettua, asensimme VRTK-lisäosan, VRTK sisältää lukuisia hyödyllisiä ja helppokäyttöisiä skripteja VR-ohjelmien tehokkaaseen toteuttamiseen. Työssämme käytetty VRTK versio oli 3.2.1 [12, VRTK - Virtual Reality Toolkit].

VRTK:sta käytimme VRTK_UI Canvas-, VRTK_StraightPointerRenderer-, VRTK_Pointer-, VRTK_UI Pointer-skripteja. Näistä VRTK_UI Canvas laitettiin jokaiseen (Kuva 40) pelimaailman valikkoon, kyseisen skriptin funktio on määrittää mitkä Unity UI -valikoista ovat käytettävissä VRTK_UI Pointerilla. Lopuilla skripteilla luotiin kaksi käsityökalua (ks luvut 9.2-9.3) joilla valikoiden kanssa pystyi vuorovaikuttamaan VR:n sisällä.



Kuva 40. Unity UI -järjestelmällä yhdessä VRTK UI -skriptien kanssa tehdyt pelimaailman valikot. Käsiin on tehty VRTK-skripteillä toiminnallisuus valikkojen kanssa vuorovaikuttamiseen.

9 Vuorovaikutus Unityssa

9.1 Käsityökalu eli oletustyökalu

Ohjelmassa oletustyökaluna toimii käsityökalu. Käsityökalua liikutetaan liikuttamalla ohjainta. Työkalu toimii myös telepottina, liikkumista varten ja sillä voi tarttua objekteihin.

Kädessä on itseasiassa kaksi teleportti ominaisuutta. Kosketuslevyä painamalla voidaan liikkua lähelle. Kun levy painetaan pohjaan, ilmestyy paraabelimainen säde (Kuva 36 vasen säde) ja kun päästät levyn pohjasta teleportaat säteen osoittamaan paikkaan. Säde lähtee käden etusormesta ja päättyy lopulta maahan. Säteen päässä on pyöreä valoalue. Jos valoalue on punainen, on teleporttaaminen mahdotonta. Koska säde on paraabelimainen, teleporttaus onnistuu vain lähelle.

Toinen teleporttityökalu (Kuva 36 oikea säde) on valikkonapissa. Tässä teleporttissa kädestä lähtee suoraan säde käden osoittamaan paikkaan. Tällä teleporttityökalulla voi liikkua niin kauas kuin pelialuetta riittää. Painamalla valikkonäppäintä ja osoittamalla pintaan säde syntyy, päästämällä valikkonäppäin pohjasta, kun säteen loppupään alue on vihreä, teleporttaat osoittamaasi paikkaan.

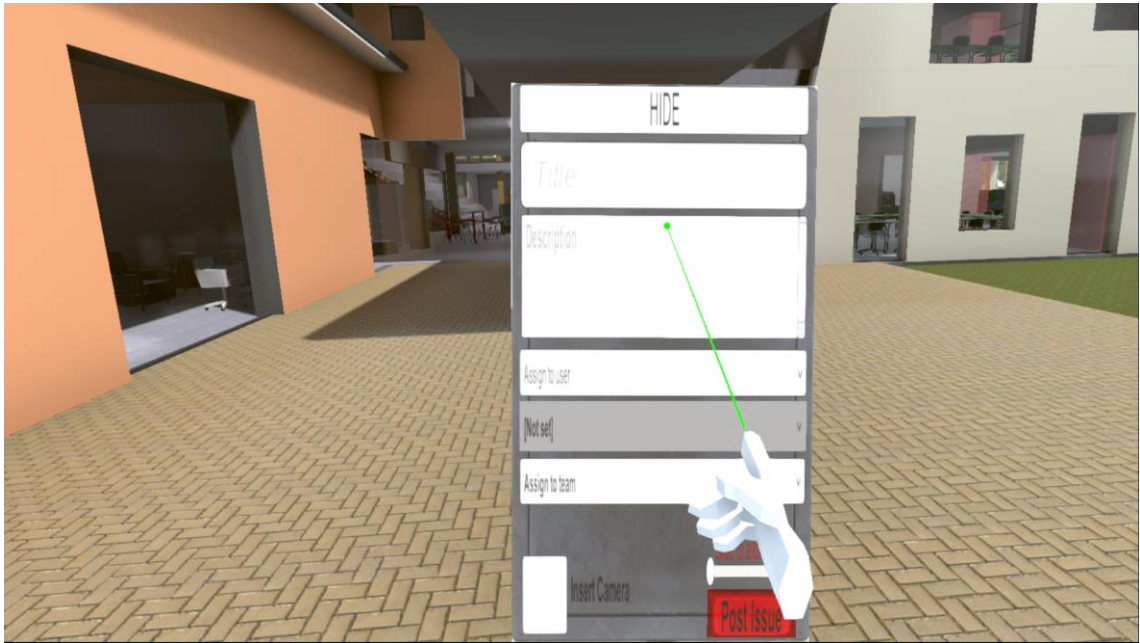
Ohjaimen liipaisinta painamalla käsityökalu voi tarttua esineisiin. Jos käsi menee esineen kohdalla hieman puristusasentoon, voidaan liipaisinta siinä tapauksessa painaa. Liipaisimen painaminen vie käden kunnolliseen puristusasentoon ja esine tarttuu silloin käteen. Liipaisimen vapauttaminen vapauttaa myös tarratun esineen. Kun käyttäjä painaa otenäppäintä missä tahansa työkalussa, valitaan oletustyökalu ja käden ympärille piirretään (Kuva 41) valikko, josta voi valita haluamansa työkalun.



