

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2019

Ville Vainio, Verner Villikka

# VIRTUAALITODELLISUUDEN HYÖDYNTÄMINEN TEOLLISUUDESSA

**TURKU AMK**   
TURKU UNIVERSITY OF  
APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja viestintätekniikka

2019 | 43 sivua

Ville Vainio, Verner Villikka

# VIRTUAALITODELLISUUDEN HYÖDYNTÄMINEN TEOLLISUUDESSA

Virtuaalitodellisuutta on viime aikoina alettu hyödyntämään teollisuudessa sen eri osa-alueilla. Virtuaalitodellisuusteknologian hyödyntäminen on kuitenkin vasta alkutekijöissään. Opinnäytetyön tavoitteena oli tutkia, kuinka virtuaalitodellisuutta hyödynnetään eri aloilla nyt, ja kuinka sen käyttöönottoa voitaisiin helpottaa tulevaisuudessa.

Nykyisiä ja tulevia virtuaalitodellisuuden hyödyntämistapoja tutkittiin analysoimalla toimeksiantajan, 360Mediatalon, asiakkuussuhteita ja palveluita kommunikoimalla asiakkaiden ja 360Mediatalon työntekijöiden kanssa. Virtuaalitodellisuuden käyttöönoton helpottamista edistettiin 360Editor-projektilla. 360Editorilla voi luoda ja jakaa virtuaalitodellisuussisältöä ilman tietoteknistä osaamista.

Virtuaalitodellisuussovellus kehitettiin Unity-pelimootorilla C#- ja JavaScript-ohjelmointikielillä Oculus GO -virtuaalitodellisuuslaseille. Sisällöntuottamiskalut toimivat verkkoselaimella.

360Editor on edelleen kehityksessä, mutta sillä on jo päivittäisiä käyttäjiä teollisuudessa. Projekti on jo tuonut uudenlaisia keinoja hyödyntää virtuaalitodellisuutta.

## ASIASANAT:

Virtuaalitodellisuus, ohjelmointi, Unity, C#, JavaScript

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2019 | 43 pages

Ville Vainio, Verner Villikka

## VIRTUAL REALITY IN INDUSTRIAL USE

360Mediatolo is a company that offers its clients the possibility to showcase and market their products in a virtual environment. It also offers a platform to create and share virtual reality (VR) content. This thesis addresses the 360Mediatolo existing and upcoming VR services and the current state of VR in industrial use. The primary source of this thesis is a project called 360Editor which is a content creation and sharing platform for VR. The aim of the project is to eventually make VR a common commodity.

The purpose of this thesis was to explore different possible use cases for VR in the industry. Using VR technology as a tool is still in its infancy. Therefore, this project aims to create tools to make adopting VR technology easier.

The current and upcoming usages of VR technology were mainly explored through the client cases of 360Mediatolo. Also, external sources, such as other studies were used. The VR app was developed with C# and JavaScript programming languages. It was created for Oculus GO VR glasses using the Unity Engine. Content creation tools are controlled through a separate website.

360Editor is still in development but it already has every day users in the industry. The project has already brought new use cases for VR.

### KEYWORDS:

Virtual reality, programming, Unity, C#, JavaScript

# SISÄLTÖ

<b>KÄYTETYT LYHTENTEET JA SANASTO</b>	<b>5</b>
<b>1 JOHDANTO</b>	<b>1</b>
<b>2 VIRTUAALITODELLISUUS TEOLLISUUDESSA</b>	<b>3</b>
2.1 Jatkettu todellisuus	3
2.2 360Mediatalon VR-palvelut	4
2.3 Muita VR-käyttömahdollisuuksia	5
2.4 Virtuaalitodellisuuden tulevaisuus	6
<b>3 360EDITOR-SOVELLUKSEN TOIMINTA</b>	<b>8</b>
3.1 Show'n luominen	9
3.2 Muita ominaisuuksia	12
<b>4 TEKNINEN TOTEUTUS</b>	<b>13</b>
4.1 Google Cloud Platform osana ympäristöä	13
4.2 VR-ympäristötyökalujen luominen verkkoselaimelle	13
4.2.1 Esivaatimukset	13
4.2.2 Palvelimen toiminta	14
4.3 Interaktiivisen VR-ympäristön ohjelmointi	18
4.3.1 Esivaatimukset	18
4.3.2 360-kuvan esittäminen Unityllä ja kokoaminen laseihin	19
4.3.3 Ohjaaminen ja interaktiivisuus VR-maailmassa	24
<b>5 LOPUKSI</b>	<b>30</b>
<b>LÄHTEET</b>	<b>31</b>

## KÄYTETYT LYHTENTEET JA SANASTO

API	ohjelmointirajapinta (Application Programming Interface)
Augmented reality	lisätty todellisuus (AR), oikeaan maailmaan lisättyjä VR-elementtejä
C#	Microsoftin kehittämä ohjelmointikieli
CSS	tyypillisesti verkkosivuille luotu tyyliohje (Cascading Style Sheets)
EJS	hahmontaa palvelimelta asiakasohjelmalle näkymän (Embedded JavaScript)
Express	Node.js palvelinsovelluskehys (framework)
Extended reality	yleisnimitys eri keinotodellisuuksille
Hosting	palvelun ylläpitäminen
HTML	verkkosivuille luotu kuvauskieli (Hypertext Markup Language)
IDE	ohjelmointiympäristö (Integrated Development Environment)
JavaScript	internetympäristössä käytettävä ohjelmointikieli
JSON	tiedonvälitykseen luotu tiedostomuoto (JavaScript Object Notation)
Mixed reality	yhdistetty todellisuus (MR), virtuaalitodellisuuden ja lisätyn todellisuuden yhdistelmä
Moduuli	itsenäinen API-kirjasto
MySQL	Oracle Corporationin omistama relaatiotietokantaohjelmisto
Nadiiri	360-kuva ja -videotuotannossa käytetty termi suoraan käyttäjän jalkojen alla olevalle, kameran jalan peittävälle objektille. Nimen alkuperä juontaa tähtitieteestä katsojan jalkojen alapuolella olevasta pisteestä.
Node.js	runtime-ympäristö JavaScriptille
Oculus Go	Facebookin kehittämä langaton VR-laite
Runtime-ympäristö	suorittaa koodin
SDK	sovelluksen luomistyökalut (software development kit)
Unity	monialustainen pelimoottori
VR	virtuaalitodellisuus (virtual reality)

XHR

pääosin tiedostojen lähetykseen käytettävä API (XMLHttpRequest)

# 1 JOHDANTO

Virtuaalitodellisuutta on pitkään hyödynnetty pääsääntöisesti vihteessä, mutta myös teollisuus on alkanut viime aikoina hyödyntämään VR-teknologiaa (360Mediatalo 2018). Virtuaalitodellisuus tarjoaa täysin uudenlaisia lähestymistapoja erilaisiin teollisuuden ongelmiin, ja sillä voi tehostaa jo käytössä olevia menetelmiä.

Monet yritykset eivät käytä virtuaalitodellisuutta, sillä sen käyttöönoton pelätään vievän aikaa ja rahaa. Myös monien ihmisten tietoisuus VR-teknologiasta, ja sen mahdollisuuksista, on vähäistä.

Tässä opinnäytetyössä käsitellään erilaisia mahdollisuuksia hyödyntää virtuaalitodellisuutta teollisuudessa. Opinnäytetyössä kerrotaan toimeksiantajan asiakkuuksista ja erilaisista tutkimuksista kerättyä tietoa hyödyntäen, kuinka virtuaalitodellisuutta käytetään teollisuudessa tänä päivänä ja kuinka sitä voisi käyttää tulevaisuudessa.

360Mediatalo tarjoaa asiakkailleen mahdollisuuden esitellä toimintaansa ja markkinoida tuotteitaan virtuaaliympäristössä (360Mediatalo 2018). Opinnäytetyössä kerrotaan myös 360Mediatalon kehittämästä 360Editor-alustasta ja sen kehittämisestä. Alustan käyttäjä voi helposti luoda ja jakaa virtuaalitodellisuussisältöä. 360Editorin päämääränä on tehdä virtuaalitodellisuuden käyttöönotosta helpompaa ja näin myös yleistää virtuaalitodellisuutta teollisuudessa.

360Editorin sisällön luominen ja jakaminen tapahtuu verkkosivujen kautta ([360editor.fi](https://360editor.fi)). Sisällön katselu toimii parhaiten Oculus GO-virtuaalitodellisuuslaseilla, jotka valittiin kehitystyön ensisijaiseksi laitteeksi niiden langattomuuden vuoksi. Alustan sovelluspuoli on kehitetty C#-ohjelmointikielellä Unity-pelimoottorilla, ja verkkosivut on kehitetty JavaScript-ohjelmointikielellä. Opinnäytetyön kirjoittamisen hetkellä alusta on vielä kehitysvaiheessa, mutta kuitenkin jo käytössä joillakin yrityksillä.

360Editorin kehityksessä Villikka vastaa verkkosivujen kehityksestä ja kertoo siitä luvussa 4.2. Vainio puolestaan vastaa VR-laseille kehityksestä ja kertoo siitä luvussa 4.3. Sekä projektissa että opinnäytetyössä muu on tehty yhteistyönä.

Aiheena virtuaalitodellisuutta on käsitelty opinnäytetöissä useasti ennenkin. Yleisesti VR-teknologian hyödyntämismahdollisuuksia on kuitenkin pohdittu vain pinnallisesti. Myös virtuaalitodellisuuden sisällöntuottamistyökalut ovat aiheena uusi.

Virtuaalitekniologia kehittyy kovaa vauhtia. Uusia, entistä parempia laitteita julkaistaan tasaiseen tahtiin. Tämän vuoksi hinnat laskevat, teknologia yleistyy ja kysyntä kasvaa nopeasti. Esimerkiksi videopelien jakelualusta Steam on kaksinkertaistanut VR-käyttäjien määrän vuosina 2017–2018 (Lang 2018).



## 2 VIRTUAALITODELLISUUS TEOLLISUUDESSA

### 2.1 Jatkettu todellisuus

Englannin kielinen termi *Extended reality* viittaa yleisesti kaikkeen todellisuutta imitoivaan teknologiaan. Pääasiassa termillä tarkoitetaan siis virtuaalitodellisuutta (VR), keinoitekoista maailmaa, johon käyttäjä uppoaa sekä lisättyä todellisuutta (AR) eli todellisuuteen lisättyjä virtuaalisia elementtejä että niiden yhdistelmää, yhdistettyä todellisuutta (MR). (Scribani 2019.)

Tässä opinnäytetyössä keskitytään pääasiassa yleisimpään jatkettun todellisuuden osa-alueeseen, virtuaalitodellisuuteen (Scribani 2019). Käyttäjä siirtyy virtuaalitodellisuuteen laittamalla päähänsä VR-lasit, jotka näyttävät käyttäjälle tietokoneella rakennetun maailman. VR-lasit seuraavat käyttäjän pään liikettä, jonka avulla käyttäjä voi katsoa ympärilleen luonnollisesti. Useimpien virtuaalilasien mukana on myös ohjaimet, joilla voidaan imitoida käsien liikettä. Jotkin VR-lasit seuraavat jopa käyttäjän fyysistä sijaintia.

Virtuaalitodellisuuden määritelmä on erittäin suurpiirteinen. Esimerkiksi SEGA kehitti vuonna 1994 virtuaalilaseiksi kutsuttua SEGA VR:ää. Nykystandardien mukaan harva varmaan luokittelisi tätä teknologiaa virtuaalitodellisuudeksi. SEGA VR oli todellisuudessa kypärä, jossa oli silmien edessä kaksi LCD-paneelia. Vuonna 2012 Oculus järjesti Kickstarter-kampanjan kuluttajakäyttöön tarkoitetuille Oculus Rift -virtuaalitodellisuuslasille. Facebook osti Oculus VR -yhtiön kahdella miljardilla dollarilla, jonka jälkeen muutkin suuryritykset, kuten Sony, Google ja Samsung, ryhtyivät kilpailemaan virtuaalitodellisuusmarkkinoilla. (Barnard 2018.)

