LAMK Lahden ammattikorkeakoulu
Lahti University of Applied Sciences

Microsoft Hololens -applikaation toteuttaminen

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Mediatekniikka
Tekninen visualisointi
Opinnäytetyö
Kevät 2017
Joel Pesu

Lahden ammattikorkeakoulu
Mediatekniikka

PESU, JOEL:

Microsoft Hololens -applikaation toteuttaminen

Mediatekniikan opinnäytetyö, 44 sivua

Kevät 2017

TIIVISTELMÄ

Opinnäytetyö tehtiin Valve Group Oy:lle. Opinnäytetyön tarkoituksena oli toteuttaa Valve Group Oy:n asiakkaalle Nokia Oyj:lle Microsoft Hololens -applikaatio Barcelona Mobile World Congress 2017 -messuille demotarkoitukseen.

Applikaatio toteutettiin käyttäen pelimoottori Unityä, 3D-mallinnusohjelma Blenderiä ja 3D-mallien maalausohjelmaa Substance Painteria.

Opinnäytetyön teoriaosuus keskittyy sovelluksessa käytetyn mixed realityn kehityshistoriaan. Esiteltyjä tekniikoita ovat virtual reality eli VR, augmented reality eli AR ja mixed reality eli MR.

Käytännön osuudessa kerrotaan applikaation kehitysvaiheista ja kuvataan kehitystyössä käytettyjä tekniikoita sekä kerrotaan lopputuotteeseen hyväksytyistä toimintatavoista.

Opinnäytetyön tuloksena applikaatio esiteltiin messuilla tilaajan suljetulla osastolla valikoidulle yleisölle.

Asiasanat: Hololens, lisätty todellisuus, virtuaalitodellisuus, mixed reality

Lahti University of Applied Sciences
Degree Programme in Media Technology

PESU, JOEL: Development of a Microsoft Hololens application

Bachelor's Thesis in Media Technology, 44 pages

Spring 2017

ABSTRACT

The objective of this study was to create a Microsoft Hololens application for Valve Group´s client Nokia Oyj to be used at the Mobile World Congress 2017 exhibition in Barcelona, Spain.

The application was developed by the Unity game engine, 3D software Blender and 3D painting application Substance Painter.

The theoretical part of the thesis concentrates on the development history of the Mixed Reality technology. The technologies introduced are virtual reality, abbreviated as VR, augmented reality, abbreviated as AR, and mixed reality, abbreviated as MR.

In the case study, the development stages, as well as the used technologies of the application are described. Procedures approved for the final product are portrayed.

As a result of the thesis the application for Hololens was introduced at the exhibition for selected visitors at the private stand of the client.

Key words: Hololens, augmented reality, virtual reality, mixed reality

SISÄLLYS

1 JOHDANTO

Tutkimuksen kohteena on Microsoftin kehittämät Mixed realityn mahdollistavat Hololens-älylasit, jotka ovat toistaiseksi ainoat markkinoilla olevat vastaavaa teknologiaa käyttävät lasit. Tutkimuksella pyritään selvittämään, miten Hololens-älylasit soveltuvat laajan yleisön käytettäviksi ja miten niiden applikaatioista tehdään käyttäjäystävällisiä. Tutkimuksen näkökulmana on kehittäjän näkökulma, ja tutkimus perustuu kuvattuun caseen.

Tutkimuksen alussa kerrotaan lyhyesti lisätyn- ja virtuaalisen todellisuuden historiasta ja siitä, kuinka Mixed realityyn ja Hololensiin on päädytty. Lisäksi tutkimuksessa kerrotaan Hololensin teknisestä puolesta, kuten laitteistosta ja sen rajapinnoista. Rajapinnoista ja ohjelmoinnista Hololensille kerrotaan enimmäkseen pelimoottori Unityn näkökulmasta.

Case-osuudessa kerrotaan Hololens-sovelluksen kehittämisestä suomalaisen tietotekniikkayrityksen näyttelydemoa varten Barcelonan Mobile World Congress 2017 –messuille.

2 MIXED REALITYN SYNTY

Mixed reality eli MR on helpointa selittää kahden sen edeltäjän, augmented realityn eli AR:n ja virtual realityn eli VR:n kautta.

2.1 AR

AR, eli augmented reality, suomeksi lisätty todellisuus, on jatkuvasti kehittyvä informaatioteknologian ala. Sillä pyritään lisäämään oikeaan maailmaan tietokoneella generoitua dataa. Lisätty data voi olla esimerkiksi todellisuuteen lisättyä tekstitietoa tai jopa 3D-malleja. Vaikka lisättyä todellisuutta on työstetty jo vuosikymmeniä, sen kehitys on ollut viime vuosikymmentä lukuun ottamatta hidasta. Tämä on johtunut ennen kaikkea laitteistojen puuttellisista tehoista AR:ä varten.

Lisättyä todellisuutta on kolmea erilaista: video based eli videopohjainen, video see-through ja optical see-through. Videopohjaisessa AR:ssä lisätään videoon, joko nauhoitettuun tai suoratoistoon, dataa, minkä jälkeen videota katsotaan monitorilta. Tämä lähestymistapa on yleinen televisiossa, etenkin urheilulähetyksissä. Esimerkiksi urheilulähetyksissä uinnista näytetään usein uimarin tullessa maaliin hänen sijoituksensa numerona uimarin radan päällä kuten kuvassa 1. Samoin keihäänheitossa saatetaan näyttää heittojen pituus lisättynä kentälle.

KUVA 1. Televisiolähetykseen lisättyä dataa (Rieder 2011)

Video see-through eroaa videopohjaisesta siten, että tässä menetelmässä kuvattu video ja siihen lisätty data katsotaan monitorin sijaan katseen eteen asetetusta lähteestä, kuten jonkinlaisesta head mounted displaystä eli HMD:stä. Tässä metodissa heikkoutena on riippuvuus kameran tai kameroiden tarkkuudesta. Huono resoluutio kuvassa saa kaiken näyttämään huonolta. Mobiililaitteiden, kuten kännyköiden ja tablettien, lisätyn todellisuuden sovellukset kuuluvat useimmiten video see-through -kategoriaan.

Optical see-throughissa silmien eteen asetetaan läpinäkyvä visiiri, jonka kautta käyttäjän näkökenttään heijastetaan lisätty data. Edellä mainittuun video see-through -tekniikkaan verrattuna, tässä teknologiassa ei tarvitse nauhoittaa oikeaa maailmaa. (Hughes & Wright 2010, 59 – 61.) Näin voidaan säästää laskutehoa videon prosessoinnista eikä kameroiden tarkkuuteen tarvitse tukeautua. Visiiriin projisoitu kuva on additiivista sen takana olevan valon kanssa. Tämän takia optical see-through -visiireihin ei tällä hetkellä voida toistaa tummia värejä.

2.2 VR

Virtuaalitodellisuudella pyritään stimuloimaan ihmisen kaikkia aisteja, kuten näköä, kuuloa ja tuntoaistia, ja täten siirtämään hänet täysin tietokoneella generoituun todellisuuteen. Teknisesti virtuaalitodellisuus on tietotekniikan laaja osa-alue. Koska kaikkia aisteja on vaikea stimuloida yhtäaikaisesti, ovat monet nykyaikaisista virtuaalitodellisuuden laitteistoista keskittyneet vain tiettyjen aistien ärsyttämiseen. Yleisimmin keskitytään näköön. Näköaistia varten on kehitetty muutamia erilaisia lähestymistapoja, kuten cave automatic virtual environment, eli CAVE, ja nykyaikaisemmat virtuaalitodellisuus HMD:t.