Kuva 41. Kuvassa etualalla oletus työkalu ja sen ympärille piirtynyt valikko.

9.2 Kehitystehtävän lisästyökalu

Työkaluvalikossa kehitystehtävän lisästyökalu näkyy dokumenttipaperi 3D-mallina, jossa on oikeassa alakulmassa plusmerkki. Käyttäjän valitessa kyseessä olevan työkalun, käyttäjän käsiohjain menee osoitus muotoon ja etusormesta loistaa vihreä laserosoitin. Painamalla valikkonappia käyttäjä saa eteensä kehitystehtävänlisäys käyttöliittymän (Kuva 42).



Kuva 42. Kehitystehtävän lisästyökalun käyttöliittymä.

Käyttöliittymä koostuu tekstilaatikoista, napeista, slidereista, alasetoalikoista ja togglelaatikoista. Käyttäjä voi kirjoittaa tekstilaatikkoihin otsikon kehitystehtävästä ja tietoa ongelmasta. Alasetoalikoista löytyy valinnat kehitystehtävän kriittisyyteen, kenelle henkilölle ja mille ryhmälle se on osoitettu. Kriittisyys laatikon väri muuttuu sen mukaan, miten on Bim Track:ssä säädetty kriittisyyden väri.

Käyttäjän painettua tekstilaatikkoa, ilmestyy käyttäjän eteen näppäimistö. Käyttäjä voi tähdätä osoitinviivalla haluamaansa näppäintä ja painaa liipaisinta. Tämä syöttää painetun kirjaimen tekstilaatikkoon. Näppäimistössä on myös nappi, jossa lukee "voice". Voice-napilla voi puhua mikrofoniin englanniksi ja puhuttu lause, tai sana kirjoittuu automaattisesti valittuun tekstilaatikkoon. Huomioi, että voice-toiminto tarvitsee IBM Watson -käyttäjätunnuksen toimiakseen.

Kun käyttäjä on tehnyt vähintään otsikon kehitystehtävän, voi käyttäjä lähettää kehitystehtävän eteenpäin. Käyttäjän on ensin liikutettava POST-napin yllä oleva liu'utin vasemmalta oikealle, että käyttöliittymä sallii postauksen. Postauksen tapahtuessa käyttäjä saa OK-viestin tai Error-viestin riippuen BIM Trackin antamasta vastauksesta. Error viestit ovat tyypiltään yleisiä HTTP Error -koodeja.

9.3 Kehitystehtävän historiatyökalu

Kehitystehtävän historiatyökalun valitessa sormi muuttuu osoittimeksi. Etusormen päästä lähtee keltainen säde, jolla voi osoittaa valikon elementtejä (Kuva 43). Kehitystehtävän historia valikko ei kuitenkaan ilmesty historia työkalua valitessa, vaan käyttäjän on erikseen painettava menu nappia. Nappia painaessa, historia valikko ilmestyy käyttäjän luokse. Käyttäjä voi valita haluamansa kehitystehtävän alavetovalikosta sen jälkeen, kun on painanut, päivitä nappia valikossa. Käyttäjän valittua kehitystehtävän, valikon tekstilaatikot täyttyvät tiedoilla, kuten ongelman kuvaus, kriittisyys ja kelle ryhmälle ja/tai käyttäjälle kehitystehtävä on tarkoitettu. Kriittisyystietokohdassa myös BIM Trackissa määritelty kriittisyyden väri näkyy käyttäjälle.



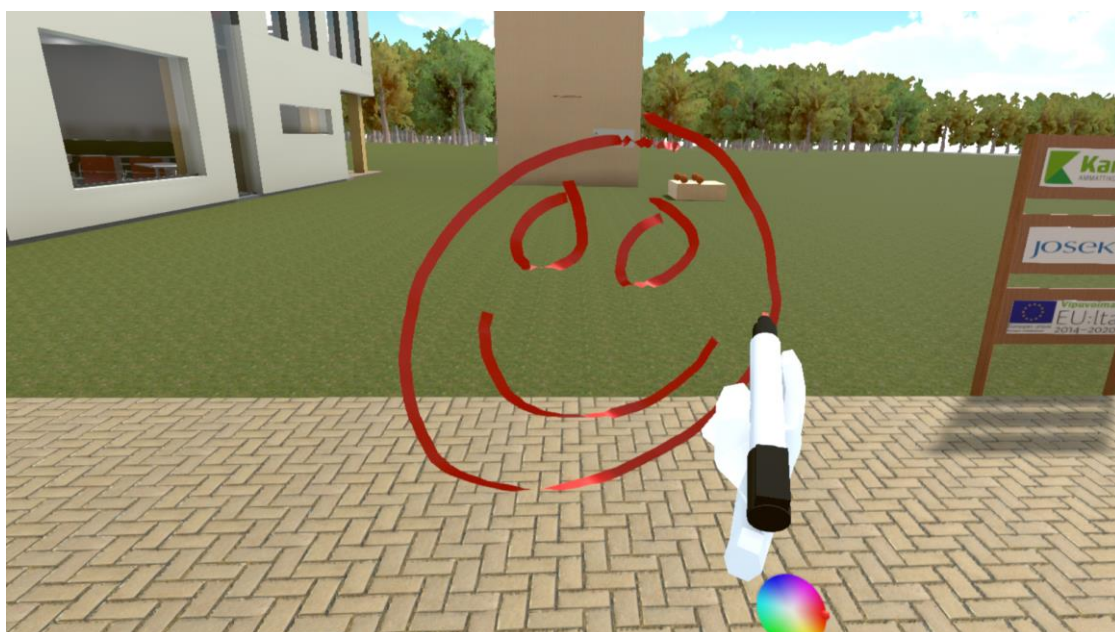
Kuva 43. Kehitystehtävä historia työkalun valikko.

Työkalun valikossa on vasemmassa alakulmassa slideri, joka siirtää käyttäjän välittömästi kehitystehtävän koordinaatteihin. Koordinaatit ovat samat, jotka on määritelty käyttäjän tehdessä kehitystehtävä kehitystehtävän lisästyökalulla, taikka BIMTrackissa.

Oikeassa alakulmassa on myös delete-nappi, jolla voi poistaa valitun ticketin. Delete-napin ohessa on slideri, joka on vedettävä sivuun, että delete-nappi aktivoituu. Käyttäjän painettua delete-nappia, ticketti siirtyy arkistoon ja ei ole enää käyttäjälle näkyvissä. Kehitystehtävä voidaan poistaa pysyvästi BIM Track -palveluun kirjautumalla.

9.4 Piirtotyökalu

Piirtotyökalun valitessa käsityökalun 3D-malli muuttuu hieman. Käteen ilmestyy kynä käteen. Tämän lisäksi ilmaan ilmestyy pyöreänmuotoinen värivalitsin. Värivalitsimessa on pieni piste, joka osoittaa valitun värin. Sormen liikuttaminen kosketinkiekkon päällä liikuttaa pientä pistettä ja valittu väri muuttuu. Myös piste muuttaa väriään valitun värin väriseksi. Värin tummuutta voi säätää pitämällä kosketinkiekkoa pohjassa ja samalla liikuttamalla sormeä ylös tai alas. Liikuttamalla sormeä alas väri tummenee, ylöspäin liikuttamalla väri vaalenee. Kynän pää osoittaa myös valitun värin.



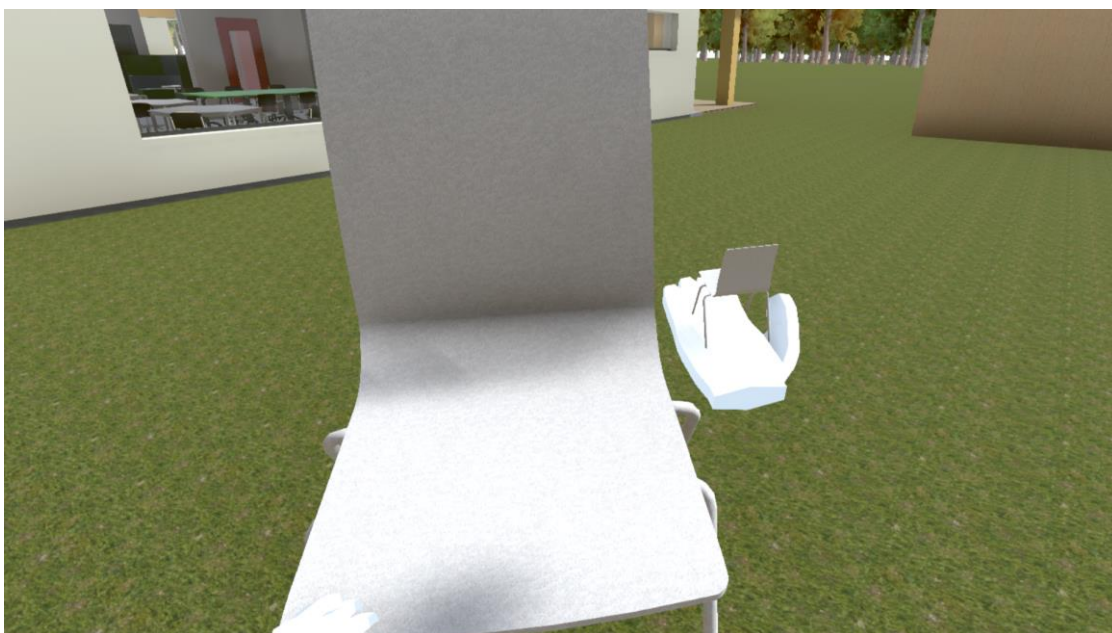
Kuva 44. Piirtotyökalu.

Kynän päästä lähtee piirtomateriaalia liipaisinta painaessa (Kuva 44). Piirtomateriaali on valitun värin värinen. Jokainen uusi painallus luo uuden materiaalin ilmaan. Materiaalissa ei ole painovoimaa, joten se jää leijumaan ilmaan.

Kynän piirtopaksutta voi säätää painamalla menu nappia. Jokainen painallus nostaa paksuutta 0,01 yksikköä, mikä vastaa yhtä senttimetriä.

9.5 Tavarankäytön lisätyökalu

Tavarankäytön lisätyökalu näyttää huonekalukuvastolta. Kun pelaaja asettaa työkaluvalikossa ohjaimensa kuvaston kohdalle, pelaajan sen hetkinen työkalu muuttuu tavarankäytön työkaluksi. Työkalu osoittaa valitun tavarankäytön käyttäjälle. Tavarankäytön on useampia ja haluamansa esineen voi valita painamalla kosketinnäppäintä oikealle ja vasemmalle. Tavara putoaa maahan, kun käyttäjä painaa liipaisinta (Kuva 45).



Kuva 45. Tavarankäytön lisätyökalu ja sillä pudotettu tuoli.

Tavarankäytön voi pudottaa niin paljon, kuin käyttäjä itse haluaa. Toisin sanoen, ei ole rajoja siinä, kuinka monta tavaraa käyttäjä voi pudottaa pelimaailmaan. Tavarankäytön on Unityn laskemat fysiikat. Käyttäjä voi tarttua oletustyökalulla tava-

roihin ja liikuttaa niitä. Myös tavaroiden heittäminen onnistuu. Tavarat törmäävät oletetusti muihin pelimaailman esineisiin, kuten seiniin, ikkunoihin ja portaisiin. Pudotetut tavarat törmäävät myös muihin pudotettuihin tavaroihin. Yksi esineistä, joita käyttäjä voi valita on niin sanottu taskulamppu. Taskulamppu on vain pallo, joka valaisee pelialuetta. Taskulamppuja voi pudottaa pelimaailmaan, ja toimii fyysiikoiltaan muiden esineiden tavoin.

10 Bimtrack integraatio

10.1 Tietoa BIM:stä

BIM-tyylinen konsepti sai alkunsa jo vuosikymmeniä sitten, mutta oli olemassa pikemminkin vain konseptina. Ennen 2000 lukua oli olemassa useita eri käytännön toteutuksia BIM konseptista, mutta 90-luvun ja 2000-luvun vaihteessa luotiin lopulta Open BIM, joka toimisi standardina. [14, A Brief History of BIM].

BIM käsittää muun muassa rakennuksen osat, kuten ovet, seinät, lattiat ja katot. Lisäksi se käsittää kustannusarviota, osien elinikää ja muuta rakennukseen liittyvää [14, A Brief History of BIM].

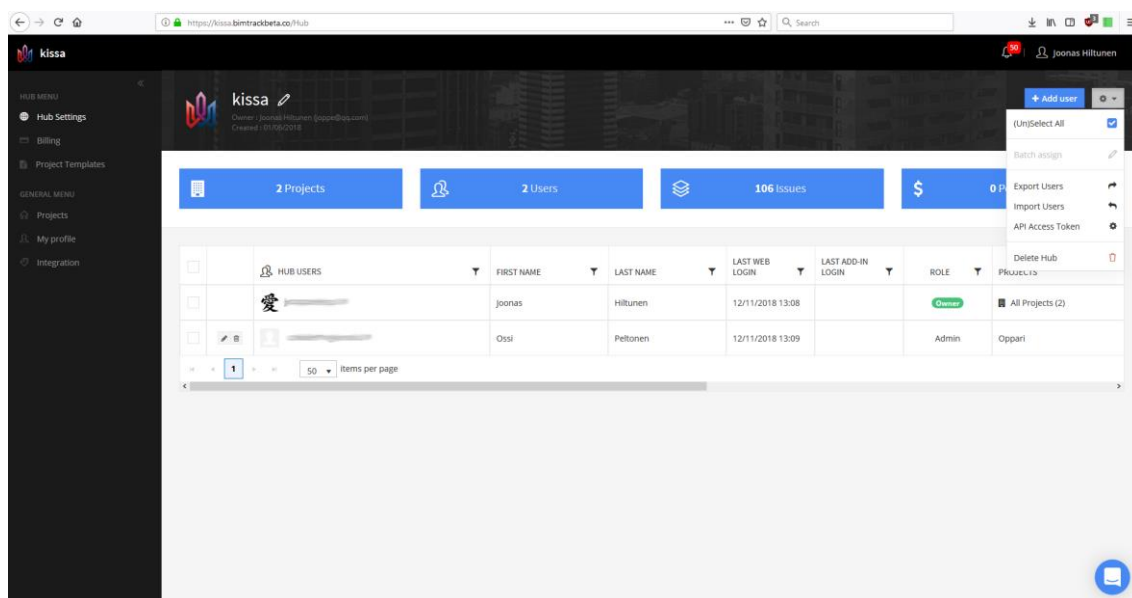
10.2 BIM Track

Bim Track on alusta BIM formaatille. Sitä kehittää kanadalainen BIM One inc. Bim Track:iin voi ladata kehitteillä olevan rakennuksen 3D-mallin, sekä tehdä kehitystehtäviä ja asettaa niitä määritetyille ryhmille tai yksilöille. Kehitystehtävään voi sisältyä lisäksi esimerkiksi rakennuksen ongelmakohtien koordinaatit, kuvakaappaus ongelmasta, taikka selitys ongelmasta tekstimuodossa.

10.3 Tunnusten luonti BIM Trackissa ja Watsonissa

Opinnäytetyötä varten BIM Track -autentikaation saamiseksi oli tarpeellista luoda tunnus BIM Track -palveluun. BIM Track -tunnuksen luonti tapahtui osoitteessa <https://bimtrackbeta.co/en/Login>. Käyttäjän luodessa tunnusta on kerrottava tunnistustietoja mukaan lukien nimensä, sähköpostiosoitteensa ja ammatinsa.

Käyttäjän luotua tunnus, on käyttäjän luotava hubi. Hubi on keskus BIM Track -projekteille. Kun hubi on luotu, voidaan hubin tokeni käydä hakemassa. Kirjoitushetkellä vain hubin luojalla on oikeus päästä käsiksi hubin tokeniin. Hubin tokenin löytää kohdasta hub settings. Hub settings ikkunassa tulee käyttäjän valita oikeasta yläkulmasta ratas, josta saa auki listan kuvan osoittamalla tavalla (Kuva 46).

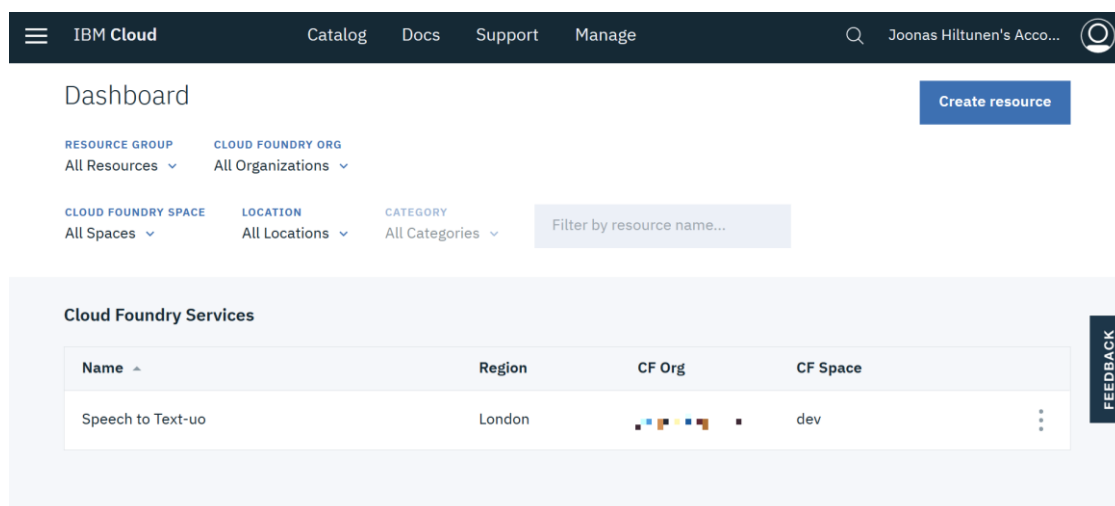


Kuva 46. Bim Track palvelun Hub settings -välilehti.

Käyttäjän painettua API Access Token, saa käyttäjä näytöllensä näkyviin BIM Track -Tokenin, jolla tunnistaudutaan VR-ohjelmassamme. Ensimmäisellä käynnistyskerralla tokeni syötetään päävalikossa ja tallennetaan XML-tiedostoon seuraavia ajoja varten.

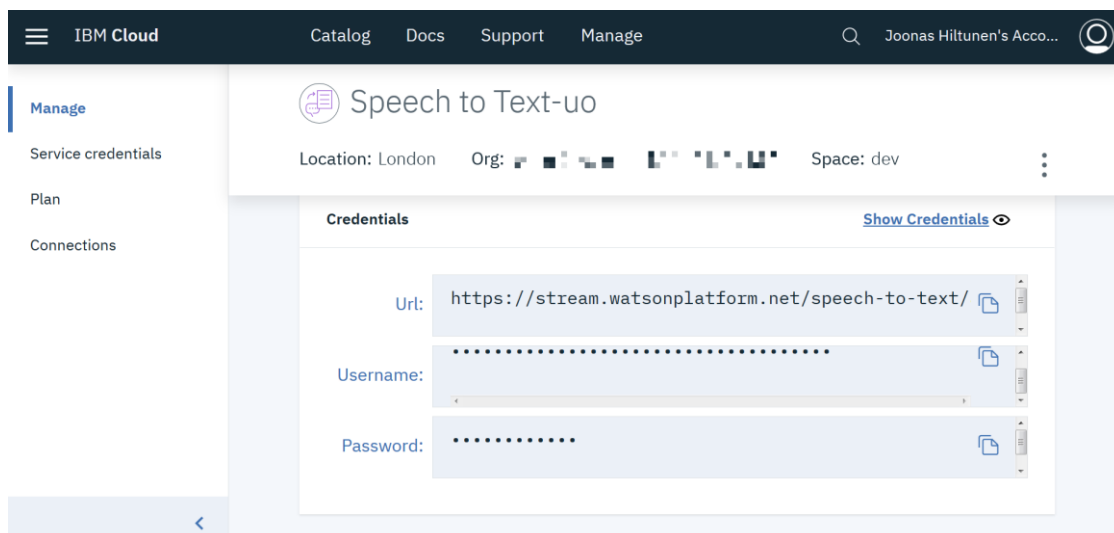
Watsonin voice-to-text -ominaisuutta varten tarvitaan Watson speech-to-text tunnus ja salasana. Tunnuksen voi tehdä osoitteessa <https://www.ibm.com/watson/>. Käyttäjän on syytä huomata, että Voice-to-text tunnus on eri asia, kuin Watson profiilisi tunnus. Lisäksi Watsonin voice-to-text -palvelu on maksullinen, mutta ensimmäiset 300 minuuttia puhetta on ilmainen kaikille käyttäjille. Käytimme juuri tätä ilmaista ominaisuutta opinnäytetyötä kehittäessämme.

Ensin on käyttäjän luotava itselleen IBM Cloud -tunnus. Kun tunnus on luotu, on käyttäjän mentävä dashboardiin. Dashboardissa voi luoda tunnukselleen palveluja. Kuten kuvassa 47 näkyy, oikeassa yläkulmassa on "Create resource"-painike. Tätä nappia painamalla voi käyttäjä luoda itselleen palvelun. Käyttäjän on valittava Speech to text.



Kuva 47. IBM Cloud Dashboard.

Kun Watson tunnus on tehty ja speech-to-text palvelu luotu, voi käyttäjä napata käyttöönsä speech-to-text käyttäjän ja salasanan. Klikkaamalla dashboardissa palvelua pääsee käsiksi käyttäjätietoihin kuvan 48 osoittamalla tavalla.



Kuva 48. Speech to Text -tunnukset IBM Cloudissa.

10.4 BIM Track -API

BIM Track -API:n toiminnan toteuttamiseen konsultoimme Peltosen Ossia, joka teki samoihin aikoihin opinnäytetyötä BIM konseptista [15, Tietomallinnuksen vuorovaikutusalustan ohjelmallinen käsittely]. API:lla voi tehdä koodissa webrequesteja, jotka ajavat saman asian kuin BIM Trackin oma nettisivu. Webrequestien onnistumista varten on käyttäjän syötettävä BIM Track tunniste. Tunniste syötetään API:n yläkulmassa olevaan tekstilaatikkoon. Koodissa vastaava tunniste tapahtuu HTTP-käskyssä “bearer” autentikaation mukaan.

API:ssa on esitetty API:n jokaisesta HTTP-metodista esimerkki Json-, tai XML-muodoissa. Lisäksi käyttäjä voi syöttää tekstilaatikoihin tietoa kuvan 49 osoittamalla tavalla.

BIM Track API

Hub Show/Hide | List Operations | Expand Operations

GET /v2/hubs Get hubs

GET /v2/hubs/{hubId} Get hub

Implementation Notes
Lists the details about the specified hub.

Response Class (Status 200)
OK

Model | Example Value

```
{
  "Id": 0,
  "Name": "string",
  "Author": {
    "Id": 0,
    "Email": "string",
    "FirstName": "string",
    "LastName": "string"
  },
  "Image": {
```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
hubId	<input type="text" value="(required)"/>	Hub id	path	integer

GET /v2/hubs/{hubId}/projecttemplates Get project templates

POST /v2/hubs/{hubId}/projecttemplates Create project template

DELETE /v2/hubs/{hubId}/projecttemplates/{projectTemplateId} Delete project template

GET /v2/hubs/{hubId}/users Get hub users

POST /v2/hubs/{hubId}/users Create hub user

Kuva 49. Bimtrack-API.

Aloitusvaiheessa käytimme BIM Track -API:n ykkösversiota, mutta BIM Trackin kehittäjät päivittivät API:nsa kakkosversioon ja muutimme koodiimme vastaavaan API:n kakkosversiota. BIM Track -API mahdollistaa POST, PUT, PATCH, DELETE toiminnot. Opinnäytetyössämme emme tarvinneet käyttää muita metodeja, kuin POST, GET ja PATCH. Alla olevassa taulukossa (Taulukko 2) on selitettyä HTTP-metodit suomeksi. [16, Using HTTP Methods for RESTful Services].

Taulukko 2. Yleisimmät HTTP-metodit.

Metodi	Suomeksi	Toiminta
POST	Luonti	Luo uutta tietoa vastaanottavalle palvelimelle
GET	Luku	Päivittää tai korvaa resurssia palvelimella
PUT	Päivitys/korvaus	Korvaa tietoa olemassa olevasta resurssista
PATCH	Päivitys/muokkaus	Lisää tietoa olemassa olevaan resurssiin
DELETE	Poisto	Poistaa resurssin kokonaan

10.4.1 Unityn ja API:n välinen kommunikointi

Unityn ja BIM Track -API:n keskustelua varten teimme API:n Json-esimerkkien pohjalta C# -luokkia. Luokkia käytettiin laittamalla niihin, tai lukemalla niistä tietoa tarvittaessa. Luokat ovat helppoja muuttaa Json muotoon ja Json muodosta on helppo muuttaa C# -luokkamuotoon.

Json lukua ja kirjoitusta varten käytimme Unityn oletuskirjaston sijasta Newtonsoftin Json-kirjastoa, jossa on enemmän ominaisuuksia ja oli helpommin yhteensopiva BIM Track -API:n kanssa [17, Json.NET]. Esimerkiksi Unityn oletuskirjaston päivämäärät eivät muuntuneet hyvin BIM Track -API:n ymmärtämään muotoon. Lisäksi Newtonsoftin Json-kirjastolla pystyi hyvin sivuuttamaan tyhjää tietoa, joita oli C#-luokissa. Esimerkiksi, jos käyttäjä ei täytä kaikkia kenttiä kehitystehtävän luomisvaiheessa, kenttien tyhjät ja tarpeettomat tiedot eivät lähde HTTP-kutsussa.

10.4.2 GET-metodin käyttö

Ensimmäisenä asiana meidän oli tiedettävä hubimme ja projektimme tunnusluvut. API:n "gethubs" GET-metodi ajetaan päävalikkoscenessä, josta saadaan tallennettua hubin tunnusluku. Tämän jälkeen ajetaan projektien gettaus, josta saadaan projektien nimet. Nimet syötetään päävalikon alasvetovalikkoon.

Projektin ja hubin GET-metodia ei tapahdu muissa kohdissa ohjelmaa, koska hubin ja projektien tiedot luetaan päävalikkoscenessä luodusta XML-tiedostosta.

Ohjelman kehitystehtävän historia työkalussa käytetään GET-metodia kehitystehtävien näyttämiseen. Ensin käytetään getissues implementaatiota, jolla saadaan alasvetovalikkoon kehitystehtävien nimet. Kun nimet ovat näkyvissä ja käyttäjä painaa valitse-nappia, ohjelma käyttää issueid-implementaatiota yksittäisen kehitystehtävän tietojen saamiseen.

10.4.3 POST-metodin käyttö

POST-metodia käytimme luomaan uusia kehitystehtäviä. Kehitystehtävän lähetys vaatii tiedot hubin ja projektin tunnisteluvuista. Nämä tiedot ohjelma lukee päävalikossa luodusta XML-tiedostosta. Kehitystehtävän luomiseen tarvitaan lisätietoa siitä, että mitä tietoja postataan. Tähän sisältyy esimerkiksi tietoa kehitystehtävän otsikosta, kuvauksesta, tai kriittisyydestä. Kehitystehtävän voi tosin luoda pelkällä otsikolla.

Ohjelmassamme kehitystehtävän luomisen yhteydessä voidaan lähettää tieto ongelman koordinaateista kuvan kera. Koodinpuolella kehitystehtävän luonti ja kuvan lisäys ovat erillisiä operaatioita. Koordinaattitiedot ja kuva postataan olemassa olevaan kehitystehtävään kehitystehtävän id:hen viitaten. Kuva luetaan levyltä ja muutetaan binäärimuotoon, jonka jälkeen ajetaan Unityn Webrequestin POST-metodia.

10.4.4 PATCH-metodin käyttö

Kehitystehtävän poistoa varten käytimme API:n Archive issue -implementaatiota. Tämä siirtää kehitystehtävän BIM Trackin kehitystehtävä va-

rastoon. Varastosta ei voida enää lukea suoraan API:n GetIssue-implemmentatiolla kehitystehtäviä. Varasto sisältää pikemminkin läjän kehitystehtäviä, jotka odottavat poistoa. Emme lisänneet ohjelmaamme lopullista poistoa, vaan käyttäjän on erikseen poistettava kehitystehtävä BIM Track -sivuilta.

PATCH-metodia ei Unityssä esiintynyt staattisena metodina, vaan jouduimme hieman kustomoimaan UnityWebRequestin lähetysominaisuuksia. Ensin määrittelimme UnityWebRequestin olevan staattista PUT-metodia, jonka jälkeen määrittelimme heti seuraavalla rivillä metodin olevan PATCH-muotoa. PATCH-metodin ajaminen ei muulla tavalla eronnut muista HTTP-metodeista. Toisin sanoen, kehitystehtävän poistamiseen tarvittiin hubin tunnisteluku, projektin tunnisteluku, kehitystehtävän tunnisteluku, sekä BIM Track -tokeni.

11 Pohdinta

11.1 Vuorovaikutus

Ennen meitä Karelia AMK:lle VR-sovelluksia ei ollut tehnyt kukaan, joten pääsimme toimimaan eräänlaisina suunnan näyttäjinä, mitä jo varsin nopealla opiskelemisella voidaan saada aikaan virtuaalitodellisuus maailmassa. Laitteistokin oli varsin uutta. Laitteisto oli hankittu vain joitain kuukausia ennen kuin aloitimme opinnäytetyön tekemisen. Olemme myös olleet mukana asentellessa laitteistoa eri tiloihin Karelia AMK:n Wärtsilä-kampuksella ja näyttäneet, miten laitteisto kannattaa asentaa.

Opinnäytetyömme edetessä esittelimme työtämme useaan otteeseen Karelia AMK:n henkilökunnalle ja muille asiasta kiinnostuneille vieraille. Työmme oli myös näytillä Karelian ura- ja rekrypäivänä, jossa sitä pääsi tapahtumaan osallistuneet opiskelijat, yritysten yhteyshenkilöt ja muut paikalla olleet testaamaan. Yleisesti palaute esitellyistä oli positiivista ja monille työmme on ollut ensimmäi-

nen VR-kokemus. Negatiivista palautetta on tullut oikeistaan vain käyttöliittymän vaikea selkoisuudesta, tätä emme ruvenneet enää sen kummemmin korjaamaan, sillä esittelytilaisuudet järjestettiin loppupuolella opinnäytetyö urakkaamme. Kaikki palaute oli erittäin tervetullutta ja sitä toivottavasti hyödynnetään tulevissa projekteissa.

Rakentamisen digitalisaation -työpaja on kurssi, jonka ohjaajamme Petri Laitinen perusti 2018 syyskaudella. Sen luonti lähti ideasta, että kun me saamme tämän opinnäytetyön valmiiksi, alamme olla viittä vaille valmistuneita. Tarvittiin opiskelijoita ikään kuin jatkamaan VR:n kanssa touhuamista Karelia Amk:ssa. Kurssille otettiin sekalainen sakki toisen vuoden opiskelijoita kone- ja tuotantotekniikan-, rakennustekniikan- ja talotekniikan -tutkinnoista. Opiskelijoille opetimme Unity-pelimoottorin käyttöä, kuinka tehdä yksinkertainen VR-sovellus. Tätä kirjoittaessa on vielä epäselvää miten VR-aiheiset opiskelut jatkuvat Karelia AMK:ssa, mutta ainakin kurssille osallistuneissa on ollut VR:sta kiinnostuneita.

11.2 Kehitysideoita

Opinnäytetyön tekemisen aikana tuli runsaasti kehitysideoita, jotka olisivat olleet enemmän ja vähemmän vaikeasti toteutettavia. Suurinta osaa näistä ei lähdetty tekemään ajankäytön säästämiseksi, johonkin oli vain vedettävä viiva ja priorisoitava jonkinlainen versio valmiiksi.

Työssämme ohjaimina toimivat käsien 3D-mallit, jotka tehtiin reagoimaan liikuttaviin objekteihin. Perusasetuksilla SteamVR luo 3D-mallit käytettävistä ohjaimista virtuaalimaailmaan. Ohjaimet ovat yksityiskohtaiset ja täsmälleen saman kokoiset kuin ohjaimet, joita käyttäjä käsissään pitää. Esittelyjen yhteydessä selostimme useaan otteeseen ohjelmamme ominaisuuksia, nämä selostukset olisivat varmasti olleet vaivattomampia, jos ohjain mitä pidetään kädessä,

olisi vastannut ohjainta minkä he näkevät virtuaalimaailmassa. Sen lisäksi meidän ei olisi tarvinnut luoda käsi 3D-malleja ollenkaan.

Kynätyökalu jäi ehkä hieman prototyypivaiheeseen, sillä pystyi piirtämään, mutta piirroksia ei pystynyt poistamaan. Jos töherrykset halusi pois, piti ohjelma aina käynnistää uudestaan. Myös ominaisuuksia voisi lisätä kynätyökaluun, kaiken maailman erityyppisiä kyniä tai vaikkapa leimasin.

Kartta on työkalu idea, jota tuli mietittyä, kun oltiin loppupuolella opinnäytetyötä. Se voisi vaikkapa olla kuva ylhäältäpäin, jossa näkyy koko koulumalli-Scene ja oma sijainti. Siinä voisi myös olla vaihtoehtoisia karttatileja, joista yksi voisi olla erittäin pelkistetty ja toinen voisi näyttää vaikkapa pohjapiirustuksen. Teoria tasolla kartan saisi tehtyä, kun lisää Unityssa uuden kameran, joka katsoo kohtisuoraan alaspäin ja käyttää kameran render-to-texture ominaisuutta, joka tekee kameran kuvasta päivittyvän tekstuurin, joka voidaan liittää haluttuun 3D-malliin.

Vuorovaikuttamista ympäristön kanssa olisi voitu lisätä. Oltaisiin voitu, vaikka luoda kouluun valonkatkaisijoilla toimivat valot. Myös olisi voitu tehdä avattaviksi, sillä ohjelmassamme ne vain olivat staattisia objekteja, joiden läpi pystyi teleporttaamaan.

11.3 Loppusanat

Aloittaessamme opinnäytetyön teon meillä ei ollut minkäänlaista kokemusta virtuaalidellisuus sovellusten teosta eikä projektipankki palvelujen käytöstä. Opinnäytetyön aikana meille kehittyi ymmärrystä miksi projektipankki palvelut ovat hyödyllisiä ja miten VR-sovelluskehitystä toteutetaan Unity-pelimoottorilla. Kaiken kaikkiaan opinnäytetyöprosessi oli meille hyvä oppimismatka VR-maailmaan ja voi vaikuttaa meidän tulevaisuuden työpaikkoihin.

Opinnäytetyön aikana toteutettiin konkreettinen VR-ohjelma, jossa on Kemien Koulun 3D-malli ympäristöineen, VR-toiminallisuus monine työkaluineen, proto-

tyyppi asteinen äänestä tekstiksi tuki ja BIM Track -API ominaisuuksia. Kokonaisuutena VR-ohjelmastamme tuli ihan hyvin toimiva kokonaisuus ja sitä voisi lähteä kehittämään monella tapaa, vaikka ihan kaupalliseksi tuotteeksi asti.

Lähteet

1. Van de Kerckhove, E. HTC Vive Tutorial for Unity. 2016. Raywenderlich.com [Viitattu 26.11.2018] Saatavissa: <https://www.raywenderlich.com/149239/htc-vive-tutorial-unity>
2. Vive, What are system requirements [viitattu 13.11.2018.] Saatavissa: https://www.vive.com/us/support/vive/category_howto/what-are-the-system-requirements.html
3. Textures.com. Terms and Conditions. 2018. Textures.com [Viitattu 4.12.2018] Saatavissa: <https://www.textures.com/terms-of-use.html>
4. Paulo, J. f.a.q. 2018. 3D Textures [Viitattu 4.12.2018] Saatavissa: <https://3dtextures.me/about/>
5. Textures, Buttons0150 [21.11.2018] Saatavissa: <https://www.textures.com/download/buttons0150/46667>
6. Paulo, J. Pavement Brick 001. 2018. 3D TEXTURES [Viitattu 26.11.2018] Saatavissa: <https://3dtextures.me/2018/01/19/pavement-brick-001/>
7. Unity. Types of light [Viitattu 5.12.2018] Saatavissa: <https://docs.unity3d.com/Manual/Lighting.html>
8. Unity. Performance Optimization. Unity Documentation [viitattu 13.11.2018.] Saatavissa: <https://unity3d.com/learn/tutorials/topics/performance-optimization>
9. Unity. Scenes. Unity Documentation 1.8.2017. [viitattu 12.11.2018.] Saatavissa: <https://docs.unity3d.com/Manual/CreatingScenes.html>
10. Biagioli, A. HTC Vive Teleportation System with Arc Pointer. 2017. GitHub [Viitattu 10.12.2018] Saatavissa: <https://github.com/FlafLa2/Vive-Teleporter>
11. Unity. UI. Unity Documentation.[Viitattu 5.12.2018]. Saatavissa: <https://docs.unity3d.com/Manual/UISystem.html>
12. VRTK team. VRTK – Virtual Reality Toolkit. 2018. [Vittattu 10.12.2018] Saatavissa: <https://vrtoolkit.readme.io/>
13. Nimimerkillä "illa3d". Unity.com Foorumilla. [Viitattu 5.12.2018]. Saatavissa: <https://answers.unity.com/questions/932607/putting-a-multiline-inputfield-in-a-scroll-rect.html>
14. Quirk, V. A Brief History of BIM. Arch Daily. 2012. [Viitattu 13.8.2018.] Saatavissa: <https://www.archdaily.com/302490/a-brief-history-of-bim>
15. Peltonen, O. Tietomallinnuksen vuorovaikutusalustan ohjelmallinen käsittely. 2018. Theseus. [Viitattu 3.12.2018] Saatavissa: <http://urn.fi/URN:NBN:fi:amk-2018112918973>
16. Using HTTP Methods for RESTful Services. 2018. RestApiTutorial [Viitattu 28.11.2018] Saatavissa: <https://www.restapitutorial.com/lessons/httpmethods.html>
17. Newtonsoft. Json.NET. 2018. [Viitattu 3.12.2018] Saatavissa: <https://www.newtonsoft.com/json>