Lisätystä todellisuudesta puhuttaessa voidaan viitata esimerkiksi AR-laseihin, mutta myös nykyiset älypuhelimet edustavat AR-teknologiaa. Muun muassa vuonna 2016 julkaistu, äärimmäisen suosion tavoittanut Pokemon GO -mobiilipeli perustui täysin lisättyyn todellisuuteen. Pelissä hahmot näkyvät puhelimen näytöllä ikään kuin ne olisivat oikeassa maailmassa. Pokemon GO ja muut AR-mobiilisovellukset hyödyntävät älylaitteen kameraa ja piirtävät AR-elementit kamerakuvan päälle. Älylaitteiden AR-mahdollisuuksia hyödynnetään myös muilla teollisuuden osa-alueilla. Esimerkiksi IKEA Place -sovelluksella käyttäjä voi kokeilla, miltä erilaiset sisustusvaihtoehdot näyttävät halutussa tilassa. Monet suuryritykset kehittävät myös omia AR-lasejaan. (Bonsor & Chandler 2018.)

Vaikka markkinoilla onkin jo laseja yhdistetyn todellisuuden nimellä, ne eivät edusta yhdistettyä todellisuutta nimen todellisessa merkityksessä. MR-laseissa olisi tarkoitus su-lavasti yhdistää AR- ja VR-teknologiat. MR-laseissa olisi edessä kamera, jolla lopulta olisi tarkoitus tuoda oikean maailman elementtejä VR-maailmaan ja toisin päin. Virtuaali-maailmasta siirtyminen todellisuuteen ja takaisin olisi helppoa ja vaivatonta. Tällä het-kellä ainoa tunnettu MR-lasimalli on Windows Mixed Reality headset, joista useilla val-mistajilla on hieman erilaiset mallit. Windows Mixed Reality headset ei kuitenkaan ni-mensä mukaisesti tue AR:ää lähes lainkaan. Lasien edessä on kamera, mutta sillä ei ole lisätyn todellisuuden funktioita. Kameraa käytetään lähinnä uudenlaiseen inside-out kä-sien liikkeentunnistukseen, jonka avulla voidaan luopua ulkoisista liikkeentunnistimista. Toisin kuin virtuaalilasit, Windows mixed reality headsetin kamera mahdollistaa myös pelialueen rajojen piirtämisen pelimaailmaan. Näin välttyään ylimääräisiltä seinin törmäi-lyiltä. Näiden ominaisuuksien vuoksi Windows on nimennyt lasit yhdistetyn todellisuuden laseiksi. (Willings & Tillman 2018.)

## 2.2 360Mediatalon VR-palvelut

Virtuaalitodellisuus on loistava koulutusympäristö. Sen vaikutusta oppimiseen on tutkittu jo jonkin verran. Esimerkiksi artikkelissa Virtual memory palaces: immersion aids recall kerrotaan koehenkilöiden muistaneen asioita 8,8 % paremmin, jos ne oli opittu virtuaali-todellisuusympäristössä, verrattuna tavalliseen tietokoneympäristöön (Krokos, Plaisant & Varshney 2018). VR-teknologia soveltuu siis hyvin esimerkiksi uusien työntekijöiden kouluttamiseen tai erilaisten laitteiden käytönopastukseen. Työntekijöiden perehdytys VR-teknologialla on usein myös perinteisiä koulutusmetodeja edullisempi, sillä VR-kou-lutus ei välttämättä vaadi kouluttajaa tai muita laitteita VR-lasien lisäksi.

VR-kokemus on immersivisempi kuin perinteinen kuvien katselu. Esimerkiksi tiloista saa huomattavasti elävämmän ja tarkemman käsityksen VR-ympäristössä. Tämän vuoksi esimerkiksi lomakohteiden esittelyä voi tehostaa virtuaalisesti. Uusimmissa VR-laseissa on jo helppokäyttöinen käyttöliittymä, jolla pystyy tekemään ison osan perinteisesti tieto-koneella tehtävistä asioista. Lomakohteitakin voi siis jo varata ja selata VR-laseilla ilman ylimääräistä vaivaa. VR-lasien käyttöliittymässä on aina suoraa mainontaa laselle sopi-vista sovelluksista ja palveluista. Tämän ansiosta ihmisiä päätyy tutkimaan erilaisia koh-teita jo pelkästään uteliaisuuttaankin.

360Mediatalo tarjoaa asiakkailleen räätälöityjä VR-palveluita, jotka koostuvat pääasiassa koulutuksista ja tuote- tai tilaesittelyistä. Myös erilaisten tapahtumien VR-taltioinnin suosio on kasvussa. Esimerkiksi konserttien tai urheilutapahtumien uudelleeneläminen virtuaalisesti on jo itsessäänkin mieleenpainuva kokemus. 360Mediatalon aina ajan tasalla olevan kaluston ansiosta, vain asiakkaan mielikuvitus on palveluiden rajana.

Asiakkaiden on mahdollista ostaa itselleen lisenssi, jolla pääsee käyttämään 360Editoria. 360Editor on alusta, jolla käyttäjä voi helposti, ilman ohjelmointitaitoja, luoda omia VR-kokemuksia, jotka näkyvät suoraan VR-laseilla. Opinnäytetyön luvut 3 ja 4 keskittyvät tähän sovellukseen.

Vaikka virtuaalitodellisuutta on eri muodoissa ollut markkinoilla jo pitkään, sen yleistymisen on alkanut vasta viime vuosina. Virtuaalitodellisuusosaamista on tästä syystä vielä suhteellisen vähän. Yhtenä palvelunaan 360Mediatalo tarjoaakin virtuaalitodellisuuden eri osien tuotantokoulutusta. Tällaista on esimerkiksi 360kuva ja -videotuotanto.

### 2.3 Muita VR-käyttömahdollisuuksia

Virtuaalitodellisuudesta puhuttaessa monille ensimmäinen ajatus on videopelit, eikä syyttä. Uusia VR-pelejä julkaistaan jatkuvasti (Hayden 2018), toinen toistaan realistisempia ja näyttävämpiä. VR-pelejä varten kehitetään jatkuvasti useita erilaisia lisälaitteita, kuten esimerkiksi suuntaukseton juoksumatto Virtuix Omni (Virtuix 2019) ja askeleen tunnistavat Cybershoes-kengät (Cybershoes 2019). Myös VR-lasien valmistajat julkaisevat ahkerasti uusia malleja tuotteistaan. Uudemmissa malleissa on yleensä esimerkiksi edellistä parempi kuvanlaatu, tarkempi ohjaus, paremmin optimoidut säätömahdollisuudet sekä korkeatasoisempi äänenlaatu. Myös johdoista pyritään pääsemään pikkuhiljaa eroon ominaisuuksista tinkimättä.

Armeijat ympäri maailmaa ovat käyttäneet virtuaalitodellisuutta jo pitkään. Virtuaalitodellisuus mahdollistaa sotilaiden harjoittelevan tilanteita, joiden harjoittaminen muuten olisi erittäin vaikeaa (Fade 2018). Sotatilanteiden harjoittelu on myös erittäin kallista ja virtuaalitodellisuus tarjoaa apua myös tähän. Virtuaalitodellisuutta voisi tulevaisuudessa myös hyödyntää armeijalaitteiden operoinnissa.

Vastapainona armeijalle, myös lääketiede hyödyntää virtuaalitodellisuutta (Medium 2018). Kuten lähes jokaisessa virtuaalitodellisuutta hyödyntävässä alassa, myös lääketieteessä käytetään virtuaalitodellisuutta koulutusympäristönä. Lääketiede on kuitenkin

vienyt virtuaalitodellisuuden hyödyntämisen vielä pidemmälle, ja käyttää sitä myös erilaisissa toimenpiteissä, jopa leikkauksissa. Virtuaalitodellisuutta käytetään myös hoitokeinona, erityisesti mielenterveyspotilailla (Temming 2018). Myös vuodepotilaiden viihtyvyyttä ja aktiivisuutta on pyritty edistämään virtuaalitodellisuuden avulla (Barchester 2017).

VR-alustoille on kehitetty jo työkaluja, joilla voi helpottaa tuotekehitysprosessia (Evans 2019). Insinöörit ympäri maailman käyttävät virtuaalitodellisuutta tuotteiden testauksessa ja jopa kehityksessä. Esimerkiksi autoteollisuudessa VR-tekniikalla tutkitaan prototyyppijä, sillä osat on helppo erittää toisistaan ja koko prototyypin voi halutessaan purkaa osiin. Tämä säästää sekä aikaa että rahaa (Steed 2017).

Urheilijoidenkin suorituksia voi mahdollisesti tehostaa virtuaalisella oppimisella (Baptiste 2018). Esimerkiksi VR-laseilla opitut liikkeet jäävät lihasmuistiin. Virtuaalilaitteet seuraavat ja tallentavat käyttäjän toimintaa, jonka ansiosta ne voivat antaa käyttäjälleen palautetta, jonka perusteella tämä voi parantaa omaa toimintaansa. Tämä pätee oikeastaan kaikkeen harrastamiseen. Ja kun virtuaalimaailmassa liikkuminen kehittyy, myös kuntoa voi ylläpitää virtuaalitodellisuuden avulla.

Jo nyt monet videopalvelut, kuten esimerkiksi Netflix ja YouTube (Walton 2017), on saatavilla myös virtuaalisena ja 360-videoita on YouTubessa todella paljon. Myös joitain suuren budjetin elokuvia on tehty VR-laseille (Looper 2018). Tämä avaa oven aivan uudenlaisen mediaformaatin maailmaan.

## 2.4 Virtuaalitodellisuuden tulevaisuus

Yksi virtuaalitodellisuuden haasteista on 360-videoiden suuret tiedostokoot. Jo muutamalla pakkaamattomalla 360-videolla voi täyttää nykyisten VR-lasien tallennuskapasiteetin. 5G-verkkojen odotettujen nopeuksien (Fulton III 2019) myötä, myös 360-videoiden suoratoisto on mahdollista (Mundy 2018). 4G-verkkojen nopeudet eivät siihen riitä. Tämä poistaisi paljon VR-tuotannon rajoituksia.

Yhdistetty todellisuus (Mixed Reality) tarkoittaa virtuaalitodellisuuden ja lisätyn todellisuuden (Augmented Reality) yhdistelmää. Teknologian mahtuminen jatkuvasti pienempään kokoon voisi mahdollistaa yhdistetyn todellisuuden sulautumisen arkielämään. Tästä esimerkkinä suuryritysten pyrkimys kehittää yhdistetyn todellisuuden silmälaseja,

joissa teknologia ei ole lainkaan ulkopuolisten ihmisten nähtävissä (Bohn 2018). Tulevaisuudessa silmälasit voisivatkin tarjota paljon muutakin kuin vain näönkorjauksen.

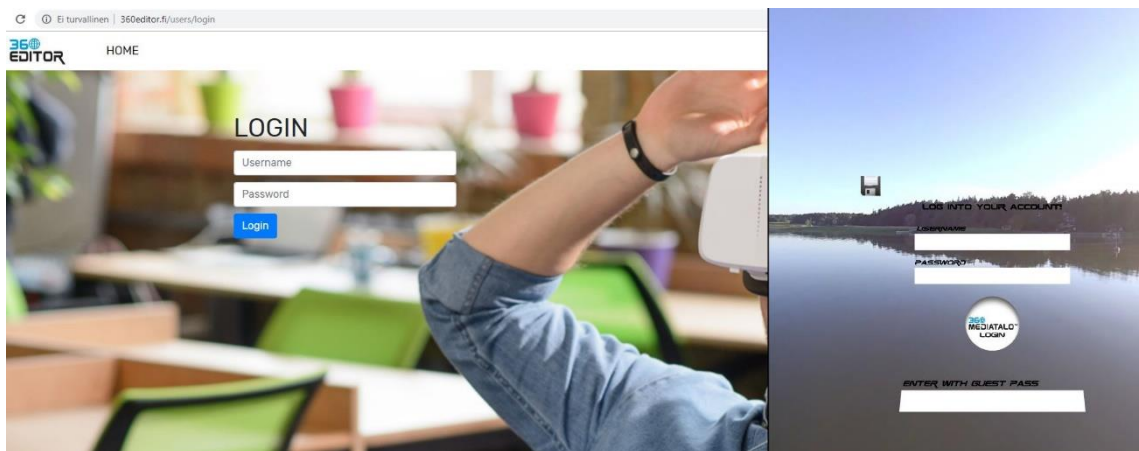
Oculus VR:n tutkimusjohtaja Michael Abrash näkee nykyisten kookkaiden virtuaalilasien vaihtumisen pienikokoisempiin laseihin jo muutamassa vuodessa. Tämä saavutettaisiin muun muassa pancake lens- ja waveguide display -teknologioiden avulla. Näillä teknologioilla myös VR-lasien näkökenttää olisi mahdollista laajentaa niin, ettei käyttäjä voi vahingossakaan katsoa VR-lasien näkökentän ulkopuolelle. Tekoälyn avulla VR-lasit voisivat tarkentaa näkökenttää katseen mukaan. (Fingas 2018.)

Ubisoft-videopeliyritys on kehittänyt yhteistyössä Renaultin kanssa VR-kokemuksia itsestään ajaville autoille (Feltham 2018). Tulevaisuudessa voitaisiinkin nähdä virtuaalitodellisuuden vievän käyttäjän mielen pois myös muista pitkävetteisistä tai epämieluisista tilanteista. Ajatuksena esimerkiksi hammaslääkärissä käyminen voisi olla jopa miellyttävä kokemus, jos potilas saisi toimenpiteen ajaksi viihdettä VR-laseista. Näin voitaisiin myös hillitä joidenkin ihmisten hammaslääkärissä käynnin pelkoa. Sama pätee esimerkiksi lentomatkustamiseen. Lentopelkoa ja pitkän matkan aiheuttamaa tylsyyttä voitaisiin molempia hallita virtuaalitodellisuuden avulla. (Metz 2018.)

Vaikka virtuaalitodellisuutta hyödynnetäänkin jonkin verran jo lähes kaikkialla, teknologian uutuus ja hinta rajoittavat vielä käyttöä. Virtuaalitodellisuuden kehittyessä syntyy uusia käyttötarkoituksia ja jo olemassa olevat käyttötarkoitukset yleistyvät. Yleistymisen myötä myös vanhat palvelut kehittyvät.

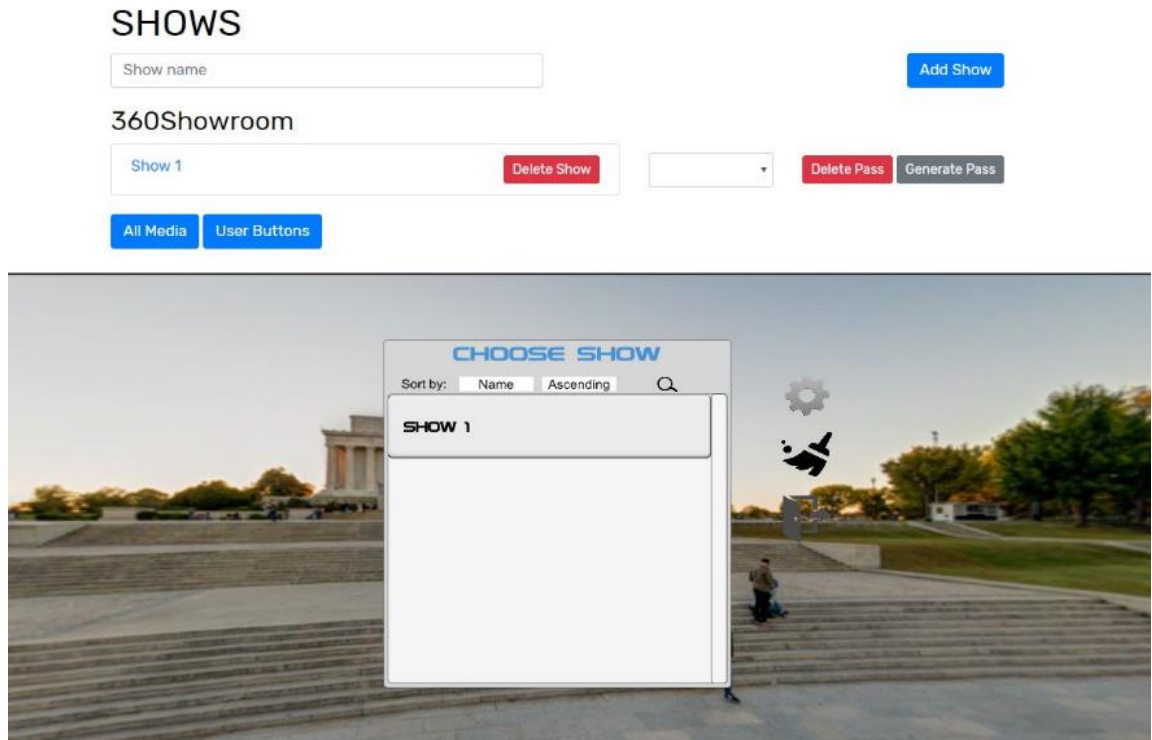
### 3 360EDITOR-SOVELLUKSEN TOIMINTA

Opinnäytetyössä käsitellään projektia, jossa luodaan sisällön jako- ja luomistyökaluja Oculus GO -laseille. Käyttäjän näkökulmasta sovelluksen käyttö alkaa käyttäjätunnuksen ja salasanan saamisesta. Samoilla tunnuksilla kirjaututaan sekä sisällönluomis- ja jakamisverkkosivuille että sovellukseen laseille, josta sisältöä voi katsoa. Kuvassa 1 vasemmalla editorin ja oikealla VR-sovelluksen kirjautumissivu. Yksittäistä show'ta voi katella myös vieraspassilla. Tästä eteenpäin verkkosivuihin viitataan termillä editori.



**Kuva 1** Kirjautumiskäytännöt.

Kirjautumisen jälkeen sovelluksen ensimmäinen näkymä on päävalikko. Päävalikosta löytyy niin editorin, kuin sovelluksenkin puolelta kuvassa 2 näkyvä show-valikko. Show luodaan editorissa ja se voi olla yksittäinen 360-media tai työkaluilla luotu laaja kokonaisuus. Editorin päävalikosta voidaan luoda myös vieraspasseja. Vieraspassi antaa katseluoikeudet yhdelle show'ille ilman käyttäjätunnusta. Sovelluksen päävalikosta löytyy show-valikon lisäksi asetukset-, datan poisto- ja uloskirjautumispainikkeet.



**Kuva 2** Show-valikot.

Asetusvalikossa käyttäjä voi vielä räätälöidä kokemusta itselleen sopivammaksi muokkaamalla esimerkiksi animaatiota. Käyttäjä voi myös halutessaan laittaa päälle joitain vasta kehitysvaiheessa olevia ominaisuuksia. Show'n painiketta painamalla editorissa käyttäjä pääsee luomaan show'n. Sovelluksessa painiketta painamalla käyttäjä pääsee katsomaan sitä. Show'n datanpoistopainikkeella käyttäjä voi tyhjentää lasit kaikesta yksittäisen show'n datasta. Tämä ei kuitenkaan poista itse show'ta pilvestä vaan vapauttaa tilaa VR-laseista. Jos esimerkiksi show'n latauksessa tapahtuu virhe, datan poisto ja uudelleenlataus saattaa korjata tämän virheen. Show'n poisto pilvestä tehdään verkkoselaimella.

### 3.1 Show'n luominen

Ensimmäinen vaihe uuden show'n rakentamisessa on median lataaminen verkkosivustolle. Tällä hetkellä sovellus tukee seuraavia mediaformaatteja:

- 360-kuvat
- 2D-kuvat
- PDF

- 360-videot
- 2D-videot.

Kuvassa 3 näkyy, kuinka ladatut mediat listautuvat latausnäkömön alapuolelle. Tästä valitaan myös show'n aloitusmedia.

## SHOW 1

% OF SHOW DATA USED:

0.05%

**CHOOSE FILE** Original filename: No file chosen

Change file name here (without extension):

No file chosen

Choose Upload Folder:



360 Images

**UPLOAD**

Upload Progress:

\_\_\_\_\_

Sort by:

NAME  NAME  NEWEST OLDEST

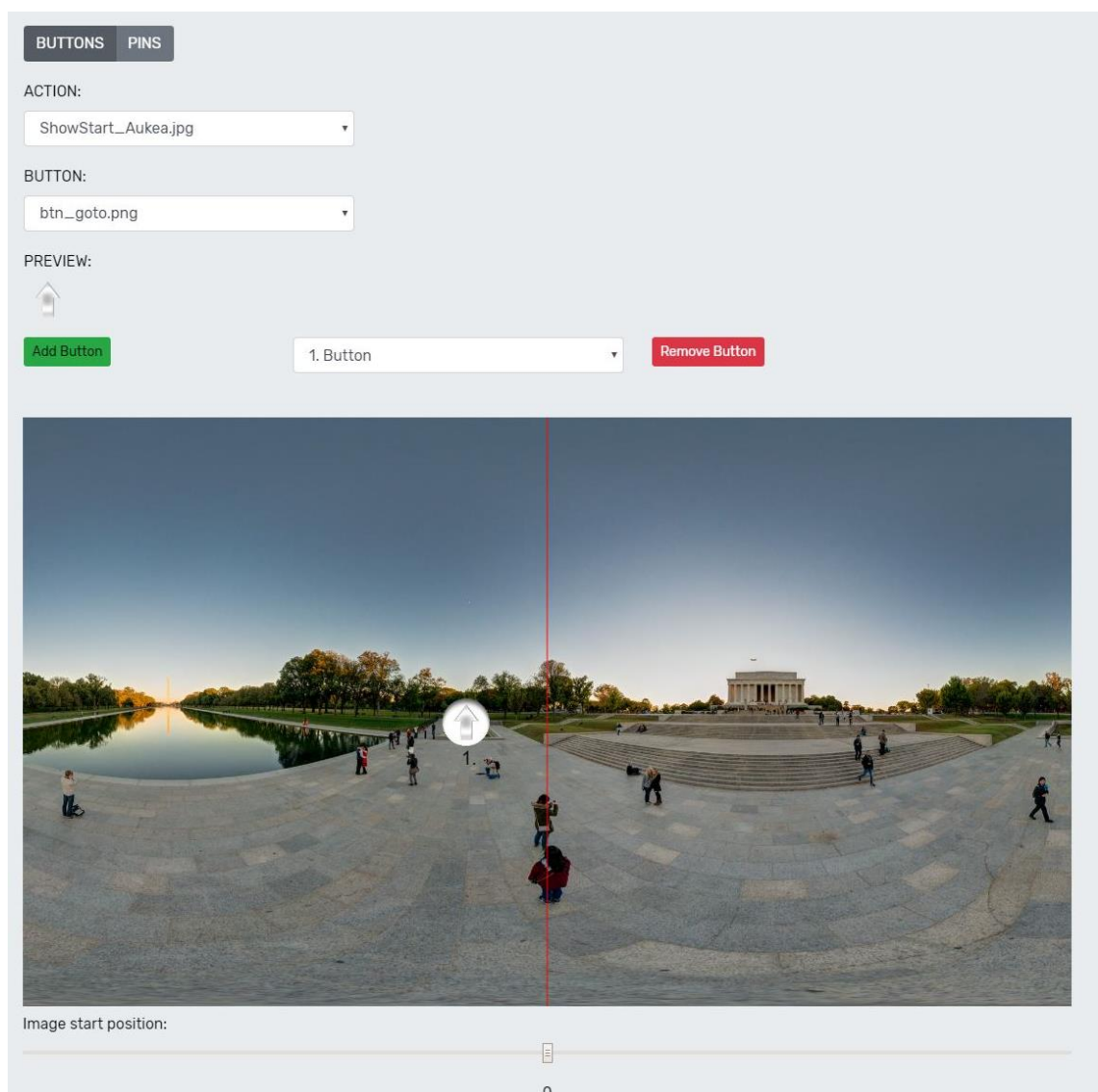
ShowStart\_Aukea.jpg



**Kuva 3** Median latausnäkömön, jonka alapuolella lista 360-kuvista.



Painamalla mediaa aukeaa kuvassa 4 näkyvä kuvastudio. Kuvastudiossa lisätään mediallyle toiminnallisuuksia. Tämän hetkisiä toiminnallisuuksia ovat painikkeet, pinnit, ja kyselykortit. Painikkeilla voi joko siirtyä toiseen 360-mediaan tai avata 2D-median. Pinnit ovat pyöreitä objekteja, jotka suurenevat painettaessa. Ne voivat sisältää tekstiä tai mediaa. Pinni voi myös olla aluksi näkymätön. Näin show'n luoja voi asettaa pinnit käyttäjän löydettäväksi. Kyselykortin vastaukset on mahdollista tallentaa, jolloin kyselyn luoja on mahdollista käydä läpi käyttäjien vastaukset. Kuvastudiosta voi myös valita, missä katselukulmassa valittu 360-media alkaa. Tämä ominaisuus on oleellinen esimerkiksi luodessa kuvien välistä navigointia.

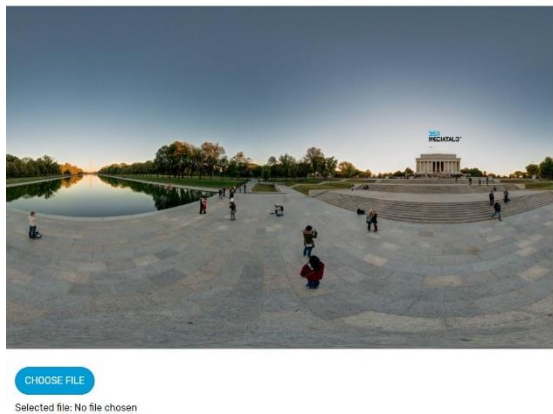


**Kuva 4** Kuvastudio.

### 3.2 Muita ominaisuuksia

Sovelluksen ulkoasun voi helposti kustomoida editorissa käyttäjälle mieluisaksi. Tämä ominaisuus on tärkeä esimerkiksi yritysasiakkaille, jotka haluavat oman brändinsä näkyviin yrityksen sisäisessä käytössä. Kuvassa 5 editoriin on ladattu 360-kuva, jossa on 360Mediatalon logo.

DATA PLAN:  
PLATINUM  
OCULUS MENU



**Kuva 5** Alustan taustakuva.

Asiakkaat voivat valita kolmesta erilaisesta 360Editorin käyttölisenssistä. Käyttäjän profiilissa on nähtävissä tämän tilauksen taso. Tällä hetkellä käytössä olevat vaihtoehdot ovat

- Silver: 1GB / show. 5 show'ta.
- Gold: 2GB / show. 15 show'ta
- Platinum: 3GB / show. 50 show'ta.

## 4 TEKNINEN TOTEUTUS

### 4.1 Google Cloud Platform osana ympäristöä

Alustan kokonaisuudessa on kaksi osa-aluetta: VR-sovellus ja editori. Sovellus ja editori tallentavat kaiken tarvittavan datan Google Cloud Platform -palveluun (jatkossa GCP), josta tarvittava data myös haetaan. Kyseistä palvelua käytetään myös kaiken muun datan säilyttämiseen.

Alustalle on kehitetty tiedostorakenne, joka luodaan GCP:hen jokaiselle uudelle käyttäjälle. Näin GCP saadaan pidettyä järjestyksessä ja sinne tallennettavaa dataa on helppo käsitellä. Osa datasta tallennetaan Google Cloud Storageen (jatkossa GCS) ja osa Cloud SQL tietokantaan. Cloud SQL mahdollistaa MySQL-tietokannan luonnin ja isännöinnin GCP:llä.

### 4.2 VR-ympäristötyökalujen luominen verkkoselaimelle

Verkkosivujen alustana toimii GCP:llä pyörivä Google App Engine (jatkossa GAE), jolla pystyy luomaan palvelittomia sovelluksia. Palvelittomalla sovelluksella tarkoitetaan yleisesti pilvipalvelussa pyörivää sovellusta, joka dynaamisesti hallitsee resursseja niiden käytön mukaan. Tämä tarjoaa mahdollisuuden keskittyä koodin kirjoittamiseen ja jättää palvelinpuolen resursseista huolehtimisen GCP:lle.

#### 4.2.1 Esivaatimukset

Jotta Node.js ympäristöä pystyy käyttää, pitää se asentaa tietokoneelle. Ympäristön voi asentaa usealla erilaisella tavalla, mutta suositeltavin tapa on ladata asennustiedosto ja asentaa ympäristö sen avustuksella. Asennustiedoston voi ladata Node.js-verkkosivuilta (Node.js). Asennuksen jälkeen projekti pitää alustaa *npm init* -konsolikomennolla, jolla projektille luodaan package.json-tiedosto, joka pitää sisällään projektin asetukset ja samalla säilyttää tiedot siitä, mitä moduuleja projekti käyttää.

## 4.2.2 Palvelimen toiminta

Palveliton sovellus on terminä hieman harhaanjohtava, sillä se ei tarkoita, etteikö verkkosivulla olisi palvelinta. Palvelin luodaan Node.js-ympäristölle tehdyllä Express-sovel-luskehyksellä, joka vastaa palvelinpuolen koodista.

*Express-* ja *body-parser* -moduulit tuodaan koodin alussa, minkä jälkeen sovellus alus-tetaan käyttämään niitä. Jotta moduuleja pystyy käyttämään, ne pitää asentaa ja halu-tesa tallentaa projektin package.json-tiedostoon, joka onnistuu *npm install <moduulin nimi>* -konsolikomennolla. Jos moduulin haluaa tallentaa package.json-tiedostoon, voi konsolikomentoon lisätä option *--save*. *Express* on moduuli, joka luo palvelimen ja *body-parser* varmistaa, että sovellus pystyy vastaanottamaan dataa asiakasohjelmalta (client). Kuvassa 6 rivillä 11 luodaan aloitussivu, joka palauttaa JSON-objektin, toisin sanoen, kun verkkosivujen aloitussivulle menee, käyttäjä näkee tämän viestin. Lopuksi palvelin alustetaan, jolloin sovellus lähettää palvelimen konsoliin viestin, kun palvelin on käynnis-tynty. Palvelin käynnistetään kutsumalla tiedostoa, joka määrittellään *npm init* -komennon aikana. Vakiona tiedoston nimi on *index.js*, tällöin käynnistys konsolissa tapahtuu *node index.js* -komennolla.

```
1  const express = require('express');
2  const bodyParser = require('body-parser');
3
4  // Init app
5  const app = express();
6
7  // Body Parser Middleware
8  app.use(bodyParser.json());
9
10 // Home Route
11 app.get('/', function(req, res) {
12   res.json({
13     message: 'Hello World!'
14   });
15 });
16
17 const port = process.env.PORT || 8080;
18 // Start Server
19 app.listen(port, function(){
20   console.log(`Server started on port ${port}...`);
21 });
```

Kuva 6 Express-palvelimen perusta.

Kuten kuvassa 7, palvelimelle pitää määrittää staattinen kansio, jota kutsutaan myös nimellä julkinen kansio. Asiakasohjelma pääsee käsiksi vain tiedostoihin, jotka ovat staattisessa kansiossa, joten käyttäjän tulee pitää huoli, ettei staattiseen kansioon laiteta minikäänlaista arkaluontoista materiaalia. Staattiseen kansioon yleisesti tallennetaan HTML-, CSS- ja JavaScript-tiedostoja. Näiden lisäksi kansioon voi tallentaa muita tiedostoja, kuten kuvia, joita sivulla halutaan käyttää.

Verkkosivuja voi hahmontaa joko palvelimen tai asiakasohjelman puolella. Opinnäytetyön projektissa on käytössä palvelinpuolen hahmonnus. Hahmonnuksen näkymämoottorina käytetään EJS-moottoria, joka mahdollistaa näkymien hahmonnuksen palvelinpuolelta asiakasohjelmalle. Näkymämoottorien avulla asiakasohjelma pystyy vastaanottamaan dataa palvelimelta, joka mahdollistaa dynaamisten sivujen, kuten käyttäjäkohtaisten sivujen luomisen.

```
// Set Public Folder
app.use(express.static(path.join(__dirname, 'public')));
app.use(favicon(path.join(__dirname, 'public/images', '360EditorPyorea.png')));

// Load view engine
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');
```

**Kuva 7** Näkymämoottorin alustaminen palvelimelle

Express tekee palvelinpuolen hahmonnuksesta helppoa. Asiakasohjelma lähettää GET-pyyntöä palvelimelle ja jos palvelimella on kyseiselle pyynnölle ohjelmoitu reitti, palvelin palauttaa näkymän asiakasohjelmalle. Kuvassa 8 on palvelimelle luotu reitti */route*, joka hahmontaa *view*-nimisen näkymän asiakasohjelmalle. Palvelin etsii näkymäkansiosta näkymän, jonka nimi on *view.ejs*. Koska palvelimelle on määritetty mikä näkymämoottori on käytössä, tiedostopäätettä ei tarvitse kirjoittaa. Ennen näkymän hahmonnusta ohjelmoijalla on mahdollisuus käsitellä palvelinpuolen koodia, jota voi hyödyntää lähettämällä sen asiakasohjelmalle näkymän mukana esimerkiksi muuttujat *title* ja *header* kuvassa 8.

```
app.get('/route', function (req, res) {
  const title = "Title of the page";
  const header = "Welcome to the page!";
  res.render('view', {
    title,
    header
  });
});
```

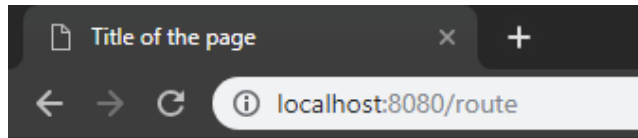
**Kuva 8** Palvelimelle luotu reitti, joka palauttaa näkymän.

Näkymämoottorien luomat näkymät perustuvat HTML-kieleen, joten koodi näyttää hyvin samalta. EJS käyttää HTML:n lisäksi syntaksia `<%= muuttuja %>`, joka antaa mahdollisuuden hyödyntää palvelimelta tulevia muuttujia. Kuvassa 9 EJS-näkymälle annetaan muuttujat *title* ja *header*.

```
<> view.ejs x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7
8   <title><%= title %></title>
9 </head>
10 <body>
11   <h1><%= header %></h1>
12 </body>
13 </html>
```

**Kuva 9** EJS näkymä.

EJS saa palvelimelta muuttujat, jotka hahmonnetaan sivulle niiden palvelimella annetuilla arvoilla. Kuvassa 10 otsikkona on muuttuja *title* ja tekstinä on muuttuja *header*, kun palvelimelle tehdään GET-pyyntö osoitteeseen */route*.



## Welcome to the page!

**Kuva 10** Selaimessa haettu sivu.

Jos asiakasohjelma haluaa lähettää dataa, asiakasohjelmasta tehdään POST-pyyntö palvelimelle. Body-parser-moduuliin lisätään asetus, jolla palvelin pystyy vastaanottamaan muutakin, kun pelkästään JSON-dataa. Kuvassa 11 mahdollistetaan myös lomaketietojen vastaanottaminen.

```
// parse application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({
  extended: false
}));
```

**Kuva 11** Lomaketietojen vastaanottamisen mahdollistava body-parser-moduuliasetus.

Jotta palvelin pystyy vastaanottamaan POST-pyyntöä, palvelimelle täytyy luoda reitti sitä varten. Palvelin vastaanottaa lomaketiedot muuttujaan *req.body*, joka voidaan purkaa yksittäisiin muuttujiin, kuten kuvassa 12 muuttujiin *name* ja *age*. Kuvassa 12 nähdään myös palvelimelle luotu POST-reitti */profile*, joka pyynnön saatuaan hahmontaa asiakasohjelmalle *profile*-näkyvän.

```
app.post("/profile", function(req, res) {
  const { name, age } = req.body;
  res.render("profile", {
    name,
    age
  });
});
```

**Kuva 12** POST-reitti */profile* palvelimella.

Palvelimelle ei pysty tallentamaan tiedostoja, eikä muutakaan tietoa. Ratkaisuna tälle on tietokannat ja muut tiedostontallennusvaihtoehdot. Kuten luvussa 4.1 mainittiin, projektissa käytetään Cloud SQL -tietokantaa ja GCS-varastoa datan tallentamiseen. Virtuaalitetodellisuudessa katseltava materiaali, esimerkiksi 360-kuvat ja -videot, tallennetaan GCS-varastoon ja Cloud SQL -tietokantaan muun muassa käyttäjän ja show'n tiedot.

GCS-varastoon ja Cloud SQL -tietokantaan tietojen tallentaminen tapahtuu POST-pyyntöillä asiakasohjelmalta palvelimelle ja palvelin lähettää datan edelleen tietokantaan. Mikäli asiakasohjelma haluaa käyttää dataa, sen täytyy tehdä GET-pyyntö palvelimelle. Palvelin hakee tietokannasta pyydetyn datan ja lähettää sen asiakasohjelmalle.

### 4.3 Interaktiivisen VR-ympäristön ohjelmointi

Kehityksessä käytetään Unity-pelimoottoria, sillä se sopii tarkoitukseen useasta erisyystä. Ensinnäkin, Oculus on itse julkaissut hyvät työkalut ja ohjeet laitteilleen kehittämistä varten Unitylle. Unityn on myös sanottu olevan helppokäyttöinen ja se tukee Microsoftin C#-ohjelmointikieltä.