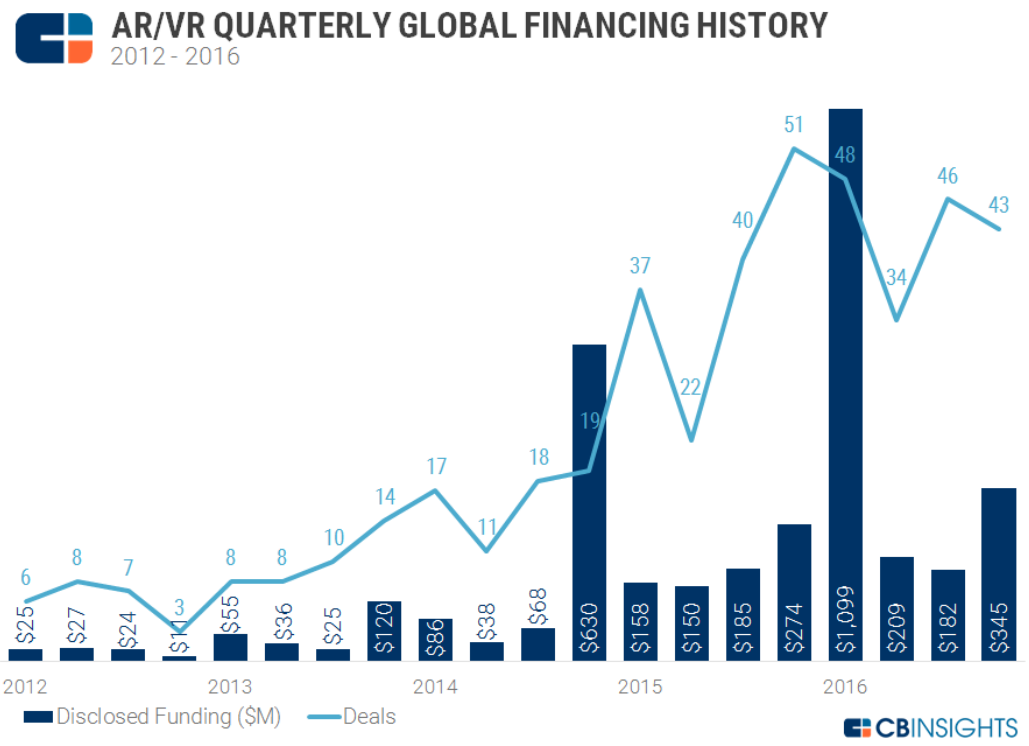
CAVE-systeemissä kuution muotoisen tilan jokaiselle tahkolle projisoidaan vuorotellen vasemman ja oikean silmän perspektiivi kyseisen tahkon virtuaaliympäristöä. Käyttäjä seisoo lasien kanssa keskellä tilaa kuten kuvassa 2. Lasit päästävät sykronoidusti läpi oikean silmän kuvan luoden syvyysvaikutelman. (Chrysanthou, Slater & Steed 2002, 7.) CAVE-

järjestelmät vaativat paljon tilaa, joten ne eivät sovellu kuluttajakäyttöön.

KUVA 2. CAVE-virtuaalitodellisuusjärjestelmä (CloudTweaks 2014)

Suosittu ja viime aikoina räjähdysmäisesti kasvanut tapa lähestyä näköaistin viemistä virtuaalitodellisuuteen ovat erilaiset HMD:t, joissa on sisäänrakennetut näytöt. HMD:ssä on usein omat näytöt molemmille silmille ja pään liikettä seurataan gyroskooppien ja erilaisten kameroiden sekä lasereiden avulla, jotta näkökulmaa virtuaalitodellisuudessa voidaan muuttaa pään liikkeen mukaisesti.

AR:n ja VR:n kehitys on ollut viime vuosina räjähdysmäisen nopeaa. Tammikuusta 2014 tammikuuhun 2016 investoinnit AR- ja VR-markkinoille kasvoivat yli 20-kertaisesti. (Digi-Capital 2016.) Markkinoiden kasvu on kuvattu kuviossa 1.



KUVIO 1. AR/VR rahoitus vuodesta 2012 vuoteen 2016 (CB Insights 2017)

2.3 Virtuaalitodellisuuden historia

VR:n historia yltää viime vuosisadan alkukymmenyksille. Ensimmäisiä VR-laitteistoja oli olemassa jo 1920-luvulla. Sellainen oli esimerkiksi Link-lentosimulaattori, joka on kuvassa 3. Se oli pienoismalli lentokoneesta, ja sitä käytettiin lentäjien koulutukseen etenkin toisessa maailmansodassa. Simulaattorin tarkoitus oli opettaa lentäjiä lukemaan ohjaamon instrumentteja, sillä ne olivat ainoat asiat, joita hytissä voitiin nähdä. (Roberson Museum and Science Center 2017.)

KUVA 3. Link C-3 -lentosimulaattori (Roberson Museum and Science Center 2017)

Muita VR:n kannalta merkittäviä laitteita olivat Morton Heiligin kuvassa 4 näkyvä Sensorama vuodelta 1957. Sensorama oli ensimmäisiä laitteita, joissa oli 3D-videota. Kuten useimmat näistä alkuajan VR-laitteista, Sensoramakaan ei tyytynyt pelkästään näön viemiseen virtuaalimaailmaan vaan kunnianhimoisesti halusi tuoda elämykseen myös hajuaistin sekä jotain myös tuntoaisteille kuten tärinän ja tuulen. (The father of virtual reality 2017.)

KUVA 4. Sensorama (Turi 2014)

Vuonna 1995 VR-markkinoille tuli uusi tuote, jonka oli tarkoitus soveltua mahdollisimman moneen kotitalouteen. Tuote oli Nintendon Virtual Boy. Sen punaisista LED-valoista koostuva 3D-kuva ei kuitenkaan vakuuttanut suurta yleisöä, ja sen 180 dollarin hinta oli vielä monille liian korkea. (Kohler 2010.) Laitetta markkinoitiin konsolipelaajille. Virtual Boyn pelit olivat kuitenkin varsin kehnoja ja pelivalikoimaa oli vähän. Verrattaessa Virtual Boyta muihin saman aikakauden pelinkonsoleihin, kuten Playstationiin tai Sega Satruniin, voidaan nopeasti todeta, että Virtual Boy oli teknisesti jäljessä kilpailijoistaan.

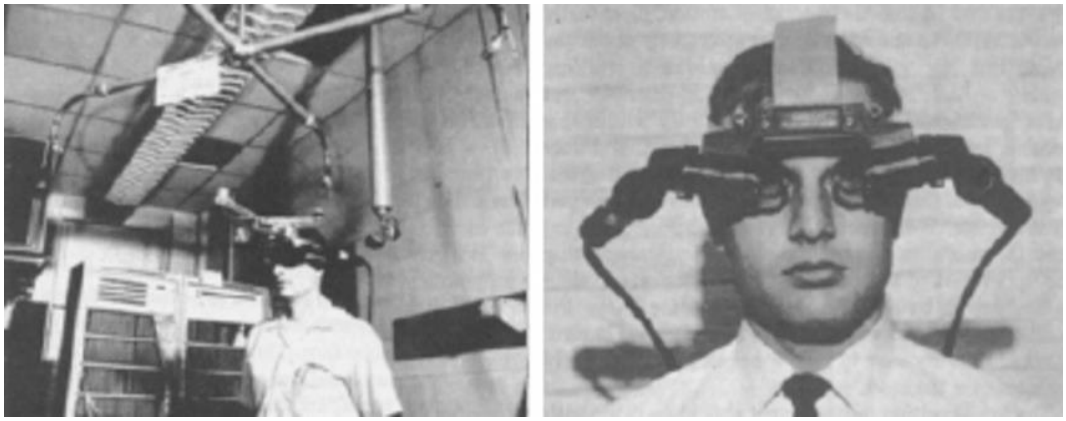
VR:n läpimurtona voidaan pitää Oculus VR:n virtuaalijärjestelmän Riftin keksimistä. Elokuussa 2012 kalifornialainen virtuaalitodellisuuden harrastaja Palmer Luckey päätti hankkia joukkorahoitusta toteuttaakseen ideansa VR-HMD:stä eli headmounted displaystä. 250 000:n dollarin raja meni rikki alle 24 tunnissa. Lopullinen kerätty rahamäärä oli kaksi miljoonaa Yhdysvaltain dollaria. Jo ensimmäistä kehittäjäversiota Oculus Riftistä pidettiin parempana kuin yhtäkään sen edeltäjistä. (Kumparak 2014.)

Kesäkuussa 2015 Oculus Riftin toista kehittäjäversiota oli myyty yli 100 000 kappaletta. Maaliskuussa 2014 Facebook osti Oculus VR:n 2,3 miljardilla dollarilla.

2.4 Lisätyn todellisuuden historia

AR on digitaalisen informaation lisäämistä käyttäjän todellisuuteen reaaliajassa. Ehkä helpoimmin ymmärettävä esimerkki AR:stä on lentäjän HUD eli heads-up display. (Kipper & Rampolla 2012, 101.) Etenkin hävittäjälentäjien visiiriin on usein projisoitu lentäjälle tärkeää informaatiota, kuten esimerkiksi asento horisonttiin nähden, suuntavektori tai taistelulentokoneissa aseiden status. HUD:n avulla lentäjä näkee paljon hyödyllistä informaatiota lennon aikana kääntämättä päätään tuulilasista muualle.

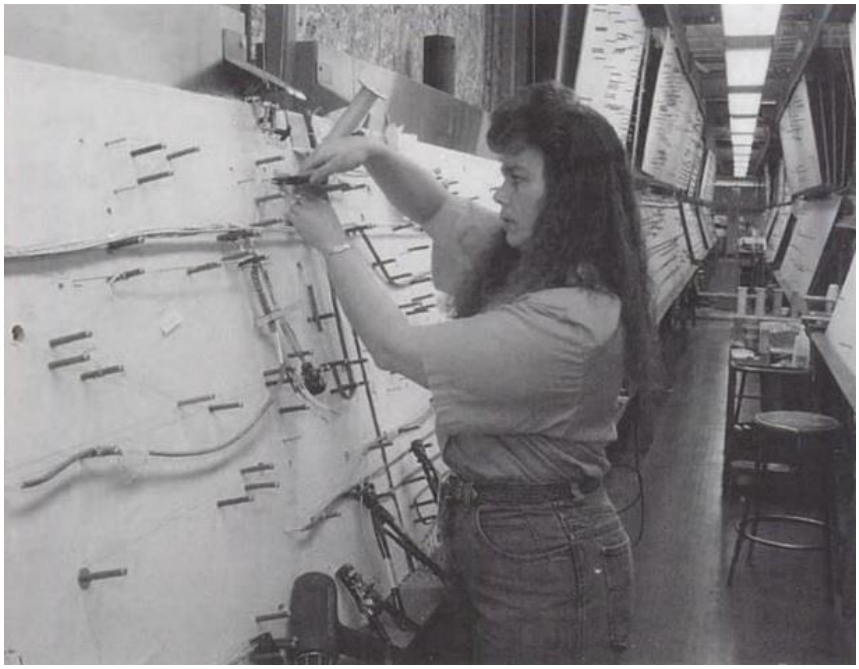
AR:n historian, ainakin käsitteenä, katsotaan alkaneen vuonna 1968. Sen isänä pidetään yhdysvaltalaista Ivan Sutherlandia, joka vuonna 1968 keksi Bob Sproullin kanssa ensimmäisen AR -head mounted displayn eli HMD:n. Laitteen nimi oli The Sword of Damocles. Laite piirsi käyttäjän näkökenttään rautalankamallin tyyppisiä 3D-malleja. Käyttäjä pystyi myös liikkumaan mallien ympärillä, mistä syystä kuvassa 5 esitettävää The Sword of Damoclesia pidetään kaikkien aikojen ensimmäisenä AR-näyttönä. (Steinicke 2016. 27.)



KUVA 5. Sword of Damocles (ResearchGate 2007)

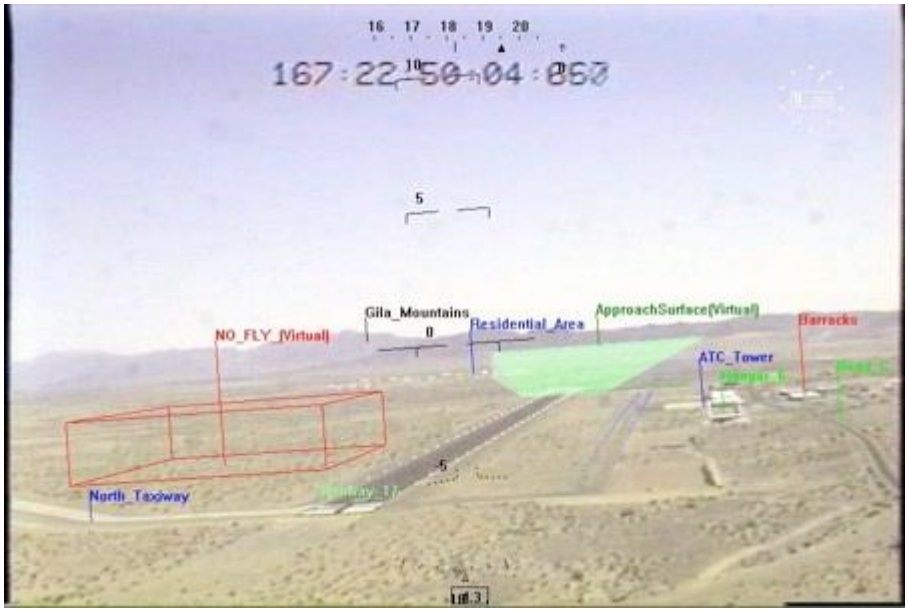
Muita AR:n historian merkittäviä hetkiä olivat muun muassa vuosi 1990, jolloin amerikkalainen lentokonevalmistaja Boeing pyysi Tom Caudellia ja David Mizelliä kehittämään tapoja hyödyntää See through VR:ää eli AR:ää kaupallisten lentokoneiden suunnittelussa ja tuotannossa.

Vielä tuolloin lentokoneiden sähköjohdot kasattiin kuvassa 6 esitetyn mukaisesti vasten tauluja, joihin oli piirretty johtojen kaaviot. Koska pohjapiirustuksia oli monia, piti näitä tauluja myös olla monia. Tom Caudell keksi tehdä HMD:n, jonka avulla tyhjän taulun päälle voitiin lisätä diagrammit. Täten tauluja ei enää tarvinnut räätälöidä monia erilaisia. Tom antoi menetelmälle nimen augmented reality. (Caudell & Barfield 2001, 447 - 449.)



KUVA 6. Lentokoneiden sähköjohdot kasattiin tauluja vasten (Caudell & Barfield 2001, 448)

Vuodesta 1998 Nasa käytti astronauttien takaisinpaluualuksessa X-38 LandForm-ohjelmistoa. LandForm lisäsi aluksen kameran kuvaan kiitoradan, lennonjohdon tornin ja hangaarien sijainnin kuvion 2 mukaisesti. Tästä oli hyvin paljon apua esimerkiksi silloin, kun aluksen kamera jäätyi eikä kuvaa jääpeitteen vuoksi voinut nähdä. (SweetHaven 2017.)



KUVIO 2. LandFormin näkymä (Landform 2017)

Vuonna 1999 SIGGRAPH-konferenssissä esiteltiin ensimmäinen versio ARToolKitistä, joka on edelleen paljon käytetty SDK (Software Development Kit). Sen avulla ohjelmoijat voivat kehittää AR-sovelluksia. ARToolKitissä on tuki Android- ja iOS-käyttöjärjestelmille ja Unity-pelimoottorille. 2001 ARToolKit julkaistiin avoimeksi lähdekoodiksi. Vuoteen 2016 mennessä ARToolKitiä ja sen variantteja on ladattu miltei miljoona kertaa. (ARToolKit 2015.)

2010-luvulla AR-teknologian kehitys on ollut räjähdysmäistä. Voidaan sanoa idean jokaisen käyttäjän henkilökohtaisista AR-laseista lähteneen Google X:ltä. Googlen kehittämien lasien nimi oli Google Glass, ja ne julkaistiin vuonna 2012. Vuonna 2013 laseja alettiin myydä nimellä Google Glass Explorer Edition, joka oli varhaisille adoptoijille ja kehittäjille tarkoitettu versio. (Wikipedia 2017.)

Google Glass esitteli yleisölle mahdollisuuden eräänlaisista silmälaseista, jotka lisäävät näkökenttään hyödyllistä tietoa ilman että käyttäjän tarvitsee esimerkiksi kaivaa puhelinta taskusta. Äänikomentojen avulla lasit pystyivät toimimaan henkilökohtaisena assistenttina, joka totteli käskyjä. Google Glassilla pystyi myös lukemaan sähköpostia ja navigoimaan. (Brownlee 2013.)

Vuoden 2015 tammikuussa Google ilmoitti lopettavansa Google Glass Explorer Editionin myynnin ja vetäytyvänsä kehittämään uutta parannettua versiota laseista (Google 2015).

2.5 Mixed reality

Mixed reality on virtuaali- ja lisätyn todellisuuden yhdistelmä. Virtuaalitodellisuudessa käyttäjä sijoitetaan täysin simuloituun ympäristöön. Tässä simuloidussa ympäristössä todellisen maailman fysiikan lait eivät välttämättä päde ja käsitys ajasta ei välttämättä vastaa todellisuutta.

Jo vuonna 1994 ATR Communication Systemsin tutkimuslaboratorion tutkimuksessa aiheesta mixed reality todettiin, että mixed realityn toteuttamiseksi pitää tietokoneen ymmärtää sekä virtuaalinen maailma että oikea maailma samanaikaisesti. Tutkimuksessa tätä kutsuttiin Extent of World Knowledgeksi eli EWK:ksi.

Virtuaalitodellisuutta simuloiva tietokone ymmärtää koko virtuaalisen tilan ja kaiken, mitä se sisältää, esimerkiksi sen, missä kukin virtuaalinen objekti on, mitä sen takana on ja missä käyttäjä on virtuaalisessa tilassa, mutta ei tiedä mitään todellisesta maailmasta.

Lisätyssä todellisuudessa lisätään oikeaan maailmaan jotain simuloitua ja tietokoneella on tieto, mihin virtuaalista dataa pitää lisätä. Esimerkiksi videokuvan perusteella tietokone tunnistaa tietyn kuvan tai objektin, esimerkiksi kuvan logosta tai pallon muodon, jonka päälle se on ohjelmoitu lisäämään tekstiä. Tietokone ei kuitenkaan tiedä, mikä tämä muoto on esimerkiksi mittasuhteiltaan tai materiaaliltaan ja kuinka kaukana se on käyttäjästä. Tietokone ei myöskään välttämättä tiedä, mitä objektin takana on. Mixed realityssä halutaan lisätä todelliseen maailmaan jotain synteettistä, joka voisi olla vuorovaikutuksessa todellisen maailman kanssa.

Esimerkiksi virtuaalinen pallo voi kieriä todellisen maailman pöydän päällä ja pudota sen kierittyä pöydän reunan yli. Maan vetovoima suhteessa

virtuaaliseen palloon voi toimia myös luonnon lakien vastaisesti ylöspäin niin, että se törmää todellisen maailman katon kanssa. Tähän mixed reality laitteistossa tarvitaan tietoisuus oikeasta ympäristöstä. Tätä varten EWK:n täytyy olla niin hyvä, että tiedetään tarkalleen, missä pöytä on ja mitkä ovat sen mittasuhteet. (Milgram, Takemura, Utsumi & Kishino 1994.)

3 HOLOLENS-TEKNOLOGIA

Microsoftin kehittämät, kuvassa 7 olevat, Mixed reality- ja Augmented reality -älylasit ovat nimeltään Hololens. Lyhyesti Hololens on päähän asetettava älylaite, jonka navigointi tapahtuu käsien liikkeillä ja puhekomennoilla.

Hololens lisää käyttäjän näkökenttään niin sanottuja hologrammeja, joita ne kuitenkaan eivät oikeasti ole. Oikeat hologrammit toteutetaan nauhoittamalla objekteista heijastuneita valonsäteitä, jotka rekonstruktoidaan uudestaan esimerkiksi laserilla. Hologrammeissa toteutuu parallaksi-ilmiö, jossa paikallaan oleva esine näyttää liikkuvan taustaa vasten etäisyyden mukaan eli mitä kauempana esine on, sen vähemmän se näyttää liikkuvan. (Wikipedia 2017.)

Hololensissä kuvan muodostavat valonsäteet taitetaan hiloilla eli lasilevyillä valonlähteestä käyttäjän silmiin (Guttag 2016). Hololensissä molempiin silmiin heijastetaan eri kuvat eri perspektiiveistä ja yhdessä ne luovat illuusion kolmiulotteisesta kuvasta hologrammin tapaan. Tutkimuksessa käytetään sanaa ”hologrammi” kuvastamaan näitä Hololensin tuottamia visuaaleja.

KUVA 7. Hololens-älylasit (Microsoft 2017a)

3.1 Hololensin tekniikka

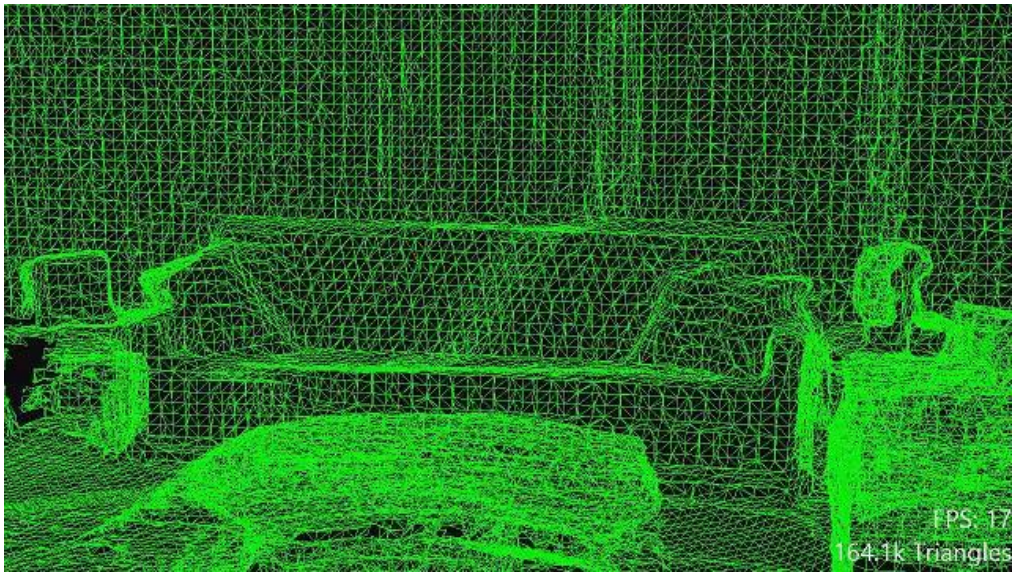
Toisin kuin VR-laseissa, joissa molempien silmien kuvat piirretään kahdelle näytölle, joita taas katsotaan erilaisten linssien takaa, Hololensissä linssit ovat läpinäkyvät. Kuva projisoidaan kahden 16:9 HD light enginen voimin. Talletustilaa Hololenssistä on 64 gigatavua ja keskusmuistia kaksi Gigatavua. CPU Hololensissä on yhden gigahertsin Intelin 32-bittisen arkkitehtuurin Atom-prosessori. Käyttöjärjestelmänä toimii Windows 10:n jatkannainen Windows Mixed Reality (Wikipedia 2017.) Hololensin paino on noin 579 grammaa. Hololensin hardwareen kuuluu myös niin sanottu IMU eli inertial measurement unit, joka sisältää muun muassa kiihtyvyysanturin, gyroskoopin ja magnetometrin.

3.2 Spatial mapping

Spatial mapping tarkoittaa käyttäjän ympäristön ymmärtämistä. CPU:n ja GPU:n lisäksi Microsoft kehitti Hololensille kolmannen prosessorin HPU:n, eli holographic processing unitin, ympäristön ja käyttäjän liikkeiden tunnistamiseksi (Holmdahl 2015).

HPU kerää tietoa Hololensin neljästä ympäristöä tulkitsevasta kamerasta ja yhdestä syvyyden tunnistavasta kamerasta, sekä ympäristön valaistusta lukevasta sensorista. (Microsoft 2017h.) Tällä tiedolla voidaan tehdä 3D-malli käyttäjän ympäristöstä, niin sanottu spatial mesh, jonka päälle hologrammit voidaan asettaa. (Fitzsimmons 2015). Kuviossa 3 oleva spatial mesh, jota toisinaan kutsutaan myös spatial mapiksi, koostuu kolmioista. Kolmioiden tiheys, eli spatial mapin tarkkuus, voidaan ohjelmallisesti asettaa. Tiheämpi ja tarkempi spatial map vaatii enemmän resursseja Hololensin GPU:lta.

Spatial mappia voidaan käyttää peittämään seinän takana olevia hologrammeja. Tätä kutsutaan okluusioksi. Lisäksi spatial mappia voidaan käyttää tekemään törmäystarkastuksia ja muita fysiikkalaskutoimituksia reaalimaailman kanssa.



KUVIO 3. Rautalankamalli Hololensin generoimasta spatial mapista (Microsoft 2016)

3.3 Spatial anchor

Spatial anchorit ovat nimensä mukaan spatiaalisia, eli avaruudellisia ankkureita. Näiden ankkurien tarkoitus on pitää hologrammit paikallaan reaalimaailmaan nähden. Ankkureilla merkitään reaalimaailman tärkeät pisteet. Ankkurin ympärille rakentuu koordinaatisto, johon voidaan liittää hologrammeja, joiden halutaan pysyvän mahdollisimman hyvin paikoillaan reaalimaailmaan nähden. Spatiaaliset ankkurit korjaavat jatkuvasti etäisyyksiä toisistaan, samalla kun Hololensin spatiaalinen ymmärrys paranee. Esimerkki: Jos Hololens luulee kahden lokaation olevan neljän metrin etäisyydellä toisistaan ja myöhemmin tajuaakin etäisyyden olevan 3,9 metriä, kyseisiin paikkoihin liitetyt hologrammit olisivat ilman ankkureita edelleen neljän metrin etäisyydellä toisistaan.

Ankkureiden lokaatiot voidaan tallentaa niin sanottuun spatial anchor storeen, josta ne voidaan Hololensin, tai sen applikaation, uudelleen käynnistettyä ladata uusiksi. Näin hologrammit voidaan pitää paikallaan jopa uudelleenkäynnistyksen jälkeen (Microsoft 2017i).

Ankkurit, spatial mesh sekä sensoridata voidaan jakaa myös muiden Hololensien kanssa. Näin eri Hololensien hologrammien paikat saadaan synkronoitua keskenään (Microsoft 2017b).

3.4 Gestures

Hololensissä on rajattu määrä tapoja, joilla siinä voidaan navigoida. Perusinteraktioihin kuuluu painallus, vapautus ja bloom. Painalluksella ja vapautuksella tarkoitetaan normaalia pressiä ja releasea. Painallus ja vapautus toimii joko napsauttamalla etusormea ja peukaloa Hololensin edessä, tai käyttämällä sen mukana toimitettavaa Clickeriä eli interaktiovälinettä. Clickerissä on vain yksi nappi, joka korvaa sormilla napsauttamisen. Bloom sen sijaan on kädenliike, jolla päästään Hololensin starttivalikkoon. Lisäksi Hololens ymmärtää puhetta, joten sitäkin voidaan tarvittaessa käyttää.

Vaikka interaktiotapoja on vähän, Hololens mahdollistaa kuitenkin muun muassa käsien seuraamisen, jos kädet ovat näkyvillä Hololensin kameroille. Näin voidaan ohjelmoimalla toteuttaa applikaatioihin monia erilaisia tapoja tehdä interaktioita hologrammien kanssa.

Yleisimpiä ovat seuraavat:

Hold: pidetään sormia tai Clickeriä pohjaan painettuna hold-kynnyksen yli. Hold-kynnys tarkoittaa aikaa, jonka verran painalluksen tulee kestää, jotta Hold-tapahtuma laukeaa.

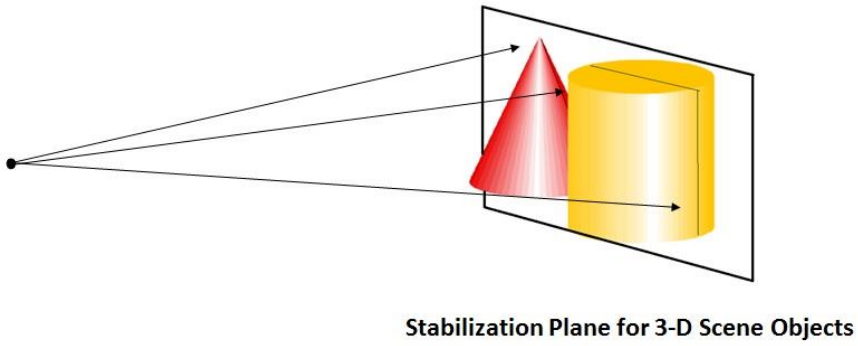
Manipulation: painallus, jota seuraa kädenliike koko 3D-tilassa

Navigation: painallus, jota seuraa kädenliike tietyllä alueella

Näitä toimenpiteitä käytetään yleensä yhdessä niin sanotun Gazen eli katseen kanssa. Gazella tarkoitetaan Hololensin katsetta ja siitä saatavaa dataa, kuten keskellä näkökenttää esiintyvää hologrammia (Microsoft 2017c.)

3.5 Stabilization plane

Stabilization plane on kameraa, eli HMD:tä kohti osoittava taso, jota Hololens parhaansa mukaan yrittää pitää stabiilina. Sen toimintaperiaate on kuvatu kuviossa 4. Hyvä stabilointi estää hologrammien värierottelua ja hologrammien liikkumista kameran mukana nopeissa liikkeissä. Tason toiminta on yksinkertainen: taso asetetaan tiettyyn pisteeseen Hololensin ymmärtämässä koordinaatistossa ja sen pintanormaali osoittamaan kohti kameraa. Kaikki tason kanssa leikkaavat hologrammit vastaanottavat maksimaalisen määrän laitteiston stabilisointia (Microsoft 2017d.)



KUVIO 4. Stabilisaatiotason toiminta (Microsoft 2017d)

Tason paikkaa voidaan vaihtaa ohjelman ajoaikana. Tasoa voidaan esimerkiksi liikuttaa kameran mukana tietyllä etäisyydellä, ja kameran katseen osuessa hologrammiin taso voidaan kiinnittää siihen ja pois katsoessa taas liittää kameraan.

3.6 Device portal ja REST-api

Hololensin tarkempaa konfigurointia ja tarkkailua varten on Microsoft kehittänyt Hololensin REST -APIa hyödyntäen Device portalin. Device portal on selaimella käytettävä hallintapaneeli, johon pääsee käsiksi, kun aktivoidaan kehittäjämoodi sekä Device portal Hololensin asetuksista. Itse portaalin osoite on joko Hololensin ip-osoite WLAN:n kautta tai http://127.0.0.1:10080 USB-kaapelin kautta. Device portalista voi nähdä hyödyllistä dataa Hololensistä, kuten Spatial meshin, CPU:n, GPU:n, tai muistin käytön ja käynnissä olevat ohjelmat. Device portalin kautta on

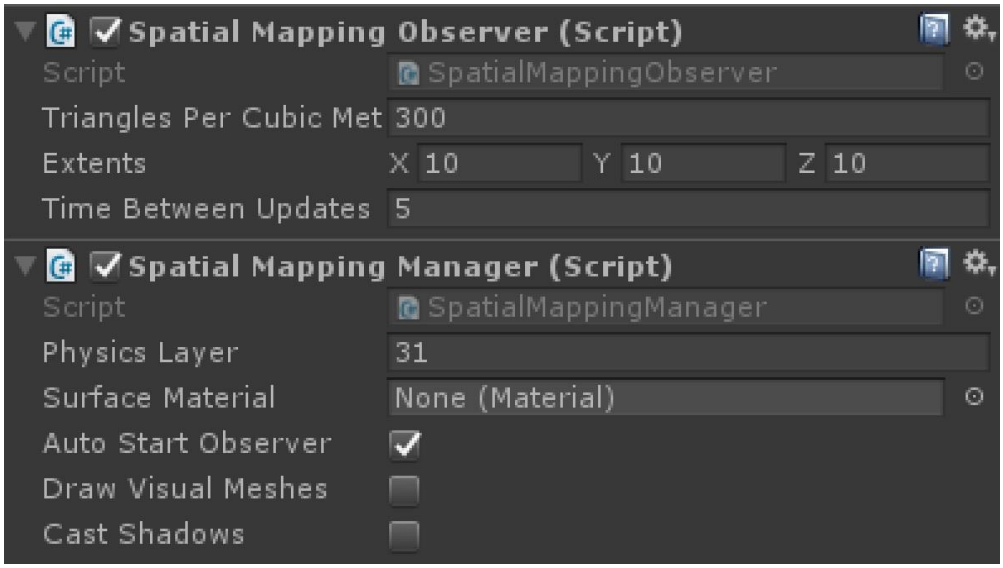
myös mahdollista asentaa applikaatioita Hololensille, kuten Windows storen ulkopuolella jaetut applikaatiopaketit, joita ei ole sertifioitu. Sertifioimattomien applikaatioiden asentaminen on yleistä applikaation kehityksessä, sillä Windows Storen sertifiointi jokaiselle päivitykselle applikaatiossa voi kestää jopa useita tunteja. Kehittäjät voivat itse tehdä erilaisia hallintapaneeleja, monitorointi- tai ylläpitoportaaleja Hololensin REST -API:lla. Muun muassa Hololensin näkymän videostream on käynnistettävissä ja napattavissa HTTP-pyynnöllä GET osoitteeseen: “/api/holographic/stream/live.mp4” (Microsoft 2017e).

4 UNITY JA HOLOLENS

Unity on Unity Technologiesin kehittämä pelimoottori, jolla voidaan tehdä 2D- ja 3D-pelejä. Vaikka Unity tulkitaan vielä pelimoottoriksi, ovat sen käyttötarkoitukset kuitenkin levinneet pelien ulkopuolelle, kuten erilaisiin visualisointeihin ja opetustarkoituksiin. Näitä voi nähdä Unityn sivuilla showcase-osiossa.

Unityn vahvoina puolina pidetään sen suurta tukea eri alustoille. Tällä hetkellä Unity on ainoa suurista pelimoottoreista, johon on integroitu yhteydet Hololesin rajapintoihin. Unityä vastaavia pelimoottoreita ovat Epic Gamesin Unreal Engine, Amazonin Lumberyard ja Crytekin CryEngine. Integroidut rajapinnat mahdollistavat helpotetun ja nopean kehityksen Hololens applikaatioille Unityllä. Microsoftin Hololens applikaation kehityssarja, Holographic Academy, on myös tehty Unityllä. (Microsoft 2017f.)

Helpottaakseen integroitujen rajapintojen käyttöä, Microsoft on tehnyt HoloToolkitin Unityä varten. HoloToolkitissä on erilaisia Unity-komponentteja ja prefabeja. Prefabit ovat eri Unity-komponenteista koottuja kokonaisuuksia, joita on helppo kopioida ja muokata. Nämä komponentit hyödyntävät Hololensin toimintoja, esimerkiksi SpatialMappingObserver- ja SpatialMappingManager-komponenttien avulla voidaan päivittää Unity-applikaatioon Hololensin generoimaa spatial mappia sekä voidaan säätää päivitystaajuutta, generoidun spatial mapin polygonien tiheyttä ja materiaalia. Spatial mapping –komponentin asetukset on esitetty kuviossa 5.



KUVIO 5. Spatial Mapping -komponenttien asetukset

HoloToolkitiin on lisätty C#-luokkia, joista voidaan vaivattomasti hakea Hololensin dataa. Lisäksi luokkiin voidaan rekisteröidä omia tapahtumakuuntelijoita. Luokat löytyvät nimiavaruuden UnityEngine.VR.WSA sisältä.

Kuvion 6 mukaisen Spatial Anchor Storen, jota Unityssä kutsutaan WorldAnchorStoreksi, ja Spatial anchoreiden, eli WorldAnchoreiden, lataaminen voidaan toteuttaa kutsumalla WorldAnchorStore-luokan GetAsync-funktiota. Funktio tarvitsee syötteekseen vain callback-funktion, joka ottaa vastaan ladatun WorldAnchorStore-referenssin. Vastaanotetusta WorldAnchorStore-referenssistä voidaan sen jälkeen ladata WorldAnchoreita kutsumalla WorldAnchorStoren Load-funktiota.

```
// Use this for initialization
void Awake()
{
    WorldAnchorStore.GetAsync(StoreLoaded); //ladataan Store ohjelman käynnistyttyä
}

private void StoreLoaded(WorldAnchorStore store)
{
    this.store = store;
    LoadAnchors();
}

private void LoadAnchors()
{
    if(this.store.Load(ObjectAnchorStoreName, gameObject)) // katsotaan löytyykö storesta ObjectAnchorStoreNamen-ID:istä anchoria
    {
        anchor = this.store.Load(ObjectAnchorStoreName, rootGameObject); // asetetaan achor
        onAnchorLoaded.Invoke(); // kutsutaan onAchorLoaded-signaalia, jos latausvaiheessa halutaan tehdä muuta
    }
}
```

KUVIO 6. WorldAnchorStoren ja WorldAnchoreiden lataaminen

WorldAnchorStoreen tallentaminen voidaan tehdä kutsumalla WorldAnchorStore-instanssin Save-funktiota, jolle syötetään ID string-muodossa sekä referenssi tallennettavaan WorldAnchor-komponenttiin, kuten kuviossa 7

```
public void SaveAnchor(WorldAnchor anchorToSave)
{
        if(this.store != null)
        {
            if(store.GetAllIds().Any(id => id == ObjectAnchorStoreName))
                // poistetaan vanha WorldAnchori storesta jos semmoinen löytyy
                this.store.Delete(ObjectAnchorStoreName);
            this.store.Save(ObjectAnchorStoreName, anchorToSave); // tallennetaan uusi Anchori
        }
}
```

KUVIO 7. WorldAnchorin tallentaminen WorldAnchorStoreen

Myös Tap-eventteihin voidaan rekisteröidä omia tapahtumakuuntelijoita InteractionManager-luokan avulla kuvion 8 mukaisesti.

```
void OnEnable() // rekisteröidään eventHandlerit komponentin aktivoituessa
{
    InteractionManager.SourceDetected += InteractionManager_SourceDetected;
    InteractionManager.SourceUpdated += InteractionManager_SourceUpdated;
    InteractionManager.SourceLost += InteractionManager_SourceLost;
    InteractionManager.SourcePressed += InteractionManager_SourcePressed;
    InteractionManager.SourceReleased += InteractionManager_SourceReleased;
}

void OnDisable() // poistetaan eventHandlerit komponentin disabloituessa
{
    InteractionManager.SourceDetected -= InteractionManager_SourceDetected;
    InteractionManager.SourceUpdated -= InteractionManager_SourceUpdated;
    InteractionManager.SourceLost -= InteractionManager_SourceLost;
    InteractionManager.SourcePressed -= InteractionManager_SourcePressed;
    InteractionManager.SourceReleased -= InteractionManager_SourceReleased;
}

void InteractionManager_SourceReleased(InteractionSourceState state)
{
    if (isPressed)
    {
        isPressed = false;
        tapEventOne.Invoke(); // kutsutaan tap-eventin signaalia
    }
}
```

KUVIO 8. Esimerkki Tap-tapahtumaan rekisteröitymisestä

Hologrammien stabilisointia varten voidaan Hololensin FocusPlanea ohjata HolographicSettings-luokan funktiolla SetFocusPointForFrame. Funktion asettama FocusPlanen transformaatio on voimassa vain yhden framen ajan. Siksi sitä kehotetaan kutsumaan Unityn Update-funktion sisällä, jota Unity kutsuu jokaisen uuden framen aikana kaikilla MonoBehaviour-luokasta peräisin olevilla luokilla, jotka on aktivoitu kyseisellä hetkellä (Unity 2017b). SetFocusPointForFramelle voidaan syöttää kolme muuttujaa, jotka ovat asetettavan lokaation maailmankoordinaatit, normaali eli suunta johon taso tulee osoittamaan, ja tason nopeus. Tason normaalia ja nopeutta ei ole pakko asettaa. Esimerkki SetFocusPointForFramen käytöstä on kuviossa 9.

```
public GameObject focusedObject;
void Update()
{
    var normal = -Camera.main.transform.forward;
    // Normally the normal is best set to be the opposite of the main camera's forward vector
    // If the content is actually all on a plane (like text), set the normal to the normal of the plane
    // and ensure the user does not pass through the plane
    var position = focusedObject.transform.position;
    UnityEngine.VR.WSA.HolographicSettings.SetFocusPointForFrame(position, normal);
}
```

KUVIO 9. FocusPointin asettaminen Update-loopin sisällä

HoloToolkitiin on tehty valmiiksi esimerkki Unity-skenet eli pelikokonaisuudet, joista voidaan tutkia, kuinka komponentteja on tarkoitus käyttää.

Kehittämistä helpottamaan Unityyn on lisätty myös niin sanottu Holographic Emulation -toiminto, jonka avulla applikaatiota voidaan suorittaa Unity editorissa tietokoneella. Emulaatiomoodeja on kaksi: remoting ja simulation. Remotingillä applikaation kuva voidaan suoratoistaa Hololensille. Hololensiltä lähetetään samanaikaisesti dataa takaisin Unitylle. Tätä dataa on muun muassa Hololensin positio ja rotaatio ja erilaiset interaktiot, kuten tap tai Clickerin painallus.

Koska applikaatio ajetaan Unity Editorissa, voidaan tapahtumia skenessä seurata editorista. Eikä applikaatiota tarvitse editorissa ajamisen takia myöskään tehdä koontiversiota jokaisella kerralla kun sitä halutaan testata Hololensillä. Tämä helpottaa paljon virheiden etsintää ja testaamista

kehitysvaiheessa. Remoting vaatii kuitenkin Hololenssille asennettavan ohjelman: Holographic Remoting Playerin.

Simulation-moodilla ei tarvita Hololensiä ollenkaan, vaan Hololensin inputit, kuten pään liike ja interaktiot, on korvattu esimerkiksi joystickillä tai muulla ohjaimella (Freese 2016).

5 CASE

Opinnäytetyön projektina Valve Groupissa toteutettiin tietoliikennealan yhtiölle Nokia Oyj:lle Hololens-applikaatio Mobile World Congress 2017 -messuja varten. Demon haluttiin kuvastavan tilannetta, jossa käyttäjän piti älylasien opastuksen mukaan pystyä seuraamaan ohjeita tietystä alkutilanteesta tapahtuman päätepisteeseen.

Demon tärkein osuus oli tilanne, jossa olemassa olevan palvelimen päälle heijastettiin siihen liittyvää dataa, kuten CPU:n ja kytkimen käyttöaste ja kovalevyjen täyttyminen. Informaation ei tarvinnut olla todellista. Keksitty data, joka muistutti oikeita arvoja, oli riittävä.

Demossa käytettiin OCP-räkkiä. OCP tarkoittaa Open Compute Projectia. Se on yhteisö, joka jakaa datakeskussuunnitelmia yritysten kesken. Sen perustivat vuonna 2011 Facebook, Intel, Rackspace, Goldman Sachs ja Sun Microsystemsin omistaja Andy Bechtolsheim toteuttaakseen laitteistoille vastaavaa avoimuutta kuin on toteutettu avoimena lähdekoodina ohjelmistokehityksessä (Open Compute Project 2017.)

Nokian OCP-räkissä, jonka päälle data heijastettiin, oli neljä erilaista komponenttia: kytkin, virtalähde, kovalevyt ja palvelin. Havainnollistavan datan lisäksi palvelimen päälle haluttiin lisätä opettavainen osanvaihto-animaatio, jonka avulla kuka tahansa osaisi vaihtaa kyseisen osan.

Koska tiedettiin, että messuilla demoa kävisi katsomassa suuri määrä ihmisiä, haluttiin kokemuksen olevan nopea. Tästä syystä jo varhain demoa hahmoteltaessa päädyttiin visioon, jossa hologrammitekoäly ohjaisi käyttäjän demon läpi. Ja koska kädentunnistuksella tehtävä tap-ele olisi ensikertalaiselle todella vaikea käyttää, päädyttiin sen sijaan käyttämään Hololensin mukana tullutta Clickeriä.

Demossa tuli olla kolme osaa: ensimmäisessä osassa näytettäisiin miniatyyri palvelinhuone, toisessa palvelin, jonka päälle oli lisätty siihen liittyvä informaatio ja viimeiseksi terminaali, josta palvelimen eri osia voitaisiin katsoa tarkemmin.

Demo toteutettiin käyttäen pelimoottori Unityä ja monia 3D-mallinnussovelluksia, kuten 3D Studio Maxia, Blenderiä, Substance Painteria, sekä 2D-grafiikan luomiseen Adobe Photoshopia. Demon tekemisen aikataulu oli nopea. Sen tekeminen aloitettiin joulukuun lopulla ja sen tuli olla valmis helmikuun 27. päivänä, jolloin messut alkoivat.

5.1 Hologrammien grafiikan luominen

Hololensissä esiintyvät hologrammit ovat 3D-malleja 3D-maailmassa. Ihmisen näön syvyydentunnistus on mahdollista kahden silmän eri perspektiivien luomasta kuvasta. Tästä syystä myös Hololensin hologrammit näyttävät sulautuvan oikeaan maailmaan, koska siitä projisoidaan molempiin silmiin kahdesta eri perspektiivistä kuvattua kuvaa virtuaalimaailmasta. Edellä mainitusta syystä hologrammien on pakko olla sidottuna virtuaalimaailman 3D-avaruuteen, eikä grafiikkaa voida lisätä kuvan päälle kuten 2D-näytöltä katsottuna.

Räkin päälle lisätty data oli kuitenkin tärkein ominaisuus koko demossa, ja koska suurimmalta osin data oli vain tekstiä ja graafeja, päätettiin siitä tehdä kaksiulotteinen 3D-taso. Tasossa ei siis ollut syvyyttä, mutta siinä oli paikka ja rotaatio 3D-avaruudessa. Räkin päälle lisätty data on kuvattu kuviossa 10.

KUVIO 10. Kuvakaappaus räkin päälle asetetusta datasta Hololensin näkökulmasta.

Messuilla näytettävästä räkistä saatiin kuva, jossa komponentit olivat paikallaan, eikä niiden paikkaa muutettu ennen messuja tai niiden aikana. Räkin ja komponenttien mittatietojen perusteella tehtiin räkin kokoinen 3D-taso, jonka päälle asetettiin komponenttien kohdalle niistä kertovat tekstit ja graafit.

Teksteihin ja graafeihin lisättiin satunnaisesti generoitua dataa, jonka ylä- ja ala-arvot olivat kuitenkin komponentille ominaisia; esimerkiksi serverin hyötykäyttö, jonka ylä- ja ala-arvoina olivat 100 ja 0 prosenttia. Hololensin linssien läpi kirkasta informaatiota oli kuitenkin vaikea välillä erottaa räkistä vaaleaa taustaa vasten. Niinpä informaation taustalle luotiin kuvanmuokkausohjelma Photoshopilla tausta, jossa hyödynnettiin asiakkaan graafista ilmettä.

Hololens ei voi toistaa tummia värejä koska mustaa valoa ei ole. Tämän takia musta väri on aina Hololensissä läpinäkyvää. Positiivisena puolena mustan värin läpinäkyvyydessä Hololens applikaatiossa oli, että läpinäkyviä shadereita varten voitiin käyttää oikeasti läpinäkymätöntä

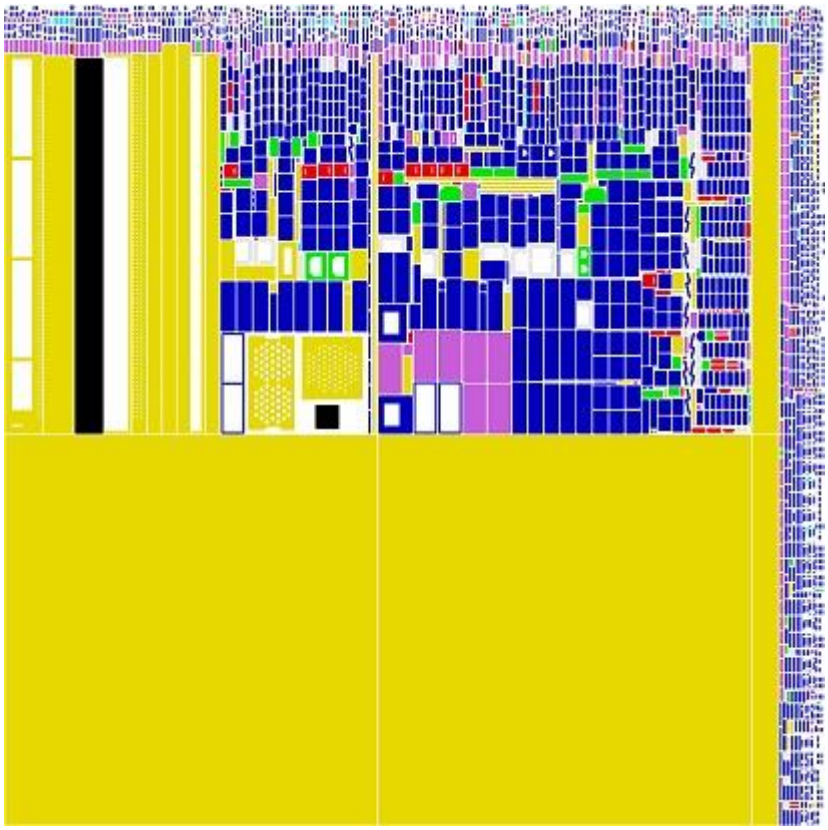
shaderia. Läpinäkyvyys shadereissa vaatii enemmän prosessointia kuin läpinäkymätön, joten tämä toimi myös optimointina piirtoaikaan nähden (Unityperformance 2016.)

Räkin komponentteja ja niiden spesifikaatioita haluttiin pystyä katsomaan myös tarkemmin. Tätä varten kehitettiin toinen hologrammi, joka toimi eräänlaisena terminaalina. Tästä terminaalista voitiin valita komponentti, jonka 3D-malli lisättiin näkyville ja komponentin spesifikaatiot piirrettiin tekstinä komponentin lähettyville 3D-tasoon. Pohjat 3D-malleille saatiin aiemmasta projektista, jota varten mallit oli alunperin tehty.

Mallit olivat kuitenkin liian raskaita Hololensille, joten niitä piti optimoida. Optimointi aloitettiin vähentämällä askelmia interpolaatioista kaikista reunoja pehmentävistä modifikaattoreista 3D-malleissa, joista niitä vielä voitiin poistaa. Samalla vähennettiin iteraatioita kaikista

TurboSmoothin omaavista malleista, tai poistettiin TurboSmooth kokonaan. TurboSmooth jakaa mallin geometrian pienempiin osiin ja pehmentää sen kulmia. TurboSmooth lisää geometrian tarkkuutta, joka laskee mallin sisältämän ohjelman tehokkuutta (Autodesk 2016.) Tämän jälkeen mallit siirrettiin Blenderiin, jossa optimointi jatkui poistamalla ylimääräistä geometriaa, kuten edgelooppeja, eli mallia ympäröiviä reunuksia ja kansien alla piilossa ollutta geometriaa.

Ruuvien ja niittien geometria poistettiin ja ne korvattiin Normal-mapeilla. Kun mallin geometria oli valmis, malliin tehtiin UV-mappaus ja tehtiin omat materiaali-ID:t eri materiaaleille, esimerkiksi metallille ja muoville. Kyseisille materiaali-ID:ille lisättiin eri värit ja väreistä renderöitiin materiaali-ID-bittikartta. Kun materiaali-ID-bittikartta oli renderöity, voitiin 3D-mallista poistaa kaikki muut paitsi yksi materiaali-ID.



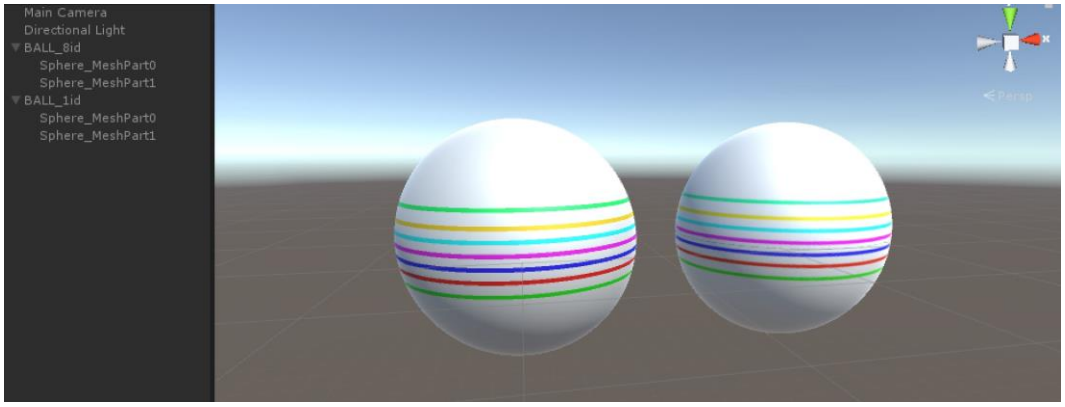
KUVIO 11. Materiaali-ID-bittikartta.

Kuviossa 11 on materiaali-ID-bittikartta, jossa keltainen on maalattua metallia, sininen maalaamatonta metallia, vihreä kiiltävää mustaa muovia, punainen oranssia muovia ja pinkki mattamustaa muovia. Materiaali-ID-bittikartan luomisen jälkeen malli tallennettiin Blenderistä fbx-muotoon ja tuotiin Substance Painteriin, jossa materiaali-ID-bittikartan mukaan voitiin lisätä oikeat materiaalit oikeisiin kohtiin 3D-mallia yhdessä materiaalikanavassa. Materiaalit vietiin Substance Painterista Unityyn tekstuureina, joita olivat diffuse-, roughness-, metalness- ja normal-bittikartat. Esimerkki maalatusta komponentistä on kuviossa 12.

KUVIO 12. Substance Painterissa maalattu komponentti.

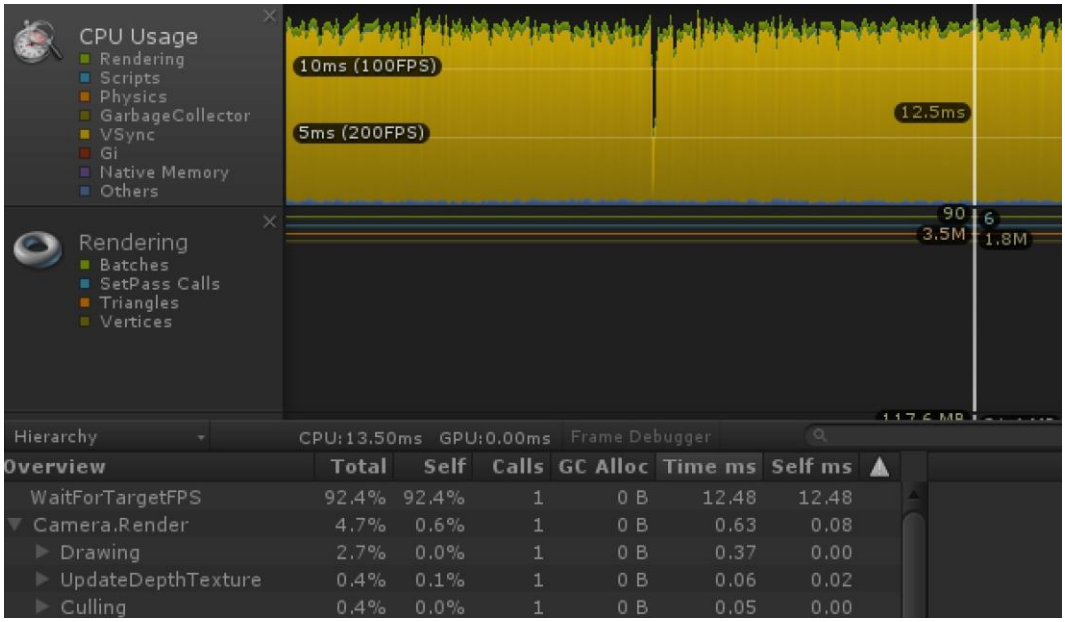
3D-malli, jossa oli erilaisia materiaaleja, voitiin toteuttaa Unityssä hyödyntäen vain yhtä materiaalikanavaa. Syy siihen, miksi haluttiin käyttää vain yhtä materiaalikanavaa, oli että jokaista materiaalikanavaa kohden 3D-moottorit joutuvat tekemään uuden niin sanotun drawcallin, vaikka kysessä olisi sama 3D-malli.

Drawcallien suuri määrä vaikuttaa hidastaa kuvanpäivitysnopeutta, joka taas vaikuttaa negatiivisesti applikaation käyttökokemukseen. Väitöksen todistamiseksi suoritettiin testi, jossa tehtiin kaksi 3D-palloa, jotka näkyvät kuviossa 13, oli kahdeksan eri väriä. Toisessa palloista kaikki värit omasivat oman materiaali-ID:nsä ja toisessa värit olivat renderöity 4096x4096-kokoiseen bittikarttaan, jota käytettiin kyseisen pallon ainoan materiaalin värikarttana. Mallit vietiin Unityyn ja pallo kerrallaan ajettiin Unityn profilointityökalua ja tarkkailtiin drawcalleja ja kameran renderöintiaikaa.

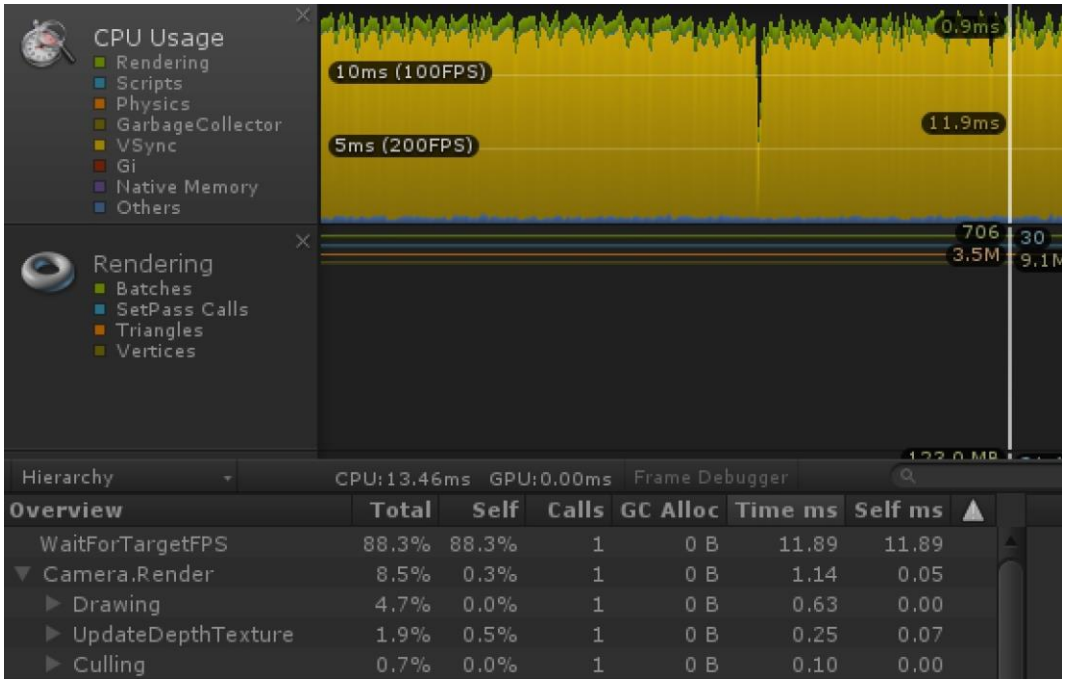


KUVIO 13. Kuvakaappaus testattavista palloista. Vasemmalla kahdeksan materiaalikanavan malli ja oikealla yhden.

Kahdeksan materiaalikanavan omaavan mallin piirtyessä drawcallien määrä oli 66, kun taas yhden materiaalikanavan omaavan piirtyessä sama lukema oli 10. Koska yhden mallin piirtyessä drawcallien määrä ei vaikuttanut piirtonopeuteen paljoa, lisättiin molempia palloja 10 kappaletta ja profiloinityökalu ajettiin uudelleen. Kun yhtätoista kahdeksan materiaalikanavan omaavaa palloa profiloitiin, saatiin kameran renderöintiajaksi keskiarvoltaan noin 1.13 millisekuntia. Yhdentoista kappaleen yhden materiaalikanavan omaavan pallon renderöintiajaksi saatiin noin 0.64 millisekuntia. Kuvakaappaukset profilointityökalun käytöstä ovat kuvioissa 14 ja 15.



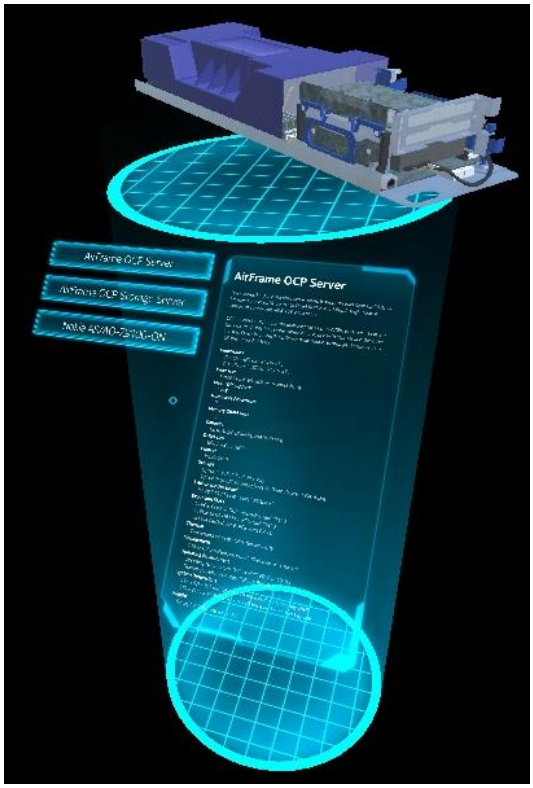
KUVIO 14. Profilointidataa yhden materiaalikanavan omaavan pallon näkyessä



KUVIO 15. Kahdeksan materiaalikanavan sisältävän pallon profilointidataa

Komponenttien terminaalin alustalle tehtiin erillinen grafiikka 3D-tasoista, joihin lisättiin additiivinