

MONIALUSTAISEN LAAJENNETUN TODELLISUUDEN SOVELLUKSEN TOTEUTUS

Nico Heinonen



TIIVISTELMÄ

Tampereen ammattikorkeakoulu Tietojenkäsittely Ohjelmistotuotanto

HEINONEN, NICO: Monialustaisen laajennetun todellisuuden sovelluksen toteutus

Opinnäytetyö 52 sivua, joista liitteitä 4 sivua Huhtikuu 2019

Opinnäytetyön tavoitteena oli selvittää, miten lisättyä todellisuutta ja virtuaalitodellisuutta käyttävä sovellus toteutetaan Unity-pelimoottorilla, kun käyttölaitteena on tietokoneen lisäksi Microsoft HoloLens. Työssä tutkittiin AR:n ja VR:n samankaltaisuuksia ja eroja, sekä miten ne otetaan huomioon sovelluskehityksessä. Tutkittiin myös erilaisia tapoja parantaa ja ylläpitää hyvää suorituskykyä optimoimalla 3D-malleja, renderöintiä ja tiedostojen käsittelyä. Lisäksi tutkittiin erilaisten toiminnallisuuksien toteuttamista sekä niiden käytettävyyttä eri laitteilla. Suunnitteluun ja toteuttamiseen käytettiin apuna kvalitatiivista tutkimusta, jossa pääasiassa tarkasteltiin teknisiä dokumentaatioita ja verkkoartikkeleita. Lisäksi tarkasteltiin olemassa olevia AR- ja VR-sovelluksia ja niiden ohella myös tarpeen mukaan pelien erilaisia teknisiä toteutuksia.

Toimeksiantaja oli AR- ja VR-sovelluksiin erikoistunut tamperelainen ohjelmistoyritys Eligo.Studio Oy. Opinnäytetyön tarkoituksena oli tuottaa Valmet Oyj:lle sovellus, jolla pääasiassa tarkastellaan erilaisia tiloja ja laitteita virtuaaliympäristössä. Sovellukseen kehitettiin useita ominaisuuksia, joilla pyrittiin vastaamaan Valmetin ja heidän asiakkaidensa moninaisiin tarpeisiin. Ominaisuudet muun muassa helpottavat navigointia, auttavat asetusten sekä mallien hallinnassa, mahdollistavat reaaliaikaisen laitedatan sisääntuonnin ja sallivat usean käyttäjän yhteistyön verkon välityksellä. Tämä opinnäytetyö myös selittää, miten erilaisiin ratkaisuihin on päädytty ja mitä ne sisältävät sekä tekniseltä kannalta että käyttäjän näkökulmasta.

Sovellus ei opinnäytetyön aikana valmistunut, eikä niin ollut tarkoituskaan, vaan sen kehitys jatkuu edelleen. Kuitenkin prosessin aikana valmistunut sovellus on käyttökelpoinen ja vastaa tämän hetken tarpeisiin. Sekä toimeksiantaja että asiakas ovat olleet tyytyväisiä tuloksiin ja halukkaita jatkokehittämään sovellusta tarpeiden, ideoiden ja teknologian kehittymisen mukaan.

ABSTRACT

Tampereen ammattikorkeakoulu Tampere University of Applied Sciences Degree Programme in Business Information Systems Option of Software Development

HEINONEN, NICO: Development of Multiplatform Mixed Reality Application

Bachelor's thesis 52 pages, appendices 4 pages April 2019

The goal of this thesis was to develop an application with both augmented reality and virtual reality features. The application was developed using Unity game engine and it can be used on Microsoft HoloLens and on a PC with or without VR. This thesis also delves into the similarities and differences between AR and VR and how to account for them in software development. Ways to improve and maintain good performance by optimizing 3D models, rendering and file handling and usability and implementation of features on different devices were some key points in the research. Qualitative research was the main source of information.

The commissioner was Eligo.Studio Oy, a software company specialized in AR and VR. The aim was to develop an application for Valmet Oyj for viewing industrial spaces and machines in a virtual environment. In order to account for Valmet's and their customer's diverse needs, several features were implemented. For each feature there is an explanation detailing the technical aspects, usability and the process leading to the final implementation.

The application did not get completed during working on this thesis. It is still in development as planned and new features are implemented based on needs, ideas and technological advances.

1	JOHDANTO				
2	TE	KNOLOGIAT JA SOVELLUKSET	. 8		
	2.1	Laajennettu todellisuus	. 8		
		2.1.1 Laitteet	. 9		
		2.1.2 Käytettävyys	12		
	2.2	Unity	13		
		2.2.1 Kolmiulotteinen sisältö pelimoottoreissa	14		
		2.2.2 Mixed Reality Toolkit	16		
	2.3	Universal Windows Platform	17		
	2.4	Vrifier	17		
	2.5	PiXYZ Studio	18		
3	KÄ	YTTÖLAITTEET	19		
	3.1	Microsoft HoloLens	19		
		3.1.1 Prosessorit ja muistit	19		
		3.1.2 Näyttö	20		
		3.1.3 Muut osat	21		
		3.1.4 Ohjaus	21		
	3.2	Windows Mixed Reality -lasit	23		
		3.2.1 Virtuaalilasien osat	23		
		3.2.2 Ohjain	24		
		3.2.3 Laitteistovaatimukset	25		
	3.3	Tavallinen tietokonenäyttö	26		
4	SO	VELLUS	27		
	4.1	Suunnittelu	27		
	4.2	Windows 10 alustana	27		
	4.3	Kontrollit	29		
	4.4	Käyttöliittymä	30		
	4.5	Mallit	32		
		4.5.1 Tiedostot	32		
		4.5.2 Mallien optimointi	33		
	4.6	Mittarit	35		
	4.7	Hälytykset	36		
	4.8	Siirtymäpisteet	36		
	4.9	Vaara-alueet	37		
	4.10)Infopaneelit	38		
	4.11	Videot	38		

	4.12 Usean käyttäjän tila	40
	4.13 Mallin koon hienosäätö	42
	4.14 Asetusten konfigurointi	42
5	POHDINTA	45
LÄ	HTEET	47
LI	LIITTEET	
	Liite 1. Mallin konfiguraatiotiedoston rakenne	49
	Liite 2. Tiedoston rakenne, jossa on käyttäjän tekemät lisäykset	50
	Liite 3. Sovelluksen asetustiedosto (1)	51
	Liite 3. Sovelluksen asetustiedosto (2)	52

LYHENTEET JA TERMIT

AR	Augmented reality, lisätty todellisuus
DoF	Degree(s) of freedom, vapausaste, laajennetun todellisuuden
	yhteydessä käytetään esimerkiksi selventämään millaista liik-
	keenseurantaa ohjain tukee
HPU	Holographic processing unit, hologrammiprosessori, jota käy-
	tetään Microsoftin HoloLensessä
IMU	Inertial measurement unit, elektrotinen laite jolla mitataan
	kappaleen liikettä ja kallistumista käyttämällä gyroskooppia,
	kiihtyvyysanturia sekä joskus myös magnetometriä
JSON	JavaScript Object Notation
LCoS	Liquid crystal on silicon, nestekide piillä, mikroprojektiotek-
	niikassa käytetty kuvanmuodostusmenetelmä
MR	Mixed Reality, laajennettu todellisuus
MRTK	Mixed Reality Toolkit, laajennetun todellisuuden laitteita var-
	ten Microsoftin tekemä kirjasto Unity-pelimoottorille
OPCUA	OPC Unified Architecture, teollisuusautomaatiossa laitteiden-
	välisessä viestinnässä käytetty protokolla
Prefab	Unityssa tallennettu peliobjekti, johon on kiinnitetty erilaisia
	komponentteja, kuten skriptejä tai malleja
Scene	Unityn skene, kokoelma peliobjekteja, joka voitaisiin verrata
	esimerkiksi pelin kenttään
SoC	System on a chip, järjestelmäpiiri, esimerkiksi mobiililait-
	teissa ja sulautetuissa järjestelmissä käytetty mikropiiri, joka
	yhdistää monien piirien eri toiminnot
UWP	Universal Windows Platform, nimitys yhdistetylle Windows
	10 alustalle, jolla tehdyt sovellukset toimivat useilla eri lait-
	teilla, kuten tietokoneilla, Xbox One:lla ja HoloLensellä
Verteksi	3D-mallin kulmapiste
VR	Virtual Reality, virtuaalitodellisuus
WMR	Windows Mixed Reality, Microsoftin laajennetun todellisuu-
	den alusta Windows 10 käyttöjärjestelmälle

1 JOHDANTO

Vaikka virtuaalitodellisuus ja lisätty todellisuus eivät ole mitään uusia konsepteja, on niiden saapuminen kuluttaja- ja yritysmarkkinoille kestänyt tovin. Vihdoin aletaan kuitenkin olla siinä pisteessä, että näitä teknologioita voidaan tosissaan hyödyntää sovelluskehityksessä, koska tietokoneet ja älypuhelimet ovat tarpeeksi tehokkaita. Pieniinkin mikropiireihin saadaan valtavasti laskentatehoa, ja juuri sellaisia voidaan käyttää esimerkiksi lisättyä todellisuutta näyttävissä laseissa.

Opinnäytetyön aiheena on laajennetun todellisuuden sovellus, joka on kehitetty Valmet Oyj:lle. Laajennetulla todellisuudella tarkoitetaan tässä tapauksessa sitä, että sovellus hyödyntää lisättyä todellisuutta sekä virtuaalitodellisuutta. Alun perin sovellus oli tarkoitus olla ainoastaan lisätyn todellisuuden sovellus Microsoft HoloLenselle. Sillä olisi tarkkailtu erilaisiin koneisiin, esimerkiksi paperikoneisiin, liittyvää dataa lisätyssä todellisuudessa. Pian kuitenkin todettiin, että sovellukselle voidaan kehittää muitakin käyttötarkoituksia, joten sovelluksen toimintoja laajennettiin tukemaan erilaisten tehdastilojen ja -koneiden tarkastelua virtuaalisessa ympäristössä. Lisäksi Windows Mixed Reality -virtuaalilasit olivat juuri saapuneet markkinoille ja projektissa käytetty Mixed Reality Toolkit -kirjasto oli yhteensopiva niiden kanssa, joten VR-toiminnallisuuden lisääminen oli helppoa.

Opinnäytetyössä perehdytään lyhyesti AR- ja VR-tekniikoiden terminologiaan sekä nykyisiin laitteisiin ja toteutustapoihin. Työssä käydään läpi myös eri teknologiat ja ohjelmat, joita hyödynnettiin kehityksessä ja lisäksi miten monimutkaiset CAD-piirrokset saatiin optimoitua käyttökelpoisiksi 3D-malleiksi. Myös sovellus ja sen toiminnot käydään läpi ja käytetyt ratkaisut selitetään sekä teknologisesta että käyttäjän näkökulmasta.

2 TEKNOLOGIAT JA SOVELLUKSET

2.1 Laajennettu todellisuus

Laajennettu todellisuus on termi, joka voi viitata todellisuuden mihin tahansa muotoon. Paul Milgram ja Fumio Kishino (1994, 3) määrittelivät tutkielmassaan A Taxonomy of Mixed Reality Visual Displays virtuaalisuuden jatkumon, jonka mukaan lisättyyn todellisuuteen kuuluvat lisätty todellisuus, lisätty virtuaalisuus, virtuaalitodellisuus sekä todellinen maailma. Laajennetusta todellisuudesta käytetään usein akronyymia MR, joka tulee englanninkielisestä termistä mixed reality.



KUVA 1. Milgramin jatkumo, A Taxonomy of Mixed Reality Visual Displays

Lisätty todellisuus tarkoittaa todellisuuden muotoa, jossa virtuaalisia asioita tuodaan oikeaan maailmaan. Esimerkiksi puhelimelle tehdyssä sovelluksessa sovellus käyttää puhelimen kameran kuvaa ja piirtää siihen päälle erilaisia virtuaalisia objekteja, kuten kuvia, 3D-malleja ja tekstiä. Tarkoitus on yleensä saada ne istumaan ympäristöön niin kuin ne olisivat oikeasti käyttäjän edessä (Azuma 1997, 2). Lisätyn todellisuuden laitteet ja sovellukset voivat tukea myös liikkumista. Tällöin käyttäjä voi asettaa esimerkiksi virtuaalisen tuolin lattialle ja tarkastella sitä eri kulmista ja etäisyyksiltä liikkumalla ympäristössä. Edistyneemmissä järjestelmissä tila voidaan kartoittaa niin, että saadaan kolmiulotteinen kuva tilasta (Vroegop 2017) ja siten virtuaaliset esineet voidaan asettaa pinnoille, kuten pöydille tai seinille. Kartoittaminen mahdollistaa myös sen, että esineet voivat jäädä oikean maailman esineiden ja ympäristön taakse piiloon. Muutoin kaikki virtuaaliset esineet piirtyvät aina oikean maailman esineiden päälle ja näkyvät esimerkiksi seinien läpi. Lisätystä todellisuudesta käytetään akronyymia AR, joka tulee englanninkielisestä sanasta augmented reality. Virtuaalitodellisuus, VR, tarkoittaa todellisuuden muotoa, jossa käyttäjä kokee immersiivisesti, eli todentuntuisesti, olevansa virtuaalisessa tilassa (Milgram & Kishino 1994, 2). Yleinen VR-laite on virtuaalilasit, mutta on ole-massa myös VR-ympäristöjä, joissa virtuaalitodellisuus on esimerkiksi heijastettu projektoreilla koko tilaan (Cruz-Neira & Sandin & DeFanti & Kenyon & Hart 1992). Immersion tasoa voidaan nostaa myös erilaisilla tekniikoilla, toteutustavoilla ja lisälaitteilla. Jotkin virtuaalilasit tukevat käyjän fyysistä liikkumista, jolloin käyttäjä voi liikkua oikean maailman tilassa ja tämä liike näkyy täysin vastaavana virtuaalimaailmassa. Yleensä tätä tilaa rajoittaa todellisen maailman käyttöalueen fyysinen koko, johtojen pituus tai sensorien laatu. Akronyymi VR tulee sanasta virtual reality.

2.1.1 Laitteet

Laajennettua todellisuutta voidaan käyttää erilaisilla laitteilla. Yhdistävä tekijä näissä on jonkinlainen näyttö tai projektori, jolla tietokonegrafiikka piirretään. Kuluttajamarkkinoilla laitteet ovat pääosin erilaisia virtuaali- tai lisätyn todellisuuden laseja. Useimmissa virtuaalitodellisuuden laseissa on sama toimintaperiaate kuvan näyttämiseen. Niiden sisällä on näyttö, johon piirretään kaksi erillistä kuvaa, vasemman ja oikean silmän kuva. Näin saadaan aikaan stereoskooppinen näkymä, joka saa kuvan näyttämään kolmiulotteiselta (Linowes 2018).

Virtuaalilaseissa näytön ja silmän välissä on myös linssi, jonka avulla saadaan aikaan laajempi näkökenttä. Kuva piirretään vastakkaiseen suuntaan vääristyneenä, ja linssit korjaavat vääristymän ja taittavat valon niin, että se näyttäisi tulevan muualtakin kuin suoraan edestä (Thompson 2018). Linssin halkaisija sekä VR-lasien syvyys vaikuttavat siihen, kuinka laaja katselukulma voidaan saada. VR-laseilla katselukulma on yleensä joko hieman alle tai yli 100°. Vastaavasti, ihmisen näkökentän molempien silmien kattama alue on yhteensä 210° (Cuervo & Chintalapudi & Kotaru 2018, 1). Lisäksi, jopa parhailla tämän hetkisillä laseilla, Vive Prolla ja Samsung Odysseylla, pikselitiheys on noin kuusi kertaa vähemmän kuin mitä täydellisen 20/20 näön omaava ihminen erottaa (Cuervo ym. 2018, 2), joten käyttäjä pystyy mahdollisesti erottamaan yksittäiset pikselit tai kuva saattaa näyttää epätarkalta.

Laite	Ympäristö	Resoluutio	Katselu-	Tukee fyysistä liikku-	Ohjain
			kulma	mista	
HTC Vive	Windows, OSX	2160 x 1200	110°	Base Station lisälait-	2 x 6DoF oh-
		2880 x 1600		teilla	jainta
		(Pro)			
Oculus Rift	Windows, OSX,	2160 x 1200	110°	Oculus Seonsor lisä-	2 x 6DoF oh-
	Linux			laitteella	jainta
Windows Mixed	Windows 10	2880 x 1440	95° - 110°	Kyllä, ilman lisälait-	2x 6DoF oh-
Reality		2880 x 1600		teita	jainta
(Eri valmistajia)					
PlayStation VR	Playstation 4	1920 x 1080	100°	PlayStation Camera li-	PlayStation 4-
				sälaitteella	ohjain
					PlayStation
					Move
Google	Android	Riippuu puheli-	100°	Ei	3DoF ohjain
Daydream View		men mallista			
Samsung	Android (Sam-	Riippuu puheli-	101°	Ei	3DoF ohjain
GearVR	sung puhelimet)	men mallista			
Oculus Go	Android (sisään-	2560 x 1440	110°	Ei	3DoF ohjain
	rakennettu)				

TAULUKKO 1. Erilaiset VR-lasit ja niiden speksit

Lisätyn todellisuuden laseissa kuva näytetään yleensä eri tavalla VR-laseihin verrattuna, koska käyttäjä näkee oikean maailman lasien läpi. Siksi ei voida käyttää vääristynyttä kuvaa ja sitä korjaavia linssejä. Esimerkiksi Microsoft HoloLens käyttää kuvan heijastamiseen nestekide piillä nimistä mikroprojektiotekniikkaa (Colaner 2016). Englanniksi se tunnetaan nimellä liquid crystal on silicon ja akronyymillä LCoS. Koska näyttö on läpinäkyvä, näkyvän sisällön värien valööri vaikuttaa sen näkyvyyteen. Mitä tummempi väri on, sen vähemmän se näkyy ja toisaalta sitä paremmin todellinen maailma erottuu. Lisäksi ympäristön kirkkaus vaikuttaa käytettävyyteen. Jos ympäristön valo on liian kirkas, katoaa kuva näkyvistä osittain tai kokonaan. Hämärässä kuva näkyy paremmin, mutta valon vähäisyys voi heikentää laitteen ympäristön hahmottamista kameran saaman vähäisen valon vuoksi.

Laite	Käyttöjärjestelmä	Näyttö	Järjestelmäpiiri	Muisti
Microsoft HoloLens	Windows 10 Holo-	2536 x 720, 60hz,	Intel Cherry Trail	2 gt
	graphic Edition	35° katselukulma	SoC	
Magic Leap One	Lumin OS		Nvidia Tegra X2	8 gt
			SoC	
Meta 2	Windows 10 (ulkoi-	2560 x 1440, 60hz,	Käyttää tietokonetta	Käyttää tietokonetta
	nen tietokone)	90° katselukulma		

TAULUKKO 2. Erilaiset AR-lasit ja niiden speksit

Nykyään myös monet älypuhelimet pystyvät näyttämään sekä virtuaalitodellisuutta että lisättyä todellisuutta. Virtuaalitodellisuuden katselemista varten tarvitaan jonkinlaiset lasit, joihin puhelin kiinnitetään. Tunnettuja ovat esimerkiksi Google Cardboard sekä Samsung GearVR, mutta markkinoilla on nykyään myös paljon eri valmistajien malleja. Tärkeintä on, että lasit ovat oikean kokoiset puhelimelle. Muuten kuva ei näy oikein käyttäjälle, koska linssit ja puhelimen näyttöön piirretty kuva eivät ole linjassa. Lisätyn todellisuuden näyttämiseen mobiililaitteilla ei vaadita mitään lisälaitteita. Käyttäjä näkee todellisen maailman puhelimen kameran läpi ja sen kuvan päälle piirretään erilaista virtuaalista sisältöä. Koska AR käyttää sekä puhelimen kameraa ja antureita, täytyy niiden olla tarpeeksi tarkat ympäristön tarkkailuun ja käyttäjän liikkeen seuraamiseen (Linowes & Babilinski 2017). Lisäksi useimmiten AR-järjestelmä on myös kalibroitu eri laitemalleille erikseen, jolloin taataan paras mahdollinen toimivuus (Zaun 2003, 2). Tästä voidaan päätellä, että mobiili AR:n saapuminen Android-laitteille on kestänyt Appleen verrattuna kauemmin heterogeenisen laitekannan vuoksi.



KUVA 2. Mobiili VR-näkymä puhelimen näytöllä (developers.google.com)

2.1.2 Käytettävyys

Koska virtuaalisessa ja lisätyssä todellisuudessa käyttöympäristö eroaa usein suuresti perinteisestä tietokoneen tai puhelimen käyttöliittymästä, tuo se aivan erilaisia ongelmia käytettävyyden suunnitteluun. Ongelmakohtia käyttäjälle saattavat olla esimerkiksi ympäristön hahmottaminen, liikkuminen ja mahdollisten käyttöliittymäelementtien ymmärtäminen sekä hyvin perinteisestä poikkeavat ohjaustavat (Sutcliffe & Deol Kaur 2000, 2). Tärkeää onkin alusta asti tavoitella sellaista käytettävyyttä, joka tukee ja ohjaa käyttäjän toimintaa kohti haluttua päämäärää.

Usein pyritään luomaan käyttöliittymiä, jotka eivät riko käyttäjän immersiota. Esimerkiksi jokin virtuaalinen laite voidaan kytkeä päälle laitteen virtapainikkeesta erillisen valikon sijaan. Haluttu immersion taso riippuu myös käyttötarkoituksesta, eikä aina haluta välttämättä tavoitella täysin immersiivistä kokemusta. Kaikissa tilanteissa tämänkaltainen käyttöliittymä ei ole mahdollinen. Tällöin perinteisiä valikoita voidaan näyttää ympäristössä piirtämällä ne käyttäjän eteen tai kohteen viereen.

Yleisesti pidetään hyvänä tapana, ettei mitään käyttöliittymäelementtejä lukita käyttäjän näkökenttään, vaan valikot ja muut elementit leijuvat tilassa (Linowes 2018). Näin käyttäjä voi liikkua niiden ohitse tai katsoa poispäin niistä, eivätkä ne peitä osaa näkökentästä jatkuvasti. Monissa virtuaalisen ja lisätyn todellisuuden sovelluksissa käytetään kaarevia valikoita, jolloin käyttäjän pitää kääntää päätään nähdäkseen käyttöliittymän kaikki elementit. Näin elementeistä saadaan suurempia, jolloin niihin on helpompi osoittaa ohjaimella tai katseella, ja teksti ja kuva pysyvät selkeinä matalillakin resoluutiolla.

Yleinen ongelma virtuaalitodellisuudessa on liikepahoinvointi, toiselta nimeltään matkapahoinvointi. Saarelma (2018) kertoo yleisimmän syyn olevan silmien ja tasapainoelimen antaman informaation ero. Hän mainitsee myös, että pahoinvointia voi ilmaantua virtuaaliympäristössä, jos henkilö on paikallaan, mutta näkökentässä tapahtuu liikettä. Oikeanlaiset liikkumistavat ovat paras keino välttää liikepahoinvoinnin aiheutumista. Yleensä virtuaalista eteenpäin kävelyä tai liukumista pidetään huonompana vaihtoehtona ja useissa sovelluksissa ja peleissä liikutaankin siirtymällä pisteestä toiseen välittömästi teleportaatiolla, mahdollisesti mustan kautta häivyttäen (Linowes 2018). Siirtymistä voidaan kontrolloida ohjaimella tai esimerkiksi rajatummissa järjestelmissä katsomalla tai osoittamalla maailmaan asetettua siirtymäpistettä. Jos laite tukee liikkumisen seurantaa, voi käyttäjä kävellä myös fyysisesti ympäriinsä, yleensä rajatulla alueella, ja siirtymällä kulkea pidempiä matkoja.

Nämä liikkumistavat eivät koske lisättyä todellisuutta, koska käyttäjä liikkuu oikeassa maailmassa ja ympäristöön on lisätty virtuaalisia asioita. Nämä virtuaaliset asiat voidaan tietenkin kohdentaa uudelleen tarpeen ja mahdollisuuksien mukaan. Esimerkiksi AR-lautapeli voidaan asettaa mille tahansa pöydälle tai pinnalle, mutta tiettyyn museoon suunniteltua lisätyn todellisuuden museokierrosta ei voi asettaa muuhun tilaan.

Tärkeä huomioonotettava asia on myös suorituskyvyn ylläpitäminen. Epävakaat ruudunpäivitysnopeudet ja hetkelliset pysähtelyt rikkovat immersion. Hidas ruudunpäivitys aiheuttaa myös liikepahoinvointia, koska käyttäjän pään todellinen liike ei vastaa virtuaalimaailmassa näkyvää liikettä (Linowes 2018). Laitteesta riippuen suositukset minimiruudunpäivitysnopeudelle vaihtelevat 75:stä 90 kuvaan sekunnissa. Hahmottamisen helpottamiseksi voidaan antaa esimerkeiksi seuraavat: tavallisessa videokuvassa kehysnopeus on 24 kuvaa sekunnissa, konsolipeleissä yleensä 30 kuvaa sekunnissa, ja perustason tietokonenäytöllä päivitysnopeus on 60 kuvaa sekunnissa.

Myös lisätyssä todellisuudessa on suositeltavaa pitää ruudunpäivitysnopeus mahdollisimman korkeana, koska muutoin virtuaalinen sisältö saattaa näyttää liikkuvan oikean maailman kuvasta jäljessä, tai koko kuva näyttää hidastetulta tai pätkivältä. Tämä tietenkin huonontaa kokemuksen laatua. Lisäksi sovelluksen hidas suoritus saattaa heikentää ympäristön hahmottamista, jolloin tilaan asetetut virtuaaliobjektit saattavat lähteä liukumaan sijoituspaikoiltaan.

2.2 Unity

Projektin toteutukseen käytettiin Unity-pelimoottoria. Vaikka Unity onkin pääasiassa suunniteltu videopelien tekemiseen, on sillä mahdollista toteuttaa muunkinlaisia sovelluksia. Ohjelmointikielenä Unityssa toimii C#. Aiemmin vaihtoehtoina on ollut myös JavaScript sekä Boo, mutta ne on jätetty pois uudemmista versioista. Projektissa käytettiin myös muutamia kolmannen osapuolen kirjastoja, kuten NewtonSoftin kirjasto JSON-tiedostojen käsittelyä varten sekä Microsoftin Mixed Reality Toolkitiä.

2.2.1 Kolmiulotteinen sisältö pelimoottoreissa

Kolmiulotteiset mallit ovat geometrian termein kuvailtuna monitahokkaita. Digitaalisessa mallinnuksessa ne kuvataan yksinkertaisimmillaan särmien kärkipisteinä, joille ilmoitetaan sijainti kolmiulotteisessa koordinaatistossa. Niiden väliin muodostuu tahkoja ja niistä syntyy mallin pinta. Malleista puhuttaessa tahkoja sanotaan yleensä polygoneiksi tai kolmioiksi tahkoon kuuluvien kulmapisteiden määrästä riippuen.

Yksinkertainen valaistus voidaan laskea Gouraud-varjostusmenetelmällä kärkipisteiden ja valonlähteen suunnan pistetulona (Buss 2003, 77). Tällä ei kuitenkaan saada kovin realistista valaistusta, koska se korostaa kärkipisteitä. Kärkipisteille voidaan myös asettaa normaali, jotka käytetään renderöinnissä valaistuksen laskemiseen. Sen jälkeen valaistus voidaan laskea käyttämällä valonlähteen suunnan ja polygonin kärkipisteistä interpoloidun normaalin pistetuloa (Buss 2003, 78). Näin saadaan aikaan paljon tasaisempi valaistus. Tämä menetelmä tunnetaan nimellä Phong-varjostus ja -valaistus kehittäjänsä Bui Tuong Phongin mukaan. Vaikka ne onkin julkaistu hänen kirjoittamassaan artikkelissa Illumination for Computer Generated Pictures jo vuonna 1975, peligrafiikassa sitä on käytetty ensimmäisen kerran vasta vuonna 1999 Segan Brave Fire Fighters arcade-pelissä (Sega Hikaru Hardware). Menetelmän iästä huolimatta esimerkiksi Id Softwaren perustajan John Carmackin mukaan (2013, 44 min) Phong varjostus- sekä valaistusmalli on edelleen tärkeä osa peligrafiikkaa. Nykyään kuitenkin siihen voidaan liittää paljon monimutkaisempia lisälaskuja, joiden avulla saadaan realistisemmin fysiikkaan perustuva valaistus ja varjostus.



KUVA 3. Sama pallo varjostettuna eri menetelmillä (Wikimedia Commons)

Pelimoottoreissa grafiikka piirretään reaaliajassa ja prosessointiaikaa käytetään myös paljon muuhunkin, kuten esimerkiksi ohjelmakoodin suorittamiseen. Siksi on tärkeää pitää mallit yksinkertaisina. Malleissa ei saa olla liikaa erillisiä kappaleita ja pinnat pyritään pitämään mahdollisimman yksinkertaisina. Grafiikan piirtämisessä käytetäänkin monia erilaisia tekniikoita, esimerkiksi erilaisia tekstuurikarttoja, mallinnustapoja ja shader-ohjelmia yksityiskohtien piirtämiseksi. Esimerkiksi sylinterin ympärys voi koostua vain kahdestatoista sivusta, mutta pelimoottorissa se saadaan kuitenkin näyttämään sileältä, jos vertekseille on laskettu siloitukset joko mallinnusohjelmassa tai pelimoottorin sisällä. Etenkin mobiililaitteilla optimointi on tärkeää, koska useimmissa ei ole erillistä grafiikkaprosessoria. Lisäksi sekä HoloLensellä että virtuaalilaseilla kuva piirretään kahdesti kolmiulotteisuuden luomiseksi, joka tietenkin lisää renderöintiin vaadittavaa suorituskykyä koska piirrettävän sisällön määrä kaksinkertaistuu.



KUVA 4. Kaksi kaksitoistasivuista sylinteriä Unity-pelimoottorin sisällä. Vasemmanpuoleisessa mallissa on laskettu sileä pinta.

Pelimoottoreissa prosessori antaa tiedot piirrettävästä grafiikasta näytönohjaimelle piirtokutsuina, joita englanniksi kutsutaan nimellä draw call. Yksinkertaisimmissa näkymissä kaikki kappaleet voidaan piirtää yhdellä kutsulla. Kuitenkin peleissä usein näkymässä olevat kappaleet käyttävät eri materiaaleja, ja jokainen erilainen materiaali täytyy piirtää eri piirtokutsussa (Jukić 2015). Lyhyesti ilmaistuna materiaali kertoo miltä kappaleen pinta näyttää. Se sisältää tiedon eri tekstuurikartoista, värityksestä sekä pinnan ominaisuuksista, kuten heijastavuudesta. Materiaali myös määrittelee käytetyn shader-ohjelman, joka taas määrittelee millä tavalla malli piirretään ja millaisia laskutoimituksia siihen käytetään. Koska shader-ohjelma suorittaa miljoonia laskuja joka sekunti (Vivo & Lowe 2015), on tärkeää huomioida myös niiden vaikutukset ja valita tilanteeseen sopivat ratkaisut, vaikka ne ovatkin äärimmäisen nopeita myös monimutkaisemmissakin laskutoimituksissa.

Prosessorin on muutettava tiedot grafiikasta näytönohjaimen käsittelemään muotoon, joten suuri määrä kutsuja saattaa johtaa ruuhkautumiseen, jos prosessori käsittelee kutsua kauemmin kuin näytönohjain piirtää edellistä (Jukić 2015). Ja kuten yllä mainittiin, prosessorin tehtävänä on myös monta muuta asiaa, joten tämä hidastaa myös niiden suorittamista. Yleensä optimoinnin kannalta on hyvä pitää materiaalien määrä alhaisena ja kierrättää niitä niin paljon kuin mahdollista. Pelimoottoreissa käytetään myös paljon erilaisia tekniikoita renderöinnin nopeuttamiseksi. Esimerkiksi Unityssa on mahdollista merkitä objektit staattisiksi, jolloin ne eivät voi liikkua tai muuttua suorituksen aikana, koska kaikki samoja materiaaleja käyttävät objektit yhdistetään keskenään.

2.2.2 Mixed Reality Toolkit

Microsoftin tekemä Mixed Reality Toolkit, joka tunnettiin aiemmin nimellä HoloToolkit, on avoimen lähdekoodin projekti, jonka tarkoituksena on helpottaa ja nopeuttaa laajennetun todellisuuden sovellus- ja pelikehitystä Unitylla. Mixed Reality Toolkit, tai lyhyemmin MRTK, tarjoaa rajapintoja, ominaisuuksia sekä valmiita Unity skenejä ja prefabeja joiden avulla kehittäjät voivat pienelläkin vaivalla luoda Windows Mixed Reality sovelluksen tai -pelin. Valmiisiin ominaisuuksiin kuuluvat muun muassa ohjaussyötteen vastaanottaminen, äänikomentojen tulkitseminen, ohjainten piirtäminen, kevyempiä shadereita sekä skriptit Unityn käyttöliittymäelementtien käyttämiseen VR-ohjaimella tai HoloLensellä. Mukana tulee myös eri toimintoihin liittyviä demoja, joista kehittäjät voivat ottaa mallia tai hyödyntää niiden osia projekteissaan. Lisäksi siihen sisältyy tarvittavat koontiskriptit Visual Studio projektin luomista varten. Tästä projektista voidaan sitten tehdä varsinainen WMR-sovellus. Sama sovellus voidaan asentaa tietokoneelle sekä HoloLenselle.

2.3 Universal Windows Platform

Sovellus käyttää Universal Windows Platform -rajapintaa, jonka Microsoft julkaisi Windows 10:n mukana. Rajapinta mahdollistaa yhteen toimivat sovellukset Windows 10, Windows 10 Mobile, Windows 10 IoT, Xbox One sekä HoloLens välillä. Sovellusten kehittämiseen käytetään Visual Studio 2015 tai 2017 ohjelmankehitysympäristöä. Ohjelmointikielinä voidaan käyttää C++, C#, Visual Basic tai JavaScript. Käyttöliittymiä voidaan tehdä XAML, DirectX tai HTML avulla. XAML-käyttöliittymien toteuttamisessa voidaan käyttää myös visuaalista työkalua. Samaa sovellusta voidaan sellaisenaan käyttää kaikilla näillä alustoilla, eikä koodiin vaadita mitään muutoksia. Myös käyttöliittymät toimivat suoraan muillakin alustoilla, jos kehittäjä noudattaa Microsoftin suunnitteluohjeita. UWP sovellukset eivät kuitenkaan toimi vanhemmilla Windows versioilla tai muilla käyttöjärjestelmillä, eikä esimerkiksi Windows 8 Metro applikaatiot ole yhteensopivia UWP:n kanssa ilman muutoksia.

2.4 Vrifier

Teatime Researchin tekemällä Vrifier-työkalulla voidaan sekä katsella että optimoida ja yksinkertaistaa monimutkaisia 3D-malleja. Perinteisen käyttöliittymän sijaan ohjelmaa voi käyttää ainoastaan virtuaalitodellisuudessa. Sovelluksella voidaan tavallisten malliformaattien lisäksi avata myös muun muassa CAD-piirroksia ja muuttaa niitä malleiksi. Projektissa käytettiin Vrifieria mallien yksinkertaistamiseen. Alkuperäisissä malleissa saattoi olla jopa miljoonia polygoneja, joka on todella paljon tehokkaallekin tietokoneelle.

Ohjelman avulla pystyimme luomaan erilliset optimoidut mallit VR- ja HoloLens-versiota varten. Sovelluksessa valitaan tavoitteellinen verteksimäärä optimoidulle mallille. Tämän jälkeen ohjelma pyrkii yksinkertaistamaan mallin niin, että sen muoto säilyy mutta päästään mahdollisimman lähelle haluttua verteksimäärää. Jos verteksimäärä asetetaan liian alhaiseksi, malliin muodostuu paljon epämuodostumia ja artefakteja. Vaikka lopullisessa käytetyssä versiossa ei tälläisiltä virheiltä täysin vältytty, olivat ne silti sen verran pieniä, että se oli parempi vaihtoehto kuin käyttää kymmeniä tunteja mallin käsin korjaamiseen.

2.5 PiXYZ Studio

Myöhemmin mallien optimointiin kokeiltiin myös PiXYZ Softwaren tekemää Studio ohjelmistoa, joka on tarkoitettu CAD-piirrosten muuntamiseen kevyiksi malleiksi. Koska sovellus on suunniteltu nimenomaan näiden piirrosten muuntamiseen malleiksi, löytyy siitä paljon hyödyllisiä toimintoja. Sen tehokkailla työkaluilla ja algoritmeilla mallin optimointi onnistuu vähemmillä virheillä. Lisäksi optimoinnissa auttavat esimerkiksi hierarkiaoptimointi, reikien paikkaus, pienien kappaleiden poistaminen sekä piiloon jäävien kappaleiden poisto. Toisaalta monimutkaisen mallin optimointi voi kestää huomattavasti kauemmin kuin Vrifierillä, jopa useita tunteja. Kaikki toiminnot voidaan konfiguroida tarpeen mukaan ja oikeilla asetuksilla voidaankin nopeuttaa prosessia.

Vaikka testeissä saatiinkin parempia tuloksia nopeammin, ei malleja silti optimoitu uudelleen, koska olemassa olevat versiot olivat riittävän hyviä. Tulevaisuudessa tätä ohjelmaa tullaan kuitenkin käyttämään mallien optimointiin.

3 ΚÄYTTÖLAITTEET

3.1 Microsoft HoloLens

HoloLens on Microsoftin kehittämä, päässä pidettävä AR-laite. Laiteessa olevan järjestelmäpiirin ansiosta se ei tarvitse erillistä tietokonetta. Lasit eivät peitä käyttäjän näkökenttää, vaan hän näkee todellisen maailman lasien läpi. Lasien sisällä on myös näyttö, johon AR-sisältö piirretään. Käyttöjärjestelmänä toimii Windows 10 Holographic Edition, joka vastaa pitkälti ulkonäöltään ja toiminnaltaan tavallista Windows 10 työpöytäversiota, mutta käyttöliittymä on suunniteltu uudelleen HoloLensellä toimivalla tavalla.



KUVA 5. Kuvasarja, joka kertoo miten HoloLens asetetaan päähän. Pään ympäri menevä panta kiristetään ja visiiri käännetään oikealle korkeudelle silmien eteen. (support.microsoft.com)

3.1.1 Prosessorit ja muistit

HoloLensen suorittimena toimii yhden gigahertsin kellotaajuuden 32-bittinen Intelin Cherry Trail -järjestelmäpiiri, joka sisältää prosessorin lisäksi myös näytönohjaimen. Tämän lisäksi HoloLensessä on myös sitä varta vasten suunniteltu ja tehty Microsoft Holographic Processing Unit, josta käytetään akronyymiä HPU. Sen tehtävänä on prosessoida reaaliaikaisesti sensoridataa ja lähettää se järjestelmäpiirille (Angelini 2016). Sensoridatalla voidaan tulkita käyttäjän suorittamia eleohjausliikkeitä, luoda ja päivittää ympäristön kolmiulotteista ymmärrystä sekä tiedostaa mihin suuntaan käyttäjä katsoo. Tätä dataa saadaan Kinect-kameraan pohjautuvasta syvyyskamerasta, joka kattaa 120°×120° alueen (Hempel 2015). Lisäksi käyttäjän pään liikkeitä seuraa sisäinen mittauslaitteisto, joka sisältää kiihtyvyysanturin, gyroskoopin ja magnetometrin. Järjestelmäpiirillä ja HPU:lla molemmilla on yksi gigatavu LPDDR3-muistia, sekä jaettu kahdeksan megatavun SRAM-muisti. Pysyvänä tallennustilana toimii 64 gigatavun flash-muisti.

3.1.2 Näyttö

HoloLensen näyttö ei vastaa perinteistä näyttöä eikä myöskään virtuaalilasien näyttöä. Kuva muodostetaan näytölle kahden projektorin avulla, joita Microsoft kutsuu nimellä light engine (Colaner 2016). Molemmat projektorit muodostavat 16:9 kuvasuhteen teräväpiirtokuvan ja heijastavat sen yhdistimen kautta käyttäjän näkyville. Kuva heijastetaan planaarista aaltojohtoa pitkin, jota voisi verrata laakeaan valokuituun. Lisäksi valo diffraktoidaan jolloin punainen, vihreä ja sininen valo saadaan heijastettua erilisille levyille (Colaner 2016). Koska näytön katselukulma on vain 35°, jää kuva hyvin kapeaksi esimerkiksi virtuaalilaseihin verrattuna.



KUVA 6. Erilliset levyt, joille HoloLensen kuva muodostetaan (tomshardware.com)

3.1.3 Muut osat

HoloLensen akku kestää aktiivisessa käytössä 2–3 tuntia ja valmiustilassa korkeintaan kaksi viikkoa. Laitteesta löytyy Bluetooth 4.1 LE lähetin ja IEEE 802.11ac standardin mukainen Wi-Fi lähetin. Painikkeita löytyy viisi kappaletta: toisen sangan päässä oleva virtapainike sekä sankojen päältä molemmilta puolilta kaksi kappaletta. Vasemman puolen painikkeet säätävät näytön kirkkautta ja oikean puolen painikkeet äänen voimakkuutta. Käyttäjä voi myös ottaa kuvan painamalla molempia äänenvoimakkuuspainikkeita saman aikaisesti tai aloittamalla videon nauhoittamisen pitämällä niitä pohjassa kolme sekuntia. Kuvissa ja videoissa näkyy sekä todellinen maailma että hologrammit.

Laitteessa on myös sivuilla sijaitsevat kaiuttimet. HoloLens pystyy simuloimaan tapaa, jolla ihmiskorva kuulee äänet. Näin ääniä voidaan toistaa niin, että käyttäjä kuulee todentuntuisesti sen tulevan mistä tahansa halutusta suunnasta (Colaner 2016). HoloLensessä on myös sarja mikrofoneja puheen tunnistusta, nauhoitusta ja muuta mahdollista äänikommunikointia varten.

3.1.4 Ohjaus

HoloLensellä osoittimena toimii näkökentän keskellä näkyvä kursori. Kursorin muoto voi vaihdella ohjelmasta riippuen, mutta Windowsissa se on normaalisti valkoinen lieriö. Sen muoto vaihtelee tietyissä tilanteissa (Vroegop 2017). Kun laite tunnistaa käyttäjän käden näpäytysvalmiudessa, se aukeaa renkaaksi. Taas kohdissa, joissa käyttäjä voi vierittää valikkoa ylös tai alas näkyy kursori, jossa on nuoli ylös ja alas.

Joissakin tapauksissa valintoja voidaan myös tehdä katseella. Silloin käyttäjän tulee katsoa jotakin kohdetta, esimerkiksi painiketta, tietyn aikaa, että valinta tapahtuu. Tästä indikaattorina on jonkinlainen mittari, joka kertoo kauanko kohdetta pitää katsoa. Mittari voi olla vaikkapa kursorin ympärillä asteittain täyttyvä rengas.

Hiiren klikkauksia ja raahaamista vastaavat eleet käyttäjät tekevät pääasiassa kädellään. Jotta laite tunnistaisi eleet, on kättä pidettävä sopivalla korkeudella ja etäisyydellä laitteen edessä (Vroegop 2017). Eleitä voi tehdä sekä oikealla, että vasemmalla kädellä, mutta vain yhdellä kädellä samanaikaisesti. Klikkaamiseen voi käyttää myös HoloLens clicker -ohjainta. Se on pieni kädessä pidettävä yhdestä painikkeesta koostuva langaton ohjain. Se asetetaan etu- ja keskisormen päälle ja painiketta painetaan peukalolla. Eleitä ei kuitenkaan voi suoraan tehdä ohjaimella, joten sovelluksen pitää tukea niitä esimerkiksi erillisillä eleitä vastaavilla painikkeilla, jos sitä halutaan käyttää yksinomaan clicker-ohjaimella.



KUVA 7. HoloLens clicker -ohjain (support.microsoft.com)

HoloLens tunnistaa kolme pääelettä: näpäytyksen, pitkän painalluksen ja vetämisen. Näpäytys vastaa useimmissa sovelluksissa hiiren vasemman painikkeen klikkaamista vastaavia toimintoja, eli yleensä sillä tehdään valintoja. Näpäytys tehdään nostamalla käsi eteen niin, että se on vähintään noin kyynärvarren mitan päässä käyttäjän edessä hieman katseen alapuolella kämmenselkä käyttäjään päin. Sormien on oltava niin, että etusormi ja peukalo erottuvat selkeästi käyttäjästä nähden ja muut sormet ovat nyrkissä. Näpäytyksen tehdäkseen käyttäjä painaa etusormen ja peukalon yhteen ja palauttaa ne aiempaan asentoon. Sormet täytyy myös pitää suorana, eli ele ei siis saa muistuttaa nipistämistä. Jos näpäytys tunnistetaan, kursori painuu kasaan renkaasta lieriöksi, kun sormet ovat yhdessä.

Samalla periaatteella toimii myös pitkä painallus sekä erilaiset vetämisliikkeet. Erona on se, että etusormi ja peukalo pidetään yhteen painettuina niin kauan kuin on tarpeen liikkeestä riippuen. Pitkän painalluksen vaadittua kestoa voi esittää mittarilla tai jollakin muulla käyttöliittymän elementillä. Vetämisliikkeitä, kuten vieritystä tai kolmiulotteista siirtämistä tehdessä, sormet pidetään painettuina yhteen niin kauan kuin liikettä halutaan jatkaa.

3.2 Windows Mixed Reality -lasit

Sovellusta voi käyttää VR-tilassa ainoastaan WMR-yhteensopivilla virtuaalilaseilla. Näistä laseista käytetään usein nimitystä Windows Mixed Reality headset. WMR valittiin, koska sovellus oli alun perin toteutettu HoloLensille käyttäen Mixed Reality Toolkit -kirjastoa. Sama kirjasto mahdollistaa myös käytön edellä mainituilla VR-laseilla. Mikään ei estäisi sovelluksen käyttöä muun tyyppisillä laseilla, mutta niitä varten pitäisi tehdä muutoksia sovelluksen osiin, jotka on rakennettu MRTK-kirjaston toimintoja käyttäen.

Kilpailijoiden VR-laitteistosta poiketen WMR tukevia laseja valmistaa useampi valmistaja. Muun muassa Acer, Dell, HP, Lenovo sekä Samsung valmistavat niitä. Kaikki virtuaalitodellisuuslasit ovat pääosin samanlaisia spekseiltään, mutta pieniä eroja löytyy. Esimerkiksi muihin malleihin verrattuna Samsung Odysseyn näytössä on suurempi tarkkuus ja se käyttää AMOLED-teknologiaa LCD:n sijaan. Laitteiden hinnat olivat myyntiin tullessaan 400 euron ja 600 euron välissä. Sittemmin laitteiden hinnat ovat tippuneet jopa alle 300 euron. Ne ovat siis kalleimmillaankin olleet noin puolet halvempia kuin Oculus Rift ja HTC Vive. Sovelluksen kehityksessä on käytetty Acerin ja Samsungin valmistamia laseja.

3.2.1 Virtuaalilasien osat

Rakenteeltaan Windows Mixed Reality virtuaalitodellisuuslasit ovat pääosin samanlaiset kuin markkinoilla tällä hetkellä olevat muutkin lasit. Yksi suurimmista eroista on kuitenkin käyttäjän liikkeen seuraamiseen käytetty tekniikka. Erillisten huoneeseen asetettavien sensoreiden sijaan lasit käyttävät tekniikkaa, jota Microsoft kutsuu nimellä inside-out tracking. Virtuaalilasien etuosassa on kaksi matalaresoluutioista mustavalkokameraa, jotka tarkkailevat tunnusmerkkejä ympäristössä (Aaron & Etienne & Zeller & Wojcia-kowski 2017). Kun tätä dataa käytetään yhdessä nopeataajuuksisen IMU-datan kanssa, pystytään tarkasti määrittelemään laitteen sijainti ympäristössä. Kameran käyttäminen mahdollistaa myös tilojen muistaminen, jolloin eri huoneissa voidaan ladata sinne ase-tettu käyttöaluerajaus. Toisaalta kamera myös rajoittaa käyttömahdollisuuksia, koska kamera ei voi tarkkailla ympäristöä, jos valoa ei ole tarpeeksi.

3.2.2 Ohjain

Virtuaalilasien kanssa voidaan myös käyttää yhtä tai kahta liikeohjainta. Ohjaimissa on useita valoja, joita järjestelmä tarkkailee virtuaalilasien sensoreiden avulla (Aaron & Etienne & Guyman & Zeller 2017). Lisäksi ohjaimissa on myös liiketunnistin, joka auttaa seuraamista, kun ohjain on pois kameran näkyvistä. Jos ohjain kuitenkin pysyy pitkään pois kameran näkökentästä, saattaa sen seuranta pettää. Asia kuitenkin korjaantuu heti kun kamera havaitsee ohjaimen seuraavan kerran.

Ohjaimessa on neljä näppäintä, sauvaohjain sekä kosketuslevy. Ohjaimen takaa löytyy liipaisinpainike, johon käyttäjän etusormi asettuu. Sitä käytetään yleensä hiiren vasenta klikkausta vastaavana toimintona, eli valitsemiseen (Aaron ym. 2017). Varressa on tarttumispainike, jota käyttäjä painaa keskisormellaan. Se on suunniteltu emuloimaan tarttumista, eli käyttäjä voisi esimerkiksi nostaa tavaran virtuaalitodellisuudessa ojentamalla kätensä sen kohdalle ja tarttumalla pitämällä painiketta pohjassa ja tiputtamalla sen vapauttamalla painikkeen. Sauvaohjaimen ja kosketuslevyn välissä on valikkopainike ja ohjaimen varressa on Windows-painike. Sauvaohjain toimii samalla tavalla kuin peliohjaimen sauvaohjain. Se antaa x- ja y-koordinaatin, joka kertoo sauvan asennon. Kosketuslevy antaa samanlaisen koordinaatin, kun käyttäjä laskee sormen sen päälle kertoen sormen sijainnin levyn päällä. Painettaessa se rekisteröi ainoastaan, että sitä on painettu. Kehittäjä joutuu siis erikseen hakemaan koordinaatin, jos sitä halutaan käyttää vaikkapa ristiohjaimen tapaan. Windows-painike on ainoa, jonka toimintoa kehittäjä ei pysty muuttamaan.



KUVA 8. Windows Mixed Reality VR-ohjain (docs.microsoft.com)

3.2.3 Laitteistovaatimukset

WMR-virtuaalilasien käyttämiseen tarvitaan Windows 10 tietokone, jossa on Windows 10 Fall Creators -päivitys tai uudempi. Vapaata tallennustilaa vaaditaan 10 gigatavua Windows Mixed Reality -ohjelman asennusta varten. Lisäksi koneessa pitää olla myös Bluetooth 4.0 -yhteys, koska ohjaimet yhdistetään tietokoneeseen sitä käyttäen.

	Windows Mixed Reality	Windows Mixed Reality Ultra
Prosessori	Intel Core i5 7200U, dual core,	Intel Core i5 4590, quad core
	Hyper Threading	
	AMD Ryzen 5 1400 3,4 ghz,	AMD Ryzen 5 1400 3,4 ghz,
	quad core	quad core
RAM	8 gt DDR3 dual channel	8 gt DDR3
Näytönohjain	Integrated Intel HD Graphics 620	Nvidia GeForce GTX 1060
	Nvidia MX150	AMD Radeon RX 470/570
	Nvidia GeForce GTX 1050	
	Nvidia 965M	
	AMD Radeon RX 460/560	
Portit	HDMI 1.4 tai DisplayPort 1.2	HDMI 2.0 tai DisplayPort 1.2
	USB 3.0 Type-A tai Type-C	USB 3.0 Type-A tai Type-C
Muuta	Bluetooth 4.0	Bluetooth 4.0

TAULUKKO 3: WMR laitteistovaatimukset (Windows Mixed Reality PC hardware guidelines)

Lisäksi käyttämiseen tarvitaan myös Mixed Reality Portal ohjelmisto. Sitä kautta on mahdollista yhdistää ja kalibroida VR-lasit sekä määrittää käyttöalueen, jos sitä halutaan käyttää. Ohjelma tuo mukanaan asetusten lisäksi myös myös Windows Mixed Reality home -käyttöliittymän, jonka avulla Windowsia voi käyttää VR-tilassa.

3.3 Tavallinen tietokonenäyttö

Myöhemmin mahdollistettiin myös sovelluksen käayttö tavallisella tietokonenäytöllä. WMR-sovellukset toimivat muutenkin jo Windows 10 tietokoneilla ainoastaan eroten siinä, miten niitä käytetään. Käytännössä siis toteutettiin ainoastaan näppäimistöä ja hiirtä käyttävä ohjaustapa. Sovelluksen käyttöliittymästä ei kuitenkaan ole toistaiseksi tehty erillistä versiota vaan se näyttää täsmälleen samalta kuin AR- ja VR-tilassa.

4 SOVELLUS

Sovellus on tehty Valmet Oyj:lle Eligo.Studio Oy:n toimesta ja sitä kutsutaan nimellä Valmet XR. Sovelluksella useampia käyttötarkoituksia: mittaridatan näyttäminen AR-tilassa, etätuen antaminen ja vastaanottaminen, kollaboratiivisena suunnitteluympäristönä toimiminen sekä myynnin edistäminen. Tällä hetkellä sitä on käytetty lähinnä kahteen jälkimmäiseen, joista enemmän myyntiesittelyihin.

Sovelluksella on mahdollista tarkastella kolmiulotteisia malleja erilaisista koneista ja tiloista lisätyssä todellisuudessa, virtuaalitodellisuudessa sekä perinteisesti tietokoneen näytöltä. Eniten sitä käytetään erilaisten paperikoneiden tarkasteluun.

4.1 Suunnittelu

Sovelluksen suunnittelu ja toteutus aloitettiin toukokuussa 2017. Alkuperäinen suunniteltu käyttötarkoitus oli asetella oikeiden tehdaskoneiden ja vastaavien laitteiden yhteyteen HoloLensellä näkyviä mittareita ja infotauluja. Niissä olisi näytetty oikeaa kohteesta tulevaa dataa. Näin tiettyjä kriittisiä arvoja olisi helppo seurata. Toinen käyttötarkoitus oli etätuen antaminen. Paikan päällä oleva, apua tarvitseva käyttäjä olisi ottanut yhteyden HoloLensellä etänä apua tarjoavaan henkilöön. Tämä henkilö olisi käyttänyt VR-laseja kentällä olevan käyttäjän avustamiseen nähden virtuaalisen version laitteesta tai tilasta. Lisäksi käyttäjillä olisi ollut puheyhteys ja VR-käyttäjän olisi ollut myös mahdollista nähdä HoloLens-käyttäjän kameran kuva. Pian kuitenkin todettiin, ettei HoloLens täytä kaikkia turvallisuusvaatimuksia. Sen kanssa ei pysty välttämättä pitämään suojakypärää ja se peittää näkökenttää liikaa. Etätukitoiminto ei kuitenkaan ole poissuljettu ja periaatteessa sovellus pystyy siihen tälläkin hetkellä. Sen kehittäminen on kuitenkin jäänyt takaalalle.

4.2 Windows 10 alustana

Koska kyseessä on Windows Mixed Reality -toiminnallisuuksia käyttävä sovellus, täytyy se toteuttaa Universal Windows Platform -rajapintoja käyttäen, eli sovellus toimii vain

Windows 10 -käyttöjärjestelmällä. Sovelluksen Unity-puolen toteutukseen tämä ei vaikuta, mutta koontiversiot tehdään hieman eri tavalla verrattuna vaikkapa x86-sovellukseen. Sen sijaan, että Unitysta saataisiin pihalle EXE-tiedosto sekä muut tarvittavat tiedostot, kootaankin sovelluksesta Visual Studio -projekti. Tästä projektista voidaan sitten tehdä paketti joko Windows 10 -kauppaan tai manuaalisesti Power Shellin avulla asennettava ohjelmapaketti. Tässä projektissa on käytetty jälkimmäistä vaihtoehtoa.

Visual Studio -projektin kokoamiseen Unityssa käytetään hieman muokattua koontikanavaa. Näin voidaan käyttää pohjana muokattua projektia, johon lisätty aloitusnäkymä sekä asetustoiminnallisuus. Aloitusnäkymässä voidaan valita, että käynnistetäänkö sovellus immersiivisesti, eli VR tilassa, vai 2D-tilassa, eli näyttöä käyttäen. Lisäksi näkymästä pääsee asetuksiin ja siinä voi selata Valmetin verkkosivua. Verkkosivun näkymiselle on olemassa myös tekninen selitys. Samaa *WebView*-näkymää käytetään sovelluksen sisällä verkkosivujen näyttämiseen, koska Unityssa ei ole sisäänrakennettuna tätä toimintoa.



KUVA 9. Sovelluksen aloitusnäkymä

4.3 Kontrollit

VR-tilassa liikkumiseen käytetään teleportaatiota ja lyhyitä askeleita. Käyttäjä työntää ohjaimen analogista sauvaohjainta eteenpäin, jolloin näkyviin ilmestyy kaareva viiva ja sen päässä oleva vaakasuuntainen rengas ja siihen ylhäältäpäin osoittava nuoli. Kun käyttäjä päästää sauvan takaisin neutraaliasentoon, kuva häivytetään nopeasti mustaksi ja takaisin näkyviin. Tässä välissä käyttäjä on siirretty pisteeseen, jossa nuoli ja rengas olivat. Käyttäjä voi siirtyä myös ylöspäin irti maanpinnasta. Täten käyttäjä voi tarkastella ympäristöä kulmasta, joka ei todellisessa maailmassa ole mahdollista, tai siirtyä nopeasti esimerkiksi ylempään kerrokseen. Sovelluksessa on mahdollista myös kääntyä paikallaan kääntämällä sauvaa vasemmalle tai oikealla sekä hypätä lyhyen matkan taaksepäin kääntämällä sauvaa alas. Kosketuslevyä käyttämällä voi siirtyä lyhyitä matkoja eteen, taakse tai sivuttain. Tämä helpottaa huomattavasti tiettyyn kohtaan liikkumista pelkkään teleportaatioon verrattuna. Sauvaohjaimella liikkumiseen käytettiin MRTK:n mukana tullutta skriptiä, mutta siihen lisättiin itse kosketuslevyllä liikkuminen.

Painikkeiden painaminen ja muut interaktiot suoritetaan osoittamalla ohjaimella haluttua kohdetta ja painamalla ohjaimen liipaisinta. Käyttäjä näkee ohjaimesta lähtevän viivan, ikään kuin laserosoittimen, joka auttaa hahmottamaan ohjaimen osoittaman suunnan. Ohjaimen osoittamassa suunnassa on renkaan muotoinen kursori, joka asettuu interaktiivisen kohteen päälle käyttäjän osoittaessa sellaista. Koodissa klikkauksia voidaan ottaa vastaan helposti käyttämällä *IInputClickHandler*-rajapintaluokkaa ja implementoimalla *OnInput-Clicked*-metodin. Kehittäjän ei siis itse tarvitse huolehtia siitä, että kursori tai ohjaimen säde osuvat klikattavaan kohteeseen, vaan MRTK:n omat skriptit hoitavat kaiken sen taustalla. Kuitankin vaadittavat prefabit syötteiden lukemiseen täytyy lisätä Unity-sekeneen.

Tavallista näyttöä käytettäessä ohjaaminen tapahtuu näppäimistöllä ja hiirellä tavalla, joka vastaa ensimmäisen persoonan pelikontrolleja. Näppäimillä W, A, S ja D käyttäjä liikkuu eri suuntiin. Suunta määräytyy sen mukaan mihin käyttäjä katsoo, eli eteenpäin on aina eteenpäin katsomissuunnan mukaan. Eteen- ja taaksepäin liikkuminen tapahtuu W ja D näppäimillä ja sivuttain liikkuminen A ja D näppäimillä. Käyttäjä voi myös nousta ylöspäin ja laskeutua alaspäin Q ja E näppäimillä. Hiirellä käyttäjä ohjaa kameraa, eli suuntaa, johon hän katsoo sovelluksessa. Osoitin, joka VR-tilassa olisi ohjaimella osoitetussa paikassa on nyt aina ruudun keskellä. Hiiren vasemmalla painikkeella on sama toiminnallisuus kuin ohjaimen liipaisinpainikkeella. Hiiren oikealla painikkeella käyttäjä voi lähentää kuvaa, joka auttaa tarkastelemaan kaukana olevia kohteita. Toiston välttämiseksi alempana toiminnoissa mainitaan vain VR-kontrollit, koska ne voi yllämainitun mukaan muuntaa toiseen muotoon.

HoloLensellä ei tietenkään ole minkäänlaista muuta liikkumistapaa, kuin käyttäjän fyysinen liikkuminen. Interaktioihin käyttäjä käyttää samaa kursoria kuin VR-tilassa, mutta nyt kursori pysyy koko ajan näkymän keskellä. Klikatakseen asioita käyttäjä osoittaa kohdetta kääntämällä päätään ja tekee näpäytyseleen tai painaa clicker-ohjaimen painiketta, jos se on käytössä. Muutoin kaikki toiminnallisuus on samanlaista lukuun ottamatta muutamia toimintoja, jotka eivät ole käytettävissä HoloLensellä. Nämä mainitaan erikseen alempana kyseisten toimintojen yhteydessä.

4.4 Käyttöliittymä

Suunniteltaessa tarkoituksena oli luoda sellainen käyttöliittymä, jota voidaan käyttää sekä HoloLensellä että VR-laseilla. Tavoitteena oli myös saada siitä tarpeeksi yksinkertainen, koska käyttötapausten kirjo on laaja eikä kaikilla käyttäjillä ole välttämättä paljoa kokemusta teknologisten laitteiden käytöstä saatikka sitten AR- tai VR-laseista. Koska käyttäminen tavallisella tietokonenäytöllä mahdollistettiin vasta myöhemmin, sitä varten ei tehty muutoksia käyttöliittymään, vaikka esimerkiksi ikkunat olisi voitu tuoda ruudulle perinteisessä kaksiulotteisessa muodossa.



KUVA 10: Työkaluvyö

Ohjelman kaikkiin perustoimintoihin pääsee käsiksi noin käyttäjän vyötärön korkeudella näkyviä painikkeita käyttämällä. Tätä painikkeiden kokoelmaa kutsutaan myös nimellä toolbelt, suomeksi siis työkaluvyö. Painikkeet on asetettu rengasmaiseen muodostelmaan, joten ne asettuvat ikään kuin ympyrän kehälle käyttäjän eteen. Ne seuraavat koko ajan käyttäjän mukana ja käyttäjä näkee ne katsoessaan alaspäin. Näin painikkeet eivät peitä käyttäjän näkökenttää, ellei hän katso alaviistoon. Koska painikkeita on paljon, on asetusten kautta mahdollista luoda omia painikekokonaisuuksia, joiden välillä voi vaihtaa sovelluksen sisällä. Osa painikkeista on lukittu ja tarvitsee muokkaustilan avaamisen toimiakseen. Muokkaustila otetaan käyttöön painamalla keltaista lukkopainiketta ja näppäilemällä PIN-koodi painikkeen yläpuolelle ilmestyvällä numeronäppäimistöllä. Työkaluvyön voi myös piilottaa, jolloin näkyviin jää vain yksi painike, jolla sen saa takaisin näkyviin. Työkaluvyön alareunassa näkyy teksti, joka ilmoittaa OPCUA-yhteyden tilan. Koska käyttäjät voivat itse luoda painikekokonaisuuksia, täytyy painikkeiden sijainti laskea dynaamisesti. Painikkeen malli on luotu ympyrän kehästä, joten yhden painikkeen leveys voidaan laskea asteina. Yksi painike on 23 astetta ja sen keskipiste on ympyrän keskipisteessä painikkeen fyysisen keskipisteen sijaan, joten painikkeiden sijainti voidaan laskea helposti seuraavalla koodilla.

```
public Transform[] buttons;
public float buttonRotation = 23f;
public float rotationOffset = 0f;
public float tiltDegrees = 25f;
public void PositionButtons()
{
    int activeCount = buttons.Count(X => X.gameObject.activeSelf);
    float startRotation = 180f + rotationOffset - buttonRotation / 2f;
    startRotation -= (activeCount * buttonRotation) / 2f;
    for (int i = 0; i < buttons.Length; i++)</pre>
    {
        if (buttons[i].gameObject.activeSelf)
        {
            buttons[i].transform.localPosition = Vector3.zero;
            startRotation += buttonRotation;
            buttons[i].localRotation = Quaternion.Euler(tiltDegrees, startRotation, 0);
        }
    }
}
```

Jotkin painikkeista avaavat ikkunoita, joissa on joko tietoa, asetuksia tai lisätoimintoja. Nämä ikkunat pysyvät käyttäjän näkökentässä ja seuraavat perässä hänen liikkuessaan ja kääntyessään, koska niitä ei ole tarkoitus jättää auki myöhempää käyttöä varten. Ikkunat eivät kuitenkaan jää keskelle näkökenttää, joten niistä on mahdollista katsoa hieman ohitse.

Projektin toteutuksessa ei ole tässä vaiheessa vielä panostettu ulkoasuun, joten käyttöliittymän graafinen tyyli on hyvin pelkistetty. Suurimmaksi osaksi grafiikat ovat Unityn tarjoamia käyttöliittymägrafiikoita, mutta esimerkiksi työkaluvyön painikkeet ovat 3D-malleja. Ne tehtiin kolmiulotteisena, että niistä saataisiin kaarevia. Värien ja kirjasimien osalta on käytetty Valmetin tyyliohjetta.

4.5 Mallit

Projektissa käytetyt mallit erilaisille tiloille ja koneille on luotu CAD-piirrosten pohjalta. Piirrokset ovat insinöörien tekemiä suunnitelmia, jotka ovat todella yksityiskohtaisia, jotta kyseinen kohde voidaan rakentaa niiden pohjalta. Kun nämä piirrokset muutetaan 3D-malleiksi, niihin tulee miljoonia verteksejä ja polygoneja. Tämänlainen malli on liian monimutkainen käsiteltäväksi jopa todella tehokkaalle tietokoneelle. Siksi mallit täytyykin optimoida ennen käyttöönottoa. Malleista voidaan tehdä sekä korkean että matalan tarkkuuden versiot, joista ensimmäistä käytetään PC:llä ja jälkimmäistä HoloLensellä.

4.5.1 Tiedostot

Itse mallitiedostojen lisäksi jokaiseen malliin liittyy yksi tai useampi asetustiedosto. Yksi tärkeimmistä on avain-arvo-pareista koostuva JSON-tiedosto (Liite 1), joka sisältää tiedot muun muassa mallin selkokielisestä nimestä, malliin liittyvistä tiedostoista sekä sijainnin, kulman ja skaalan, joita käytetään mallin asemointiin sovelluksen sisällä. Jos käyttäjä on tehnyt jotain lisäyksiä sovelluksen sisällä, kuten siirtymäpisteitä, on myös niiden tallentamista varten luotu erillinen tiedosto (Liite 2). Malliin voi liittyä myös tiedosto, jossa on lista OPCUA-yhteydessä käytettävistä viitteistä. Näiden avulla tiedetään mitä mittaridataa oikean maailman kohteeseen liittyy ja siten käyttäjä voi valita mitä tietoja hän haluaa

näytettäväksi sovelluksessa. Nämä tiedostot säilytetään Unityn määrittelemässä kansiossa. Kansion sijainti riippuu kohdealustasta. UWP-alustalla se löytyy *LocalState*-kansiosta samasta paikasta, minne sovelluspaketti on asentunut Windows-käyttäjän *AppData*kansiossa. Kaikki nämä tiedostot voidaan myös kopioida koneelta toiselle helposti joko käsin tai tarkoitusta varten toteutetun toiminnallisuuden avulla.

Tallennettavat tiedot riippuvat kyseistä kohdetta kuvaavan luokan muuttujista ja niille asetetuista sarjoitusrajoituksista. Sarjoituksella tarkoitetaan esimerkiksi jonkin olion tietojen tallentamista toiseen muotoon, josta ne voidaan myöhemmin lukea ja muuttaa takaisin samaksi olioksi. JSON-muotoon tallentamiseen käytetään Newtonsoftin JSON.net kirjastoa. Tärkeänä seikkana mainittakoon, että sijainnit ja rotaatiot tallennetaan aina mallin avaruudessa. Tämä johtuu siitä, että mallit voidaan ladata skeneen eri sijainteihin ja kokoihin. Näin pisteille voidaan laskea oikea sijainti käyttämällä Unityn tarjoamaa *Transform*-luokan *TransformPoint*-metodia. Rotaatio taas saadaan kertomalla mallin rotaation kvaternio tallennetun rotaation kvaterniolla.

4.5.2 Mallien optimointi

Aluksi mallien geometriaa optimoitiin Vrifier-työkalulla. Tällä työkalulla mallit saatiin tarpeeksi yksikertaisiksi, että niitä voitiin katsella tietokoneella VR versiossa ilman suurempia ongelmia. Kuitenkaan HoloLenselle ei saatu toimivaa versiota niin helpolla. Ongelma oli etenkin HoloLensen vähäinen muisti ja heikko suorituskyky. Vaikka sille tehtiin VR-versiotakin yksinkertaisemmat versiot malleista, olivat ne silti liian monimutkaisia.

Useimmiten Unityssa 3D-mallit ja muukin sisältö on lisätty projektiin ennen sovelluksen koonnin tekemistä. Tässä tapauksessa kuitenkin tarvittiin ratkaisu, jolla malleja voitiin ladata suorituksen aikana. Käytimme myös Teatime Researchin tekemää kirjastoa, joka pystyy avaamaan OBJ-malleja ja muuttamaan niitä Unityn käyttämään muotoon ajon aikana. Koska OBJ-mallit ovat tekstimuotoisia, eivätkä esimerkiksi binäärimuotoisia, ovat ne tiedostokooltaan valtavia, jopa satoja megatavuja. Lisäksi roskankeruu ei poistanut näitä tietoja muistista. Kun malli vielä ladattiin näkyviin ruudulle, kului muistia lisää. Etenkin jos vaihtoi useamman kerran mallista toiseen peräkkäin, muisti loppui kesken ja sovellus kaatui. Mallit päätettiin lopulta tallentaa käyttäjän laitteelle binäärimuodossa, josta ne on helppo lukea takaisin Unityn käyttämään malliformaattiin. Kun malli on muutettu ensin Unityn malliformaattiin tallennetaan se sen jälkeen *LocalState*-kansioon. Tiedostojen kirjoittamiseen ja lukemiseen käytetään itse toteutettua ratkaisua, koska näin säästetään muistia sekä aikaa verrattuna esimerkiksi .NET kirjaston vastaaviin toteutuksiin. Tiedostojen formaatti on myös kehitetty itse ja se perustuu Unityn käyttämän *Mesh*-luokkaan sisältöön. Otsikkolohkossa on lueteltu versio, Unityn käyttämä bittisyys indekseille sekä verteksien, normaalien, mallien ja kolmioiden lukumäärä. Nämä lukumäärät ovat tärkeitä, koska niiden avulla määritellään mikä osa datasta luetaan kutakin taulukkoa luodessa. Tiedot kirjoitetaan ja luetaan käyttämällä *System.IO*-kirjaston *BinaryReader*- ja *BinaryWriter*-luokkia. Lisäksi tiedosto pakataan käyttämällä gzip-formaattia. Näin luotujen tiedostojen koko on huomattavasti pienempi. Ne ovat kooltaan vain noin 10% alkuperäisestä Vrifiertyökalusta tallennetusta tekstimuotoisista OBJ-tiedostosta. Tiedostot käsitellään myös omassa säikeessä, jolloin ohjelman suoritus ei jäädy tiedostoja ladattaessa.

Verteksit		
Komponentti	Тууррі	Muistissa (tavua)
Sijainti	float[3]	12
Normaali	float[3]	12
Tekstuurikoordinaatti	float[2]	8
Väri	float[4]	16
Yhteensä	48	
Kolmiot		
Taulukko, jokaisella kolmiolla kolme arvoa	int[]	6

TAULUKKO 4. Unityn mallidatan kuvaus ja muistin käyttö

Kaikki mallin materiaalit korvataan tallennuksen yhteydessä yhdellä materiaalilla. Alkuperäisissä malleissa on usein kymmeniä materiaaleja ja jokainen materiaali aiheuttaa uuden piirtokutsun. Suorituskyvyn kannalta on tärkeää minimoida kutsujen määrä, varsinkin stereoskooppisessa renderöinnissä, jossa kuva piirretään aina kahdesti. Jotta värit saataisiin näkyviin, materiaalien värit tallennetaan verteksin väriarvoiksi ja näytetään käyttämällä itse toteutettua shader-ratkaisua. Sama shader-ohjelma hoitaa myös mallin varjostamisen kevyellä laskukaavalla, joka laskee varjostuksen tummuuden verteksin normaalin ja asetetun valon suunnan pistetulon avulla. Aiemmin ei ollut mahdollisuutta käyttää minkäänlaista varjostusta malleissa, koska mallit ovat niin monimutkaisia, että Unityn laskema valaistus vie liikaa tehoja. Shader-ratkaisussa käytettiin seuraavaa koodia.

```
v2f vert2 (appdata_full v)
{
   v2f o;
   o.pos = UnityObjectToClipPos(v.vertex);
   o.normal = normalize(mul(float4(v.normal, 0.0), unity WorldToObject).xyz);
    o.color = v.color;
    return o;
}
fixed4 frag (v2f i) : SV_Target
{
    fixed4 col = fixed4(i.color, 1.0);
    float3 lDir = float3(-_LightDirection.x, -_LightDirection.y, -_LightDirection.z);
   float d = (dot(i.normal, lDir) + 1) / 2;
    d *= LightDirection.w;
   return col - ((float4(1, 1, 1, 1) - _ShadowColor) * d);
}
```

4.6 Mittarit

Malleihin ja ympäristöihin voidaan asettaa erilaisia mittareita ja näyttötauluja, jotka pystyvät näyttämään oikeaa dataa jostakin kohteesta. Data tulee sovellukseen Valmetin palvelimilta OPCUA-protokollaa käyttäen. Kerätty data on peräisin erilaisista laitteista ja vastaa todellisia sen hetkisiä mittausarvoja kyseisessä laitteessa. Testaamiseen on käytetty myös kannettavalla tietokoneella ajettua testipalvelinta, jolla datan arvoja voidaan muuttaa manuaalisesti. Mittarit lisätään VR-näkymässä. Käyttäjän tulee tietää datavirrassa käytetty tagi saadakseen halutun arvon mittariin. Tagit näkyvät listattuna sovelluksen sisällä ja listasta voi myös hakea tageja, koska niitä on satoja. Mittarityyppi voidaan valita sopivaksi näytettävän datan perusteella. Tyyppejä ovat muun muassa viisarimittari ja veden korkeuden näyttävä säiliö. Tyypin valinta vaikuttaa vain mittarin graafiseen elementtiin, ja vaikka jotkut grafiikoista voivatkin näyttää mittausarvon, näytetään se silti aina myös tekstimuodossa selkeyden säilyttämiseksi.



KUVA 11. Erilaisia mittarigrafiikoita

4.7 Hälytykset

Mittarit voivat myös antaa hälytyksiä. Hälytys laukaistaan datavirran kautta tulevan tiedon perusteella, esimerkiksi kun jokin asetettu raja-arvo ylittyy tai alittuu. Silloin käyttäjän näkökenttään ilmaantuu oranssi nuoli, joka osoittaa hälytyksen saaneen mittarin suuntaan. Tarkastamisen jälkeen käyttäjä voi kuitata hälytyksen. Työkaluvyön painikkeesta *Alarms* pääsee tarkastelemaan kuittaamattomia hälytyksiä ja siirtymään hälytyksen antaneen mittarin luokse.

4.8 Siirtymäpisteet

Koska ympäristöt ja mallit olivat valtavan suuria ja monimutkaisia, sovellukseen tarvittiin jokin helppo tapa siirtyä tiettyyn, esiasetettuun pisteeseen. Käyttäjä voi lisätä siirtymäpisteen valitsemalla alavalikon painikkeista *Teleports* ja sitten aukeavasta ikkunasta *New Location Here*. Aukeavaan kenttään kirjoitetaan nimi siirtymäpisteelle myöhempää tunnistamista varten ja painetaan *Save* jos halutaan tallentaa, tai *Cancel* jos halutaan peruuttaa luominen. Jos käyttäjä tallensi pisteen, se ilmestyy siihen, missä hän sillä hetkellä on osoittaen suuntaan, johon hän katsoo. Pisteet tallennetaan malliin liittyvään JSON-tiedostoon (Liite 2).

Pisteisiin voi siirtyä saman valikon kautta. Listassa näkyy kaikki lisätyt pisteet ja niiden nimet. Käyttäjä osoittaa listasta pistettä, johon hän haluaa siirtyä ja painaa sitten ohjaimen liipaisinta. Käyttäjä siirtyy sinne, mihin piste on asetettu ja hänet käännetään katsomaan siirtymäpisteeseen tallennettuun suuntaan. Jos muokkaustoiminto on aktivoitu, käyttäjä voi uudelleennimetä tai poistaa siirtymäpisteitä.

Siirtymäpisteiden lisäämistä ja käyttämistä ei voi tehdä HoloLensillä, koska käyttäjä voi liikkua ainoastaan fyysisesti oikeassa maailmassa.

4.9 Vaara-alueet

Sovelluksessa voidaan merkitä vaara-alueita. Alueet näytetään ilmassa leijuvalla, mustakeltaraidallisella nauhalla. Vaara-alueet ovat vain visuaalisia, eikä niillä ole mitään varsinaista toiminnallisuutta. Ne kuitenkin auttavat havainnollistamaan vaarallisia alueita koneiden lähellä tai ympäristössä samalla tavalla kuin todellisuudessa. Asettaakseen uuden vaara-alueen, käyttäjä painaa alhaalta painikkeista *Danger Zones*. Tällöin aukeaa valikko, jossa on vaihtoehdot *New*, *Edit* ja *Delete*. Valitsemalla New käyttäjä aloittaa uuden vaara-alueen merkitsemisen. Vaara-alueet tallentuvat malliin liittyvään tiedostoon (Liite 2).

Merkitessä vaara-aluetta, käyttäjä merkkaa pisteitä ympäristössä ja niiden välille piirretään aluetta merkkaava nauha Unityn line renderer -komponenttia käyttämällä. VR-tilassa merkitsemisen voi tehdä kahdella tavalla. Käyttäjä voi osoittaa ohjaimella sinne, minne haluaa pisteen asettuvan. Tällöin käyttäjän täytyy osoittaa maahan, että ohjaimesta lähtevä säteensuuntaus osuu johonkin pintaan. Kosketuslevyä ylös tai alas painamalla käyttäjä voi kuitenkin nostaa tai laskea pistettä. Ohjaimen varressa olevaa painiketta painamalla voidaan vaihtaa toiseen tilaan, jossa pisteet asettuvat ohjaimen kohdalle. Molemmissa tiloissa käyttäjä asettaa pisteen paikalleen liipaisimella, jonka jälkeen hän voi asettaa seuraavan pisteen tai lopettaa merkitsemisen, jos vähintään kaksi pistettä on asetettu. Jos käyttäjä valitsee *Edit* hän voi siirtää nauhojen pisteitä. Ne ilmestyvät näkyviin vihreinä palloina. Niihin voi tarttua osoittamalla niitä ja painamalla liipaisinta ja laskemalla ne paikalleen painamalla liipaisinta uudelleen. *Delete*-tilassa samat pallot näkyvät, mutta nyt niiden valitseminen poistaa koko nauhan. Vaara-alueita ei pysty lisäämään, muokkaamaan tai poistamaan HoloLensillä rajoitettujen kontrollien vuoksi, mutta ne on mahdollista nähdä.



KUVA 12. Varoitusnauhalla merkitty vaara-alue

4.10 Infopaneelit

Mallissa voidaan näyttää infopaneeleita. Nämä paneelit näytetään sovelluksen ulkopuolella luodun datan perusteella. Ne voivat sisältää tekstimuotoista tietoa esimerkiksi laitteen yksittäisistä osista, kuten tietystä venttiilistä tai kytkimestä. Ympäristössä paneelit näkyvät kauempaa vihreinä kuutioina. Kun käyttäjä lähestyy niitä, ne aukeavat tauluiksi, jossa itse tiedot lukevat. Suljettuna ollessa tiedot eivät näy, ettei ympäristössä olisi liikaa visuaalisia ärsykkeitä. Tämä myös parantaa suorituskykyä.

4.11 Videot

Ympäristöön on mahdollista lisätä sekä tavallisia että 360 asteen videoita. Ennen kuin videoita voidaan lisätä sovelluksen sisältä, ne täytyy lisätä asetustiedostoon (Liite 3) joko

käsin tai asetustoiminnallisuuden avulla. Asetustiedostossa on nimi ja linkki videolle, jotka näkyvät sovelluksen sisällä videota lisättäessä.

Videoikkuna aukeaa alhaalta työkaluvyön *Videos*-painikkeesta. Videoiden lisäämiseen käytetään samanlaista valikkoa kuin siirtymäpisteiden lisäämiseen. Kun käyttäjä valitsee *New Video Here* aukeaa valikko, jossa käyttäjä valitsee pudotusvalikosta, mikä video lisätään tähän kohtaan. Valikon alapuolella näkyy myös videon URL valitsemisen jälkeen. Käyttäjä voi painaa *Done* lisätäkseen videon kohtaan, jossa hän sillä hetkellä seisoo tai *Cancel* peruuttaakseen toiminnon. Videoita voi myös uudelleennimetä ja poistaa tässä ikkunassa siirtymäpisteiden tapaan.

Lisätyt videot näkyvät ympäristössä ilmassa leijuvina virheinä palloina, joidenka sisällä on play-symbolia kuvastava kolmio. Kun käyttäjä aktivoi tavallisen videon, hänen eteensä ilmestyy videoikkuna. Videoikkunan alla on aikajana sekä tavalliset videokontrollit. Jos kyseessä on 360 asteen video, hänen ympärilleen ilmestyy muun ympäristön peittävä pallo, jonka sisäpintaan video piirretään. Pallo liikkuu käyttäjän mukana, joten hän on aina videon keskellä, mutta kääntyminen on tietenkin mahdollista. Pallon sisällä on myös samanlaiset videokontrollit kuin 2D-videoiden alla. Videoiden toistamiseen käytetään Unitylle suunniteltua AVPro nimistä pakettia sen moninaisten toimintojen ja tuettujen formaattien vuoksi.



KUVA 13. Video-objekti ympäristössä

4.12 Usean käyttäjän tila

Sovellusta on myös mahdollista käyttää kollaboratiivisessa tilassa, jossa useampi käyttäjä voi tarkastella mallia samanaikaisesti. Tässä tilassa käyttäjät pystyvät näkemään toisensa, mutta mitään varsinaista käyttäjien välistä interaktiivisuutta ei ole. Sovelluksen sisällä on mahdollista käyttää puhetta kommunikointiin, jos käytössä on mikrofoni. Käyttäjän puheääni lähetetään automaattisesti muille käyttäjille, kun sen voimakkuus ylittää tietyn rajan. Puheyhteyden toteutuksessa on käytetty Dissonance Voice Chat -nimistä Unity-pakettia. Käyttäjät voivat myös osoittaa asioita ohjaimesta lähtevällä lasersäteellä aktivoituaan sen grip-painikkeesta.

Sovelluksessa on myös mahdollista räätälöidä muille näkyvää hahmoaan. Muut käyttäjät näkyvät päästä ja torsosta koostuvina hieman robottimaisina hahmoina. Käyttäjä voi valita näytettävän nimen, hahmon värin sekä hahmon rinnassa näkyvän kuvan, johon on tarkoitus asettaa esimerkiksi yrityksen logo. Nämä kuvat pitää erikseen lisätä *LocalState*-kansion alle *logos*-kansioon. Jotta nämä räätälöinnit näkyisivät muille käyttäjille, täytyy nekin lähettää verkon yli. Nimen ja värin lähettäminen onnistuu helposti Unityn omilla toiminnallisuuksilla, mutta koska Unityn UNET-rajapinta ei tue tiedostojen lähettämistä suoraan, täytyi siihen tehdä oma toteutus.



KUVA 14. Hahmon muokkausnäkymä. Käyttäjä näkee vasemmalla hahmonsa, koska ei muuten voi sitä nähdä.

Kun käyttäjä liittyy palvelimelle tai vaihtaa kuvaa räätälöintivalikosta, asiakasohjelma lähettää viestin palvelimelle. Viestissä on asiakasohjelman uniikki, UNET-rajapinnan määrittämä tunnistenumero sekä kuvan tiedostokoko tavuina. Palvelin kirjaa tiedot muistiin ja alkaa odottamaan tiedonsiirron alkamista. Kuva lähetetään palvelimelle 1024 tavun viesteinä sekvensoitua luotettavaa kanavaa pitkin, joka pitää huolen siitä, että viestit saapuvat oikeassa järjestyksessä perille ja hukkuneet paketit lähetetään uudelleen. Tiedostoa ei voida lähettää kerralla, koska viestien maksimipituutta rajoittaa TCP:n maksimi segmenttikoko, joka on tyypillisesti 1460 tavua (Donato 2018). Kun kaikki tavut on vastaanotettu, palvelin lähettää tiedot eteenpäin kaikille yhdistäneille asiakassovelluksille paitsi tiedoston alkuperäiselle lähettäjälle. Näin myös muut palvelimelle yhdistäneet käyttäjät näkevät uuden kuvan. Palvelin myös pitää tallessa jokaiselle kuuluvan kuvan, jotta se voidaan lähettää myös niille, jotka yhdistävät palvelimelle alkuperäisen tiedostonsiirron jälkeen.

Palvelin on erillinen ohjelma, joka kootaan, kun Unityssa alustaksi valitaan x86. Se siis kootaan täysin samasta projektista. Palvelimen ollessa käynnissä ruudulla näkyy ainoastaan tiedot liittyneistä pelaajista. Palvelinohjelman kautta ei siis voi osallistua sessioon asiakasohjelman tapaan tai edes toimia katsojana.

4.13 Mallin koon hienosäätö

Jos mallista halutaan paremmin todellisuutta vastaavan kokoinen, sen kokoa pystyy hienosäätämään sovelluksen sisällä. Säätötoiminto avautuu työkaluvyön *Measure*-painikkeesta. Toiminnossa käyttäjä mittaa jonkin mallin osan asettamalla mittanauhan samaan tapaan kuin varoitusnauhoja asetetaan. Käyttäjä näkee mittauksen tuloksen ja voi sen jälkeen ilmoittaa odotetun tuloksen, eli siis todellisen mitan. Eli toisin sanoen, jos käyttäjä mittasi vaikkapa koneen sivun pituuden, hän laittaa odotettuun tulokseen koneen sivun todellisen maailman pituuden. Tämän jälkeen sovellus laskee mallin skaalan Unity-skenessä uusiksi niin, että mittaus vastaisi odotettua tulosta. Tämä muutos tallennetaan mallin asetustiedostoon (Liite 1), joten mittausta ei tarvitse tehdä uudelleen, kun malli taas ladataan.

4.14 Asetusten konfigurointi

Koska sovelluksessa on paljon asetuksia ja ulkoisia tiedostoja tehtiin niiden hallitsemista varten toiminnallisuus. Alun perin koko sovelluksen toimintaa hallitsevaa asetustiedostoa (Liite 3) ja malleja hallittiin käsin. Myöhemmin asetustiedoston muokkaamiseen käytettiin myös erillistä sovellusta, mutta uusimmassa versiossa toiminnallisuus lisättiin osaksi samaa sovellusta. Asetuksiin pääsee käynnistyksen yhteydessä klikkaamalla aloitusvalikossa (Kuva 9) *Settings*-painiketta. Sovelluksen aloitus- sekä asetustoiminnallisuus on toteutettu UWP-rajapintoja ja tekniikoita käyttäen. Se näyttää samalta kuin muutkin Windows 10 -sovellukset, vaikkakin kaikkia Microsoftin suunnitteluohjeita ei ole noudatettu ajan ja vaivan säästämiseksi.

Koska asetuksia ja muita toimintoja on monta, on ne jaoteltu eri kategorioihin. Asetusten ensimmäisessä ruudussa käyttäjälle näytetään eri kategoriat ja toiminnot. Eri asetukset on jaettu osioihin niiden runsaan määrän vuoksi. Verkkoyhteysasetuksista löytyy palvelinten

osoitteet sekä etätukeen ja kollaboraatioon liittyvien yhteystietojen lisääminen, muokkaaminen ja poistaminen. Linkit osiosta löytyvät verkko- ja videolinkkien lisääminen, muokkaaminen ja poistaminen. Sovelluksen sisäiset asetukset kattavassa osiossa voidaan vaihtaa esimerkiksi muokkaustilan aktivointiin käytettyä PIN-koodia, aloituksessa ladattavaa mallia ja liikkumiseen liittyviä asetuksia. Lisäksi osiosta löytyy pelaajan nimen ja värin vaihtaminen, jonka voi myös tehdä sovelluksen sisällä, sekä uusien värivaihtoehtojen lisääminen listaan, joka näkyy avataria muokatessa.

Valr	netXR			_	×
<i>\</i>	P	Ð,			
			Model Configs		
			VAN_40/540 VAN_40/540 VAN_40/540		
			Name		
			VAH_407340		
			Visuals Visual Name		
			VAH_407340_1.0M		
			Scale Import Scale		
			0.8669233 1		
			High Detail Model Low Detail Model		
			VAH_407340_1.0M Select File VAH_407340_1.0M Select File		
			+ 🛍		
			Render Inside Warehouse Render In Water		
			Off On		
			Render Floor		
			Off		
			Floor Levels		
			0 m 10 m 20 m 30 m -10 m -20 m		
			上 向		

KUVA 15. Mallien hallinta asetuksissa

Lisäksi käyttäjä voi tätä kautta lisätä uusia malleja ja luoda niille asetustiedoston. Näin ne saadaan helposti paikalleen, eikä käyttäjän tarvitse etsiä ja avata sovelluksen kansiota. Asetuksista ja malleista voi lisäksi luoda kopion pakattuna tiedostona, jolloin ne voidaan helposti siirtää toiselle koneelle. Pakatun tiedoston voi myös purkaa suoraan oikeaan kansioon tätä kautta. Sovelluksen avulla on myös mahdollista tallentaa asetukset suoraan toiselle laitteelle tai ladata ne sieltä. Tähän toimintoon käytetään Windows Device Portal -rajapintaa. Kyseinen rajapinta mahdollistaa Windows 10:ä käyttävän tietokoneen tai laitteen konfiguroinnin ja hallinnan etänä verkkosivukäyttöliittymän kautta. Tätä varten Microsoft on myös tehnyt UWP-kirjaston, jolla voidaan helposti käyttää Device Portalin REST-rajapintoja. Toiminto on otettava erikseen käyttöön laitteen asetuksissa. On myös syytä ottaa huomioon, että väärin konfiguroituna siihen liittyy valtavia tietoturvariskejä.

5 POHDINTA

Nykyaikaista virtuaalitodellisuutta, ja vähemmän myös lisättyä todellisuutta, on tutkittu tieteellisesti ja käytetty peleissä ja sovelluksissa jo melko paljon. Silti niihin liittyy vielä paljon haasteita. Tekniset ratkaisut, toteutustavat ja käytettävyyden aspektit ovat vielä jokseenkin vakiintumattomia, joten teknologiaa hyödyntäessä kokematon henkilö voi helposti tehdä huonoja ja kestämättömiä ratkaisuja. Samoin myös kokenutkin kehittäjä voi ajautua huonoihin ratkaisuihin, koska teknologia muuttuu edelleen nopeasti.

Esimerkiksi vielä muutama vuosi sitten VR-laseilla kuvan laatu oli todella huono matalasta resoluutiosta johtuen. Niinpä käyttöliittymäelementeistä saatettiin tehdä valtavan kokoisia, että ne olisivat selkeitä. Nykyään parhailla laseilla resoluutio on yli tuplasti parempi. Tärkeää onkin ymmärtää, että vielä ei voida suunnitella sovellusta, joka on varmasti yhteensopiva muutaman vuoden päästä sen hetkisten laitteiden tai tekniikoiden kanssa.

Onneksi kuitenkin monet asiat ovat vakiintumassa sellaisiksi, että yllämainittujen kaltaisten ongelmien ratkaiseminen olisi helppoa, tai että niitä ei syntyisi ollenkaan. Unity ja Microsoft tarjoavat rajapintoja AR- ja VR-sovelluksissa käytettäväksi. Ja kun näitä käytetään, tarvitsee ainoastaan päivittää rajapinnan tarjoava sovelluskirjasto. Olettaen siis, ettei ulospäin näkyvä rajapinta itsessään muutu.

Kannattaako yritysten sitten ylipäätään satsata teknologiaan, joka on vielä lastenkengissä. Se riippuu paljon siitä, mitä lisäarvoa lisättyä todellisuutta, virtuaalitodellisuutta tai niiden yhdistelmää käyttävä sovelluksen nähdään tuovan. On myös syytä pohtia, kuinka laajaa kokonaisuutta ajatellaan ja millaisesta työmäärästä on kyse. Jos sovelluksen kehittäminen käyttökelpoiseksi kestää vuosia, se ei ehkä ole järkevä sijoitus vielä tällä hetkellä. Samoin, jos työmäärä on suuri, mutta saatava hyöty pieni. Esimerkiksi vaateketju voisi tehdä AR-sovelluksen, jolla asiakas voi sovittaa vaatteita, mutta se vaatisi myös mahdollisesti tuhansien vaatteiden digitalisoinnin.

Siksi onkin järkevää kehittää jotain, mitä voidaan viedä eteenpäin pienissä osissa. Yllä mainittua vaateketjun sovellustakin voisi kehittää näin. Digitalisoitaisiin muutamia vaatteita, ehkä ketjun myydyimpiä, jonka jälkeen tarkastellaan sovelluksen käyttäjämääriä ja päätetään jatkokehityksestä sen mukaan. Myös opinnäytetyön aiheena olevaa sovellusta on kehitetty pienissä osissa. On ideoitu ja toteutettu pieniä kokonaisuuksia tai yksittäisiä ominaisuuksia ja testattu niitä. Näin on saatu arvokasta, kehitystä eteenpäin vievää ymmärrystä siitä mikä toimii ja mikä ei. Aiempaa ratkaisua on voitu muuttaa tai siitä opittua on voitu hyödyntää uusissa ominaisuuksissa.

Ennen kaikkea, sovellus on ollut siitä asti käytettävä, kun tärkeimmät ominaisuudet on saatu implementoitua. Näin Valmet on voinut käyttää sitä sisäisesti, sekä esitellä ja markkinoida sitä asiakkailleen, vaikka kokonaisuus ei olekaan valmis. Tämä myös lisää takaisinpäin tulevan palautteen määrää, joka edesauttaa sovelluksen kehittämisessä paremmaksi. Kaiken tämän tutkimisen, oppimisen ja toteuttamisen tuloksena on sovellus, joka pitää sisällään laajan skaalan ominaisuuksia. Toteutuksessa ei ole ajateltu vain yhtä käyttäjää ja yhtä käyttötapausta, vaan on pyritty vastaamaan nykyisten käyttäjien toiveisiin sekä mahdollisiin tulevaisuudessa ilmaantuviin tarpeisiin.

Ratkaisuja ja toimintatapoja kannattaa siis pohtia ja suunnitella, mutta samalla lähestyä niitä ehkä tavallista sovelluskehitystä rennommalla otteella. Jos kaikki tässäkin olisi suunniteltu ensin monen sivun dokumentteihin, joita olisi sitten seurattu joustamattomasti, ei tuloksena olisi varmasti puoliakaan nykyisestä sovelluksesta. Ja kuten mainittiin, vakiintuneita toteutustapoja ei juurikaan ole, joten on vaikea tietää etukäteen, mikä toimii ja mikä ei. On myös suositeltavaa kokeilla useampaa lähestymistapaa tai ratkaisujen kokonaisuutta ennen kuin hylkää koko idean. Esimerkiksi aiheena olevan sovelluksen mallien optimoinnin ja renderöinnin lopullinen tuotos oli monen asian summa. Kokonaisuutta kehitettiin eteenpäin sitä mukaa, kun tekninen ymmärrys parani tai käyttöön saatiin uusia apuohjelmia.

Jos uusia teknologioita hyödynnetään sopivalla tavalla ja sovelluksen tai tuotteen kehittämisprosessia lähestytään oikein, voi tuloksena olla kova etulyöntiasema tulevaisuudessa. Tärkeää on löytää juuri ne asiat, jotka tuovat lisäarvoa nyt ja ehkä vielä enemmän lähitulevaisuudessa ja keskittyä niihin. Aaron, P. & Etienne & Guyman, W. & Zeller, M. 2017. Motion controllers. Microsoft Mixed Reality developer documentation. Luettu 20.3.2019. <u>https://docs.microsoft.com/en-us/windows/mixed-reality/enthusiast-guide/motion-cont-rollers</u>

Aaron, P & Etienne & Zeller, M & Wojciakowski, M. 2016. Inside-out tracking. Microsoft Mixed Reality developer documentation. Luettu 7.2.2019. <u>https://docs.microsoft.com/en-us/windows/mixed-reality/enthusiast-guide/tracking-system</u>

Angelini, C. 2016. Microsoft HoloLens: HPU Architecture, Detailed. Luettu 20.3.2019. https://www.tomshardware.com/news/microsoft-hololens-hpu-architecure-28nm,32586.html

Azuma, R. T. 1997. A Survey of Augmented Reality. Hughes Research Laboratories.

Buss, S. R. 2003. 3D Computer Graphics: A Mathematical Introduction with OpenGL. Cambridge University Press.

Carmack, J. 2013. Principles of Lighting and Rendering with John Carmack at QuakeCon 2013. Youtube.com. Ladattu 27.2.2019 https://www.youtube.com/watch?v=IyUgHPs86XM

Colaner, S. 2016. What's Inside Microsoft's HoloLens And How It Works. Luettu 20.3.2019.

https://www.tomshardware.com/news/microsoft-hololens-components-hpu-28nm,32546.html

Cuervo, E & Chintalapudi, K & Kotaru, M. 2018. Creating the Perfect Illusion : What will it take to CreateLife-Like Virtual Reality Headsets?

Cruz-Neira, C. & Sandin, D. & DeFanti, T. & Kenyon, R. & Hart, J. 1992. The CAVE: audio visual experience automatic virtual environment. Communications of the ACM, 35, 64-72.

Donato, R. 2018. MTU, Jumbo Frames and MSS Explained. Packetflow. Luettu 20.3.2019.

https://www.packetflow.co.uk/mtu-jumbo-mss/

Hempel, J. 2015. Project HoloLens. Our Exclusive Hands-On With Microsoft's Holographic Goggles. Luettu 20.3.2019. <u>https://www.wired.com/2015/01/microsoft-hands-on/</u>

Jukić, T. Draw calls in a nutshell. 2015. Medium.com. Luettu 27.2.2019 https://medium.com/@toncijukic/draw-calls-in-a-nutshell-597330a85381

Lanham, M. 2017. Augmented Reality Game Development. Packt Publishing.

Linowes, J. 2018. Unity Virtual Reality Projects - Second Edition. Packt Publishing.

Linowes, J. & Babilinski, K. 2017. Augmented Reality for Developers. Packt Publishing.

Milgram, P. & Kishino, F. 1994. A Taxonomy of Mixed Reality Visual Displays. IEICE Transactions on Information Systems, Vol E77-D, No.12.

Saarelma, Osmo. 2018. Matkapahoinvointi. Duodecim Terveyskirjasto. Luettu 27.2.2019. https://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=dlk00301

Schnabel, M. & Wang. X & Seichter, H. & Kvan, T. 2007. From Virtuality to Reality and Back. INTERNATIONAL ASSOCIATION OF SOCIETIES OF DESIGN RE-SEARCH 2007 (IASDR07).

Sutcliffe, A. & Deol Kaur, K. 2000. Evaluating the usability of virtual reality user interfaces. Behaviour & Information Technology - Behaviour & IT.

Sega Hikaru Hardware. System 16. Luettu 27.2.2019 http://www.system16.com/hardware.php?id=724

Thompson, S. VR Lens Basics. Present And Future. Tom's Hardware. 2018. Luettu 20.3.2019. https://www.tomshardware.com/news/virtual-reality-lens-basics-vr,36182.html

Vivo, P. G. & Lowe, J. Getting started. The Book of Shaders. 2015. Luettu 20.3.2019.

https://thebookofshaders.com/01/

Vroegop, D. Microsoft HoloLens Developer's Guide. 2017. Packt Publishing.

Windows Mixed Reality PC hardware guidelines. Microsoft Windows support. Luettu 13.2.2019.

https://support.microsoft.com/en-us/help/4039260/windows-10-mixed-reality-pc-hard-ware-guidelines

Zaun, B. 2003. Calibration of Virtual Cameras for Augmented Reality. Technische Universität München Fakultät für Informatik.

LIITTEET

Liite 1. Mallin konfiguraatiotiedoston rakenne

Liite 2. Tiedoston rakenne, jossa on käyttäjän tekemät lisäykset

```
{
    "points": [
       {
         "tag": "TEST-AM",
"description": "Belt Tension",
          "xyz": {
    "x": -2.43889785,
    "y": 2.250462,
    "z": 2.76565623
          },
"orientation": 0.0,
"object": "measurement"
      }
   ],
"teleports": [
       {
          "name": "Teleport 1",
          "position": {
             "x": -2.34121510,
"y": 1.22145154,
"z": 9.62312156
          },
"rotation": 78.48347512
      }
    ],
     "videos": [
       {
          "name": "Test Video 1",
"url": "https://drive.google.com/uc?export=download&id=xxxxxxxx",
          "position": {
             "x": 0.0,
            "y": 1.2,
"z": 0.0
          "aspectRatio": 0.0
       },
       {
          "name": "Test Video 2",
"url": "https://drive.google.com/uc?export=download&id=xxxxxxxx",
           "position": {
             "x": 2.95756245,
             "y": 1.374165,
"z": -0.382958382
         },
"videoType": 1,
"aspectRatio": 1.777778
       }
   ],
"warningTapes": [
       {
          "positions": [
            },
             {
               "x": -7.430767059326172,
"y": 1.0000005960464478,
"z": -2.7894201278686525
             },
             {
                 "x": -9.880241394042969,
"y": 1.0000007152557374,
"z": -2.7830214500427248
             },
             {
                 "x": -10.12930679321289,
"y": 1.0000009536743165,
"z": 0.5117200016975403
            }
         ]
      }
],
}
```

```
"name": "admin",
       "password": "password"
   },
    "wifiNetworks": {
      "internet": {
"SSID": "",
"password": ""
      },
"factory": {
    "SSID": ""
    "
    "
    "

         "password": ""
       },
        'valmet": {
"SSID": "",
"password": ""
      }
   },
"OPCUAserverAddress": "192.168.1.10:26544",
"weblinkBaseAddress": "192.168.1.10:9090",
" .thinke". [
      {
         "url": "http://url",
"description": "Interlockings",
"wifi": "Factory"
      },
      {
         "url": "http://url",
          "description": "Function description",
          "wifi":<sup>'</sup>""
      },
      {
         "url": "https://url",
"description": "MaintALMA",
"wifi": ""
      },
      {
         "url": "https://url",
"description": "eServices",
"wifi": "Internet"
      }
   ],
    "activeModel": -1,
    "windowExpandDistance": 5.0,
    "keypadSequence": [
      2,
      4,
      6,
      8
   ],
    "maxPointingDistance": 10.0,
   "movementRotationDegrees": 45.0,
"movementStrafeDistance": 1.0,
    "videoLinks": [
     {
    "name": "Test",
    "url": "http://url",
    "videoType": "SphereMono",
    "aspectRatio": 0
}
   ],
    "remoteSupportInfo": {
    "pointrItems": [
         {
            "name": "",
             "phoneNumber": ""
         }
      ],
       "webRtcItems": [
         {
            "name": "",
"appName": ""
         }
      ],
       "collabItems": [
         {
            "name": "localhost",
            "serverIp": "127.0.0.1",
"serverPort": 7790
         }
```

Liite 3. Sovelluksen asetustiedosto (2)

}