#### 4.3.1 Esivaatimukset

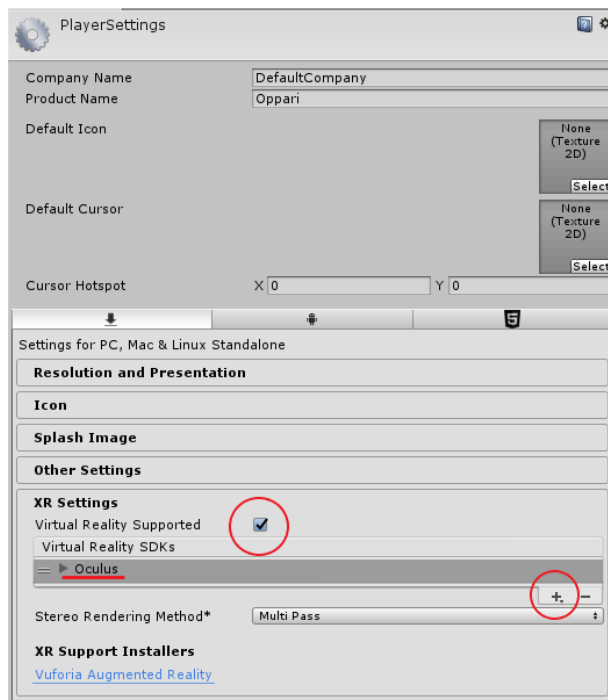
Oculus GO -laseissa on Android-käyttöjärjestelmä. Jotta kehitys voidaan aloittaa, Unityyn täytyy ensin asentaa Android SDK. Tähän löytyy hyvät ohjeet Unityn omasta dokumentaatiosta (Unity 2018), joten aiheeseen ei mennä sen syvemmin. Yksinkertaisin tapa asentaa Android SDK on dokumentaation kohta 1b: Android Studio IDE:n asentaminen (Unity 2018).

Oculus on kehittänyt Unitylle hyvät, ilmaiset työkalut (Oculus). Työkalut sisältävät muun muassa tuen Oculusin ohjaimille sekä valmiiksi ohjelmoidun pään asennonseurannan. Työkalut täytyy olla asennettuna ennen kehityksen aloitusta.



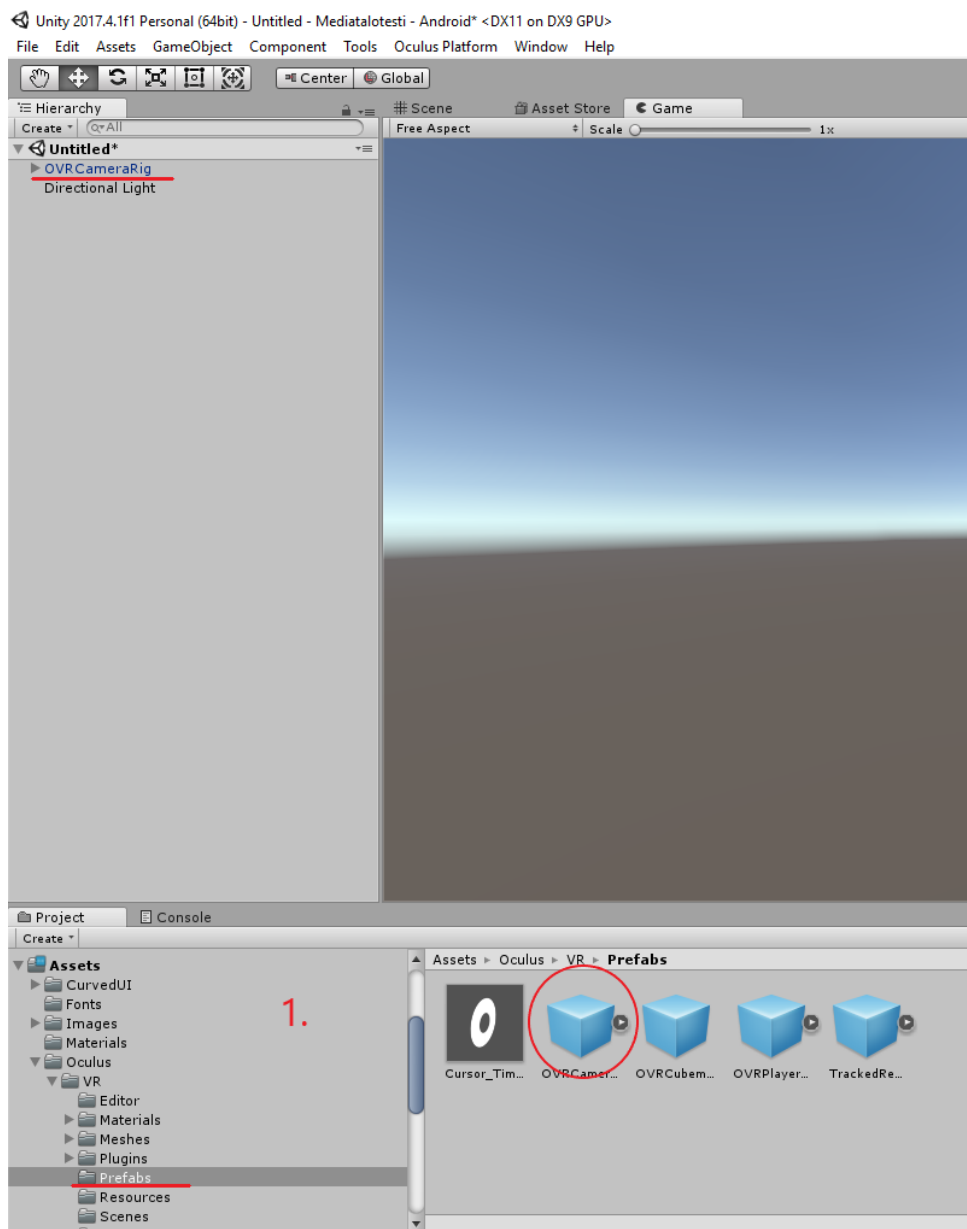
### 4.3.2 360-kuvan esittäminen Unityllä ja kokoaminen laseihin

Esivaatimusten jälkeen koontiasetukset täytyy määrittää. Koontiasetusvalikon saa auki valitsemalla Unityn valikosta *File -> Build Settings*. Valikon *Platform*-kohdasta käyttöjärjestelmäksi valitaan Android. Seuraavaksi koontiasetusvalikosta valitaan *Player Settings... -> XR Settings*. Tästä näkymästä valitaan ”Virtual Reality Supported” -valintaruutu. Alle ilmestyvästä +-näppäimestä painamalla listaan voidaan lisätä Oculus. Valmiit asetukset on nähtävissä kuvassa 17.



**Kuva 13** Virtuaalitodellisuustuen päälle laittaminen.

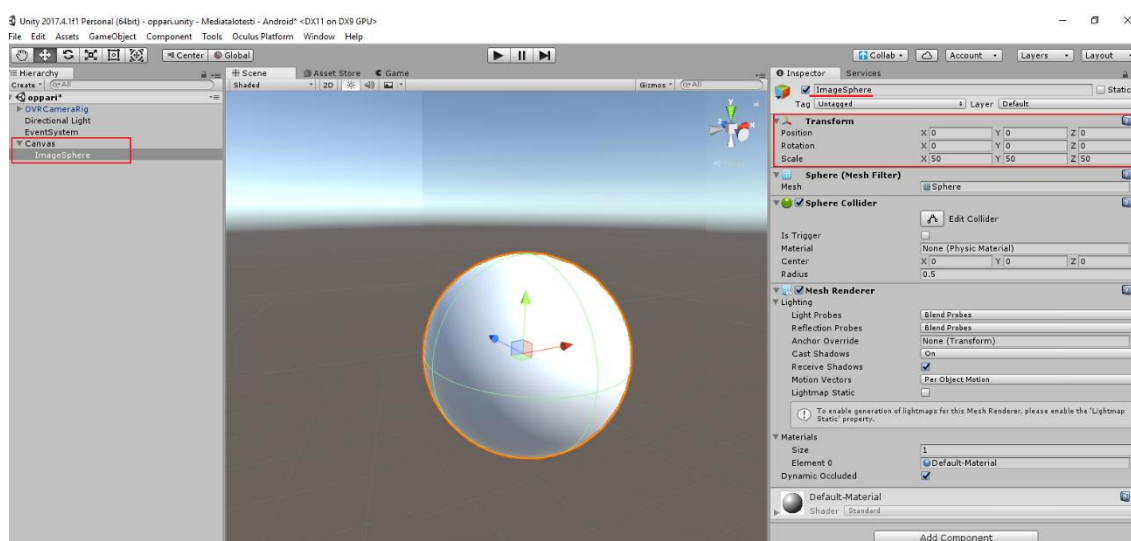
Unityn oletusnäkyvässä on vasemmalla hierarkiapalkki ja alhaalla projektikansioiden navigointipalkki. Hierarkiapalkissa on oletuksena Main Camera -objekti. Tämä voidaan poistaa valitsemalla se ensin hiiren vasemmalla näppäimellä ja painamalla sitten *Delete*. Myös Directional Light -objektin voi poistaa. Poistetun kameran tilalle laitetaan OVRCameraRig-prefab, joka voidaan raahata hiirellä navigointipalkista hierarkiaan. Kuvassa 18 nähdään, miltä oikein asetettu OVRCameraRig-prefab näyttää. OVRCameraRig-prefab tulee Oculusin Unity-työkalujen mukana ja sen avulla voidaan käyttää Oculus GO - lasien pään asennon seuranta kameran liikuttamiseen.



**Kuva 14** OVRCameraRig-prefabin käyttöönotto.

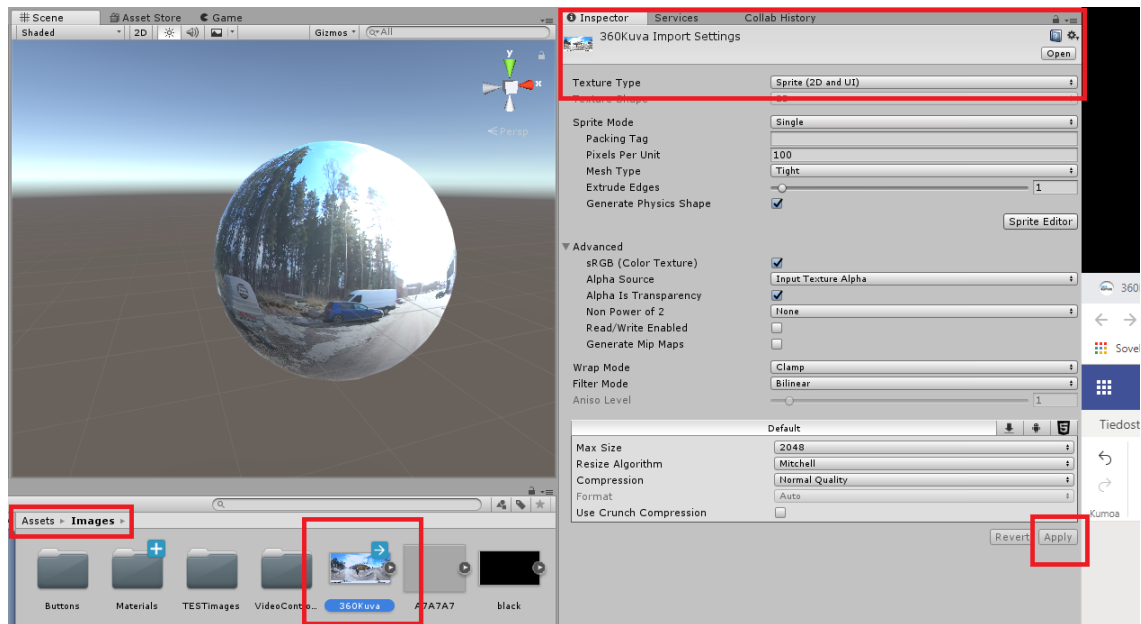
Jotta 360-kuva voidaan esittää kameralle, sen ympärille tarvitaan kuvassa 19 näkyvä pallo-objekti, johon 360-kuva asetetaan. Ensin on kuitenkin varmistettava, että kamera on pelimaailman keskellä valitsemalla OVRCameraRig-objekti hierarkiasta. Unityn oletusnäkyvässä oikealla on Inspector-palkki. Inspectorissa ensimmäisen ominaisuuden (transform) ensimmäinen kohta Position, kertoo valitun objektin sijainnin pelimaailmassa. Kaikkien koordinaattien täytyy tässä vaiheessa olla 0. Seuraavaksi luodaan Canvas-objekti, jonka lapsiobjekteiksi asetetaan pelimaailmassa näkyvät objektit, painamalla

hierarkiasta hiiren oikealla näppäimellä ja valitsemalla *UI -> Canvas*. Canvas-objekti näkyy hierarkiassa. Canvas-objektilla on Inspectorissa Canvas-ominaisuus, jossa on *Render Mode* -valikko, josta oletusasetus *Screen Space – Overlay* vaihdetaan *World Space* -vaihtoehdoksi. Näin käyttäjä voi itse valita Canvas-objektin koordinaatit, jotka täytyy myös tässä vaiheessa asettaa nolaksi. Painamalla hierarkiasta Canvas-objektia hiiren oikealla näppäimellä, voidaan luoda pallo sen lapsiobjektiksi valitsemalla *3D Object -> Sphere*. Inspectorin ensimmäinen ominaisuus on objektin nimi. Pallon nimeksi vaihdetaan *ImageSphere* työskentelyn selkeyttämiseksi. Myös uuden *ImageSphere*-objektin koordinaattien on oltava 0. *ImageSpheren* transform ominaisuuden *Scale*-kohtaan asetetaan kaikkien akselien arvoksi 50. Näin pallon sisälle mahtuu myös muita objekteja.



**Kuva 15** *ImageSphere* pelimaailman keskellä.

Navigointipalkkiin voi tuoda minkä tahansa objektin hiirellä vetämällä. Sitä ennen kannattaa kuitenkin luoda kuville oma kansio navigointipalkkiin työskentelyn selkeyttämiseksi. Tämä onnistuu painamalla navigointipalkkia hiiren oikealla näppäimellä ja valitsemalla valikosta *Create -> Folder*. Jotta kuva voidaan näyttää objektissa, sen tekstuurityypiksi on ensin asetettava *Sprite*. Kuvan tekstuurityypin voi vaihtaa valitsemalla kuva navigointipalkista ja valitsemalla Inspectorin *Texture Type* -valikosta *Sprite (2D and UI)*. Lopuksi painetaan *Apply*. Tekstuurityypin muuttaminen näkyy kokonaisuudessaan kuvassa 20. Nyt kuvan voi raahata navigointipalkista pallo-objektiin ja se näkyy pallon tekstuurina.



**Kuva 16** Kuvan tekstuurityypin muuttaminen Spriteksi.

Unityn ohjelmointitavan takia, Unity näyttää objektien tekstuurit objektin ulkopuolella. Jotta tekstuuri saadaan näkyviin pallon sisäpuolelle, ja näin myös kameralle, pallo-objektille on luotava varjostin (*shader*). Myös varjostimille kannattaa luoda oma kansio navigointipalkkiin. Shader luodaan painamalla navigointipalkkia oikealla hiiren näppäimellä ja valitsemalla *Create -> Shader -> Standard Surface Shader*. Varjostimen voi kirjoittaa tuplaklikkaamalla sen ikonia navigointipalkissa. Kuvassa 21 näkyy esimerkkivarjostinkoodi *Insideout*, joka piirtää tekstuurin objektin sisäpuolelle.

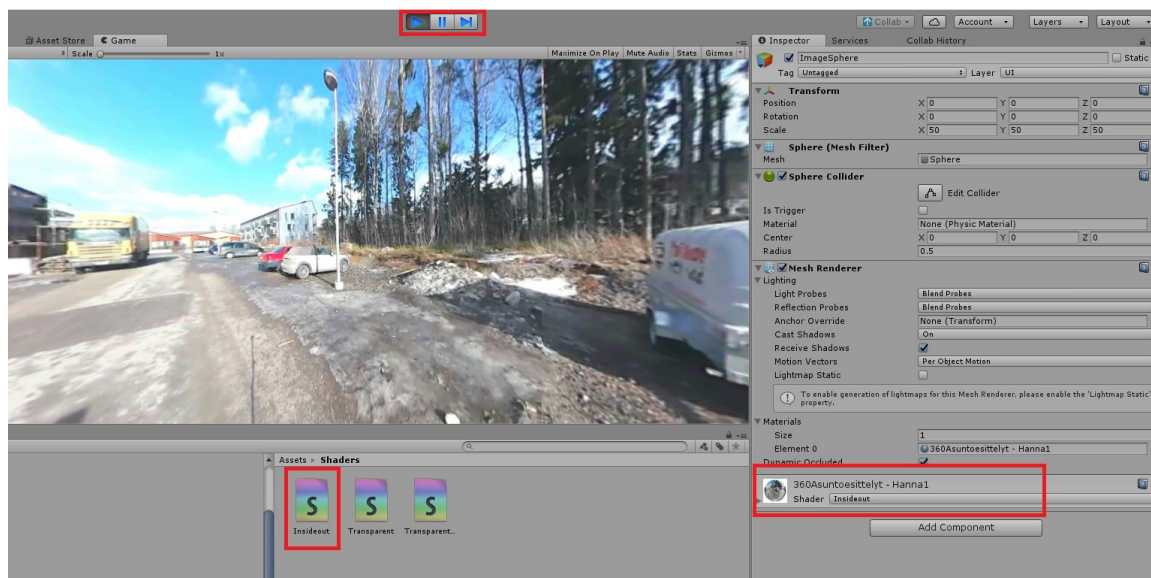
```

5 Shader "Insideout" {
6 Properties {
7     _MainTex ("Base (RGB)", 2D) = "white" {}
8 }
9
10 SubShader {
11     Tags { "RenderType"="Opaque" }
12     Cull front // ADDED BY BERNIE, TO FLIP THE SURFACES
13     LOD 100
14
15     Pass {
16         CGPROGRAM
17             #pragma vertex vert
18             #pragma fragment frag
19
20             #include "UnityCG.cginc"
21
22             struct appdata_t {
23                 float4 vertex : POSITION;
24                 float2 texcoord : TEXCOORD0;
25             };
26
27             struct v2f {
28                 float4 vertex : SV_POSITION;
29                 half2 texcoord : TEXCOORD0;
30             };
31
32             sampler2D _MainTex;
33             float4 _MainTex_ST;
34
35             v2f vert (appdata_t v)
36             {
37                 v2f o;
38                 o.vertex = UnityObjectToClipPos(v.vertex);
39                 // ADDED BY BERNIE:
40                 v.texcoord.x = 1 - v.texcoord.x;
41                 o.texcoord = TRANSFORM_TEX(v.texcoord, _MainTex);
42                 return o;
43             }
44
45             fixed4 frag (v2f i) : SV_Target
46             {
47                 fixed4 col = tex2D(_MainTex, i.texcoord);
48                 return col;
49             }
50         ENDCG
51     }
52 }
53
54 }

```

**Kuva 17** Inside Out -shader.

Uuden varjostimen voi asettaa pallo-objektille sen inspectorin alimmaisesta ominaisuudesta *Shader*. Sovelluksen voi käynnistää painamalla play-painiketta Unityn yläreunasta. Pelaaja on nyt kuvan keskellä, kuten kuvassa 22.

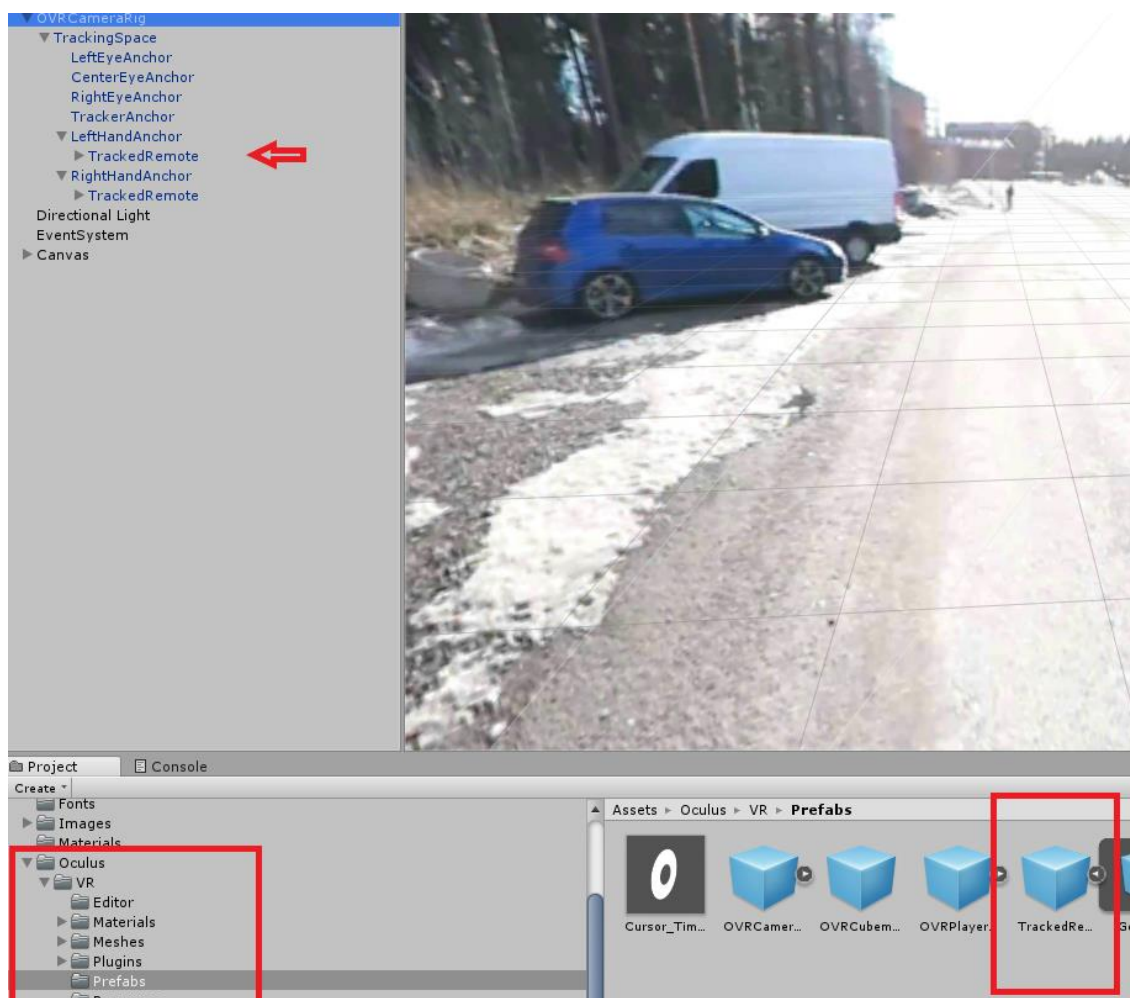


**Kuva 18** Pelaaja on 360-kuvan keskellä.

Ennen kuin sovelluksen voi rakentaa laseihin, lasien täytyy olla USB-johdolla kiinni tietokoneessa ja laseissa täytyy olla kehittäjätila päällä (Oculus 2018). Koontiasetusvalikon *Add open scenes* -painike lisää luodun näkymän laseille koottavaksi. *Build And Run* -painike kokoaa sovelluksen laseihin. Kun rakentaminen on valmis, sovelluksen voi käynnistää laseissa. Pelaajan ympärillä näkyy aiemmin lisätty kuva ja pelaaja voi päätään liikuttamalla katsoa ympärilleen.

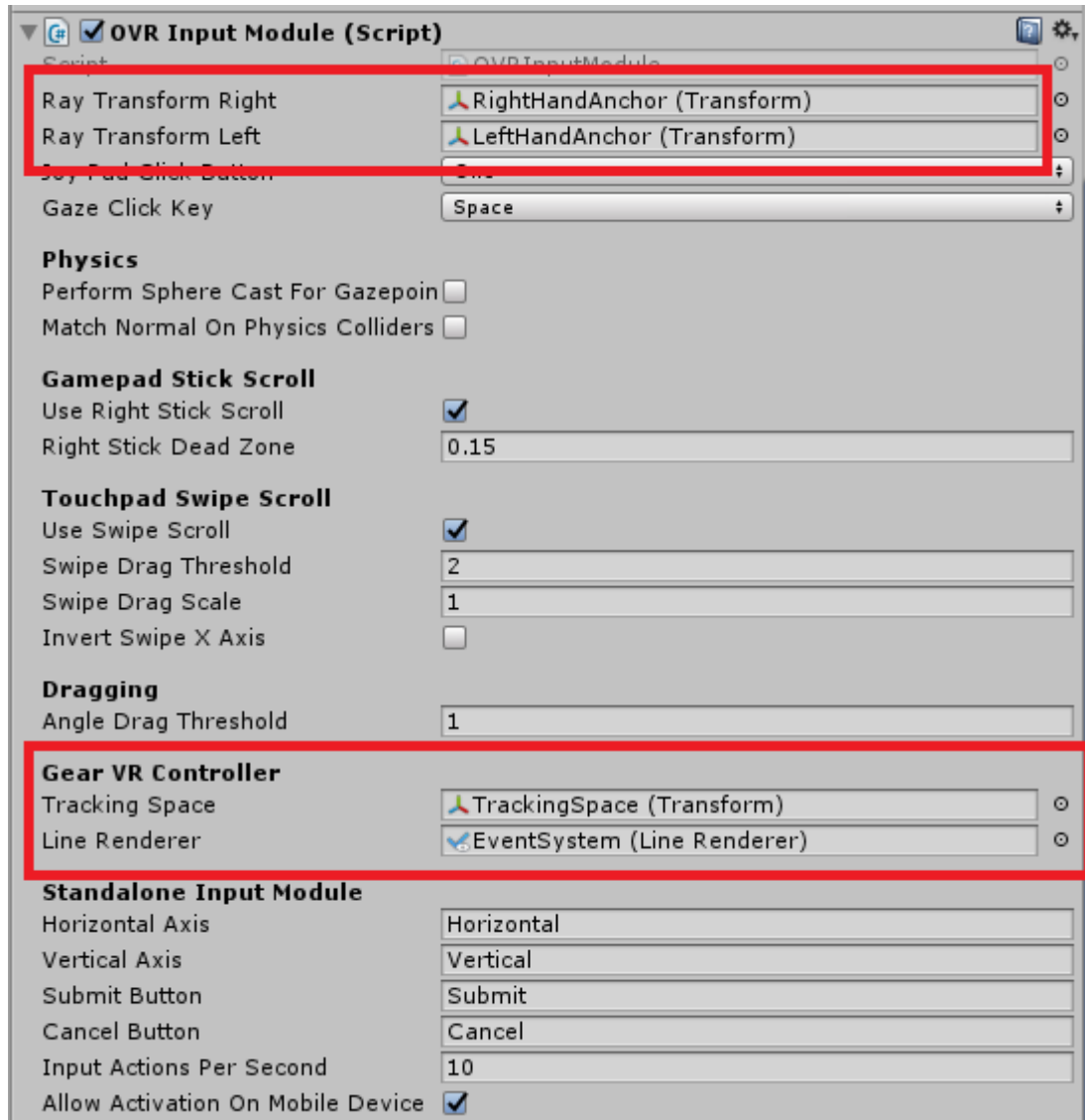
#### 4.3.3 Ohjaaminen ja interaktiivisuus VR-maailmassa

Tässä luvussa pelimaailmaan luodaan ohjain ja painike. Painikkeesta tehdään interaktiivinen ohjaimen kanssa ja painiketta painamalla pallo-objektin 360-kuva vaihtuu. Oculusin Unity-työkaluista löytyy ohjaimille valmiit mallit nimellä *TrackedRemote*. *TrackedRemote*-prefabin löytää navigoimalla projektikansioissa *Oculus* -> *Prefabs*-kansioon. Tämä prefab asetetaan hiirellä raahaamalla *RightHandAnchor* ja *LeftHandAnchor*-peiliobjektien lapsiobjekteiksi, kuten kuvassa 23. Nämä objektit puolestaan löytyvät *OVRCameraRigin* lapsiobjekteista, jotka saa näkyviin painamalla pientä nuolta peliobjektin nimen vasemmalla puolella hierarkiassa. Lapsiobjektit voi myös piilottaa samalla nappulalla. Molempien *TrackedRemote*-objektien Inspectorissa on kohta *Controller*, johon täytyy määrittää kumman käden ohjainta malli esittää.



**Kuva 19** TrackedRemote-prefab hierarkiassa.

Hierarkiassa on valmiina EventSystem-peliobjekti, joka vastaa tiedonkulusta syöttölaitteen ja peliobjektien välillä. Objektin Inspectorissa on Standalone Input Module -komponentti, joka voidaan poistaa painamalla komponentin oikeassa yläkulmassa olevaa ratsikonäköä ja valitsemalla Remove Component. Inspectorin Add Component -painikkeella objektiin lisätään Rect Transform, Canvas Renderer, OVR Input Module ja Line Renderer -komponentit. OVR Input Module komponenttiin Ray Transform Right kohtaan tulee raahata hierarkiasta RightHandAnchor-objekti. Vastaavasti myös LeftHandAnchor-objekti Ray Transform Left kohtaan. Tracking Space kohtaan raahataan saman niminen objekti OVR Camera Rigin lapsiobjektista. Line Renderer kohtaan raahataan itse EventSystem-objekti. OVR Input Modulen valmiit asetukset on nähtävissä kuvassa 24.

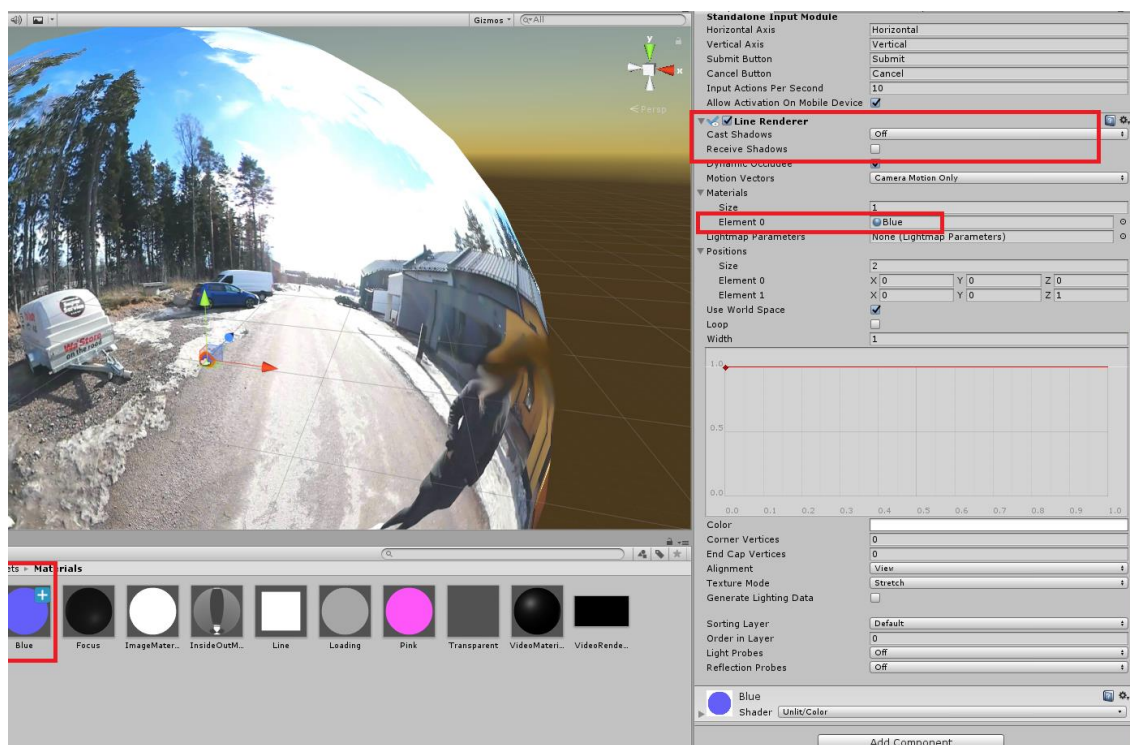


**Kuva 20** OVR Input Module.

Line Renderer -komponentin Cast Shadows kohtaan valitaan vaihtoehto Off. Myös Receive Shadows kohdan valinta täytyy poistaa. Kohdassa Width valitaan osoittimen paksuus, jota kannattaa pienentää huomattavasti. Komponentin Materials-otsikon alla on kohta Element 0, johon valitaan osoittimen värimateriaali. Tämä materiaali on kuitenkin ensin luotava. Myös materiaaleille on hyvä luoda oma kansio navigointipalkkiin. Materiaali luodaan painamalla hiiren oikealla näppäimellä navigointipalkkia ja valitsemalla *Create->Material*. Materiaalille kannattaa antaa yksilöivä nimi esimerkiksi halutun värin mukaan. Materiaalin inspectorissa shaderiksi valitaan *Unlit->Color*. Main Color kohdasta voidaan valita osoittimelle haluttu väri. Materiaali on valmis ja se voidaan asettaa osoit-

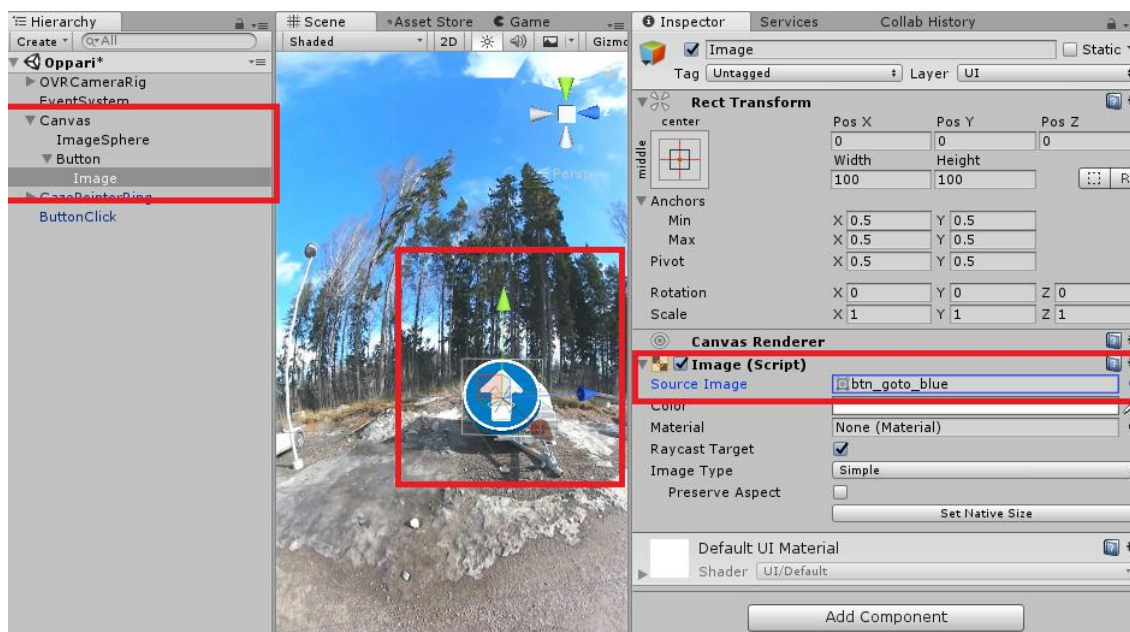


timen väriksi palaamalla Line Rendererin inspektoriin ja painamalla pientä ympyrää Element 0 kentän vieressä. Tämä on nähtävissä kokonaisuudessaan kuvassa 25. Aukeavasta valikosta valitaan juuri luotu materiaali. Samalla tavalla myös OVRCameraRigiin on lisättävä komponentit OVRRaycaster ja OVRPhysics Raycaster, jotka tulivat Oculusin työkalujen mukana. Näihin komponentteihin ei tarvitse tehdä muutoksia. Jotta Canvas-objektin lapsiobjektit voivat reagoida ohjaimen toimintoihin, OVRRaycaster-komponentti on lisättävä myös Canvas-objektiin.



**Kuva 21** Line Renderer ja siihen luotu Blue-materiaali.

Ensimmäinen vaihe VR-ohjauksen kanssa toimivan painikkeen rakentamisessa on tyhjän peliobjektin luominen Canvas-objektin lapsiobjektiksi. Näin voi tehdä klikkaamalla Canvasta hiiren oikealla painikkeella ja valitsemalla valikosta *Create Empty*. Uusi *GameObject*-niminen peliobjekti voidaan nimetä esimerkiksi *Buttoniksi*. Painikkeelle annetaan kuva painamalla sitä hiiren oikealla näppäimellä ja valitsemalla *UI->Image*. Kuva on painikkeen lapsiobjekti ja sille voidaan määrittää Inspectorissa lähdekuva painamalla pientä ympyräkuvaketta *Source Image* -kohdassa. Kuten kuvassa 26 nähdään, painikkeen voi sijoittaa pelimaailmaan joko sen ympärillä näkyvillä nuolilla, tai Inspectorin *Position*-kohdassa.



**Kuva 22** Uusi painike on luotu ja näkyy pelimaalimassa.

Painikkeelle annetaan funktio valitsemalla sen Inspectorista *Add Component* -> *Event Trigger*. Event Trigger käynnistää ohjelmistossa jonkin tapahtuman. Jos Event Triggeristä valitaan *Add New Event Type* -> *PointerClick*, sovellus tietää suorittaa tapahtuman, kun painiketta painetaan. Jotta Event Triggerille voidaan antaa suoritettava tapahtuma, se täytyy ensin ohjelmoida. C#-kooditiedosto luodaan painamalla projektin navigointipalkkia hiiren oikealla näppäimellä ja valitsemalla *Create* -> *C# Script*. Hyvä tiedostonimi voisi olla esimerkiksi *ImageChanger*, sillä tiedoston tehtävä on vaihtaa ohjelmiston taustalla näkyvä kuva. Kooditiedosto avautuu kaksoisnapauttamalla. Kooditiedoston *Start*- ja *Update*-funktiot voi poistaa ja niiden tilalle kirjoitetaan painiketta painamalla kutsuttava funktio. Kuvassa 27 on esimerkkikoodi, joka vaihtaa taustakuvan. On tärkeää huomata, että jotta painike voi kutsua funktiota, funktion tyyppin täytyy olla *public*.

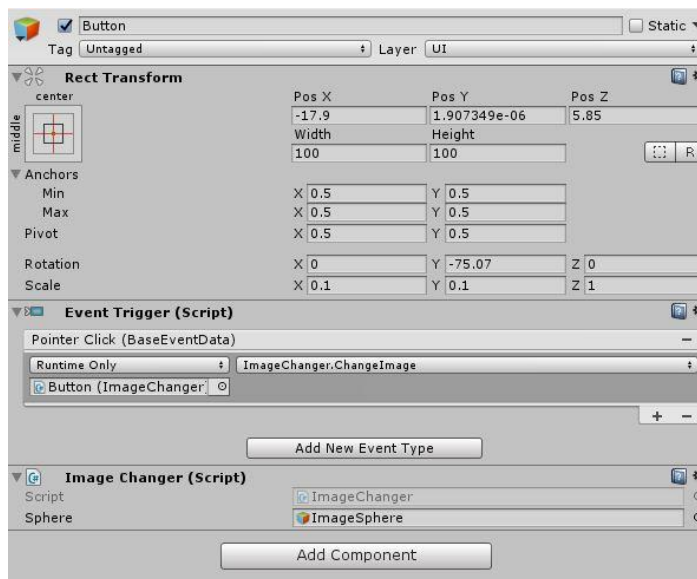
Kuvassa 28 kooditiedosto lisätään luotuun painikkeeseen sen Inspectorin *Add Component*-painikkeella. Kooditiedosto löytyy kirjoittamalla sen nimi vasta avautuneeseen tekstilaatikkoon. Julkisesti määritettävät muuttujat, kuten esimerkiksi kuvan 27 sphere-muuttuja, raahataan hierarkiasta omille nimetyille laatikoilleen. Painike saadaan kutsu- maan funktiota painamalla ensin sen Event Trigger -komponentin +-painiketta. Esiin avautuvaan laatikkoon raahataan se objekti, jossa koodi on, eli tässä tapauksessa itse *Button*-peliohjekti. Event Triggerin funktio-valintapalkista valitaan ensin kooditiedosto ja sitten suoritettava funktio. Kun sovellus on rakennettu uudelleen VR-laseille, sen ohjaimella voidaan painiketta painamalla vaihtaa taustakuva.

```

ImageChanger.cs
Mediatotalotesti ImageChanger
1 using System.Collections;
2 using System.Collections.Generic;
3 using System.IO;
4 using UnityEngine;
5
6 public class ImageChanger : MonoBehaviour {
7
8     // Peliobjekti asetetaan sovelluksessa
9     public GameObject sphere;
10
11     public void ChangeImage()
12     {
13         // Application.PersistentDataPath on projektin juurikansio
14         byte[] bytes = File.ReadAllBytes(Application.persistentDataPath + "/3600ppari.png");
15         Texture2D image = new Texture2D(1, 1);
16         image.LoadImage(bytes);
17         // Uusi kuva asetetaan pallon kuvaksi
18         sphere.GetComponent<Renderer>().material.mainTexture = image;
19     }
20 }

```

Kuva 23 Kooditiedosto, jonka ChangeImage-funktio vaihtaa pallon kuvan.



Kuva 24 Näkymä toimivan kuvanvaihtopainikkeen inspectorista.

## 5 LOPUKSI

Yksi opinnäytetyön tavoitteista oli tutkia virtuaaliteollisuuden käyttömahdollisuuksia teollisuudessa. Aihetta tutkittiin pääasiassa toimeksiantajan asiakkuuksista, mutta myös ulkopuolisia tutkimuksia ja artikkeleita hyödyntäen. Jatkuvasti suositaan kasvattava virtuaaliteollisuus tarjoaa yhä uudenlaisia ratkaisuja niin teollisuuden kuin myös arkielämänkin ongelmiin ja on jo ennestäänkin ollut käytössä eri aloilla erilaisissa tehtävissä. Yleisin käyttötarkoitus teollisuudessa VR-teknologialle on koulutus, jonka hyödyistä on saatu lupaavia tutkimustuloksia.

Virtuaaliteollisuus kasvattaa suosiotaan edelleen jatkuvasti, eikä ole syytä epäillä, ett-eikö kasvu jatkuisi tulevaisuudessakin teknologian kehittyessä ja virtuaalilasi-hintojen laskiessa. Teknologia pienenee ja yleistyy, minkä seurauksena lähitulevaisuudessa voidaan nähdä sosiaalisesti hyväksyttävän näköisiä AR-laseja arkielämässäänkin (Fingas 2018).

Toinen tavoite oli luoda työkalut VR-sisällön luomiseen ja jakamiseen. 360Editor on jo myynnissä, ja sillä on aktiivisia käyttäjiä. Tässä opinnäytetyössä ei käsitelty 360Editorin kehitystä kokonaisuudessaan, sillä alustaa kehitetään eteenpäin jatkuvasti. Kuitenkin teoksen avulla on helppo aloittaa oma VR- ja verkkosivuprojekti.

Projektin perusta ja sen käyttöön vaadittavat perusominaisuudet ovat toimivat, mutta niitä voi aina kehittää eteenpäin. Lisäominaisuuksia kehitetään myös niin nopeasti kuin mahdollista, asiakkaiden kehitysehdotuksia kuunnellen.

Projektina 360Editor on siinä mielessä onnistunut, että sillä on kysyntää jo alle vuoden kehitystyön jälkeen. Vaikka sovellus on vasta alkutekijöissään, sille on isoja suunnitelmia. 360Editorin lopullinen päämäärä on helpottaa virtuaaliteknologian käyttöönottoa ja tehdä näin virtuaaliteollisuudesta arkipäiväisempää teollisuuden parissa. Sovelluksella on jo nyt aktiivisia päivittäiskäyttäjiä, joista suurin osa on yritysasiakkaita. Voidaan siis todeta, että projekti on lisännyt virtuaaliteollisuuden käyttöä teollisuudessa.

## LÄHTEET

- 360Mediatalo 2019. Referenssit. Viitattu 23.4.2019 <https://www.360mediatalo.fi/referenssit/>.
- 360Mediatalo 2019. Palvelut. Viitattu 23.4.2019 <https://www.360mediatalo.fi/palvelut/>.
- Baptiste, G. How VR Sports Training Can Take Athletes To The Next Level 2018. Viitattu 23.4.2019 <https://provr.io/blog-en/vr-sports-training/>.
- Barchester. Virtual reality offers respite for bedridden people 2017. Viitattu 23.4.2019 <https://www.barchester.com/news/virtual-reality-offers-respite-bedridden-people>.
- Barnard, D. History of VR - Timeline of Events and Tech Development 2018. Viitattu 23.4.2019 <https://virtualspeech.com/blog/history-of-vr>.
- Bohn, D. Intel Made Smart Glasses That Look Normal 2018. Viitattu 23.4.2019 <https://www.theverge.com/2018/2/5/16966530/intel-vaunt-smart-glasses-announced-ar-video>.
- Bonsor, K & Chandler, N. How Augmented Reality Works 2018. Viitattu 23.4.2019 <https://computer.howstuffworks.com/augmented-reality5.htm>.
- Cybershoes 2019. Product. Viitattu 23.4.2019 <https://www.cybershoes.io/product/cybershoes-dk2/>.
- Evans, D. IrisVR Takes Home Technological Innovation of the Year at the 2019 New York City Construction Awards 2019. Viitattu 23.4.2019 <https://blog.irisvr.com/technological-innovation-of-the-year-2019-nycca-awards>.
- Fade, L. How Virtual Reality is Transforming Military Training 2018. Viitattu 23.4.2019 <https://vrvisiongroup.com/how-virtual-reality-is-transforming-military-training/>.
- Feltham, J. Ubisoft Made A VR Experience To Use In Driverless Cars And It Looks Terrifying 2018. Viitattu 23.4.2019 <https://uploadvr.com/ubisoft-made-vr-experience-use-driverless-cars/>.
- Fingas, J. Oculus predicts a VR future that includes ultra-thin headsets 2018. Viitattu 23.4.2019 <https://www.engadget.com/2018/09/26/oculus-predicts-ultra-thin-vr-headsets/>.
- Fulton III, S. What is 5G? All you need to know about the next generation of wireless technology 2019. Viitattu 23.4.2019 <https://www.zdnet.com/article/what-is-5g-everything-you-need-to-know/>.
- Hayden, S. 15 VR Games We Can't Wait to Play in 2019 2018. Viitattu 23.4.2019 <https://www.roadtovr.com/15-vr-games-cant-wait-play-2019/>.
- Krokos, E, Plaisant, C. & Varshney, A. Virtual Reality 2018. Viitattu 23.4.2019 <https://doi.org/10.1007/s10055-018-0346-3>
- Lang, B. To the 'VR is Dying' Crowd: There's More VR Users on Steam Than Ever Before 2018. Viitattu 23.4.2019 <https://www.roadtovr.com/steam-vr-user-population-usage/>.
- Looper, C. The best VR movies: experience the next era of storytelling 2018. Viitattu 23.4.2019 <https://www.techradar.com/news/the-best-vr-movies-experience-the-next-era-of-storytelling>.
- Medium. The Present and the Future of VR Technology in Medicine: successful cases and prospects for development 2018. Viitattu 23.4.2019 <https://medium.com/@AltairVR/the-present-and-the-future-of-vr-technology-in-medicine-12f127317a62>.
- Metz, R. How VR is helping flyers and dental patients calm down 2018. Viitattu 23.4.2019 <https://edition.cnn.com/2018/12/10/tech/vr-dentist/index.html>.

Mundy, J. How will 5G boost VR and AR? 2018. Viitattu 23.4.2019 <https://5g.co.uk/guides/5g-virtual-reality-and-augmented-reality/>.

Node.js. Download the Node.js source code or a pre-built installer for your platform, and start developing today. Viitattu 23.4.2019 <https://nodejs.org/en/download/>.

Oculus. Device Setup - Oculus Go. Viitattu 23.4.2019 <https://developer.oculus.com/documentation/mobilesdk/latest/concepts/mobile-device-setup-go/>.

Oculus. Oculus Unity Getting Started Guide. Viitattu 23.4.2019 <https://developer.oculus.com/documentation/unity/latest/concepts/book-unity-gsg/>.

Scribani, J. What is Extended Reality (XR)? 2019. Viitattu 23.4.2019 <https://www.visualcapitalist.com/extended-reality-xr/>.

Stackable, W. State of the VR Arcade Industry 2018. Viitattu 23.4.2019 <https://medium.com/@springboardVR/state-of-the-vr-arcade-industry-5715e485f668>.

Steed, A. How virtual reality is changing engineering 2017. Viitattu 23.4.2019 <https://www.ingenia.org.uk/Ingenia/Articles/46eb0338-dee8-4322-adae-97f339e0118a>.

Temming, M. Virtual reality therapy has real-life benefits for some mental disorders 2018. Viitattu 23.4.2019 <https://www.sciencenews.org/article/virtual-reality-therapy-has-real-life-benefits-some-mental-disorders>.

Unity. Android environment setup 2018. Viitattu 23.4.2019 <https://docs.unity3d.com/Manual/android-sdksetup.html>.

Virtuix (2019). Products. Viitattu 23.4.2019 <https://www.virtuix.com/products/>.

Walton, B. Netflix Virtual Reality: 15 Movies That Will Blow Your Mind 2017. Viitattu 23.4.2019 <https://screenrant.com/netflix-virtual-reality-vr-movies-best/>.

Willings, A & Tillman M. Windows Mixed Reality: What is it, what headsets are available and which should you buy? 2018. Viitattu 23.4.2019 <https://www.pocket-lint.com/ar-vr/news/microsoft/141215-microsoft-vr-what-is-windows-mixed-reality-and-what-headsets-are-available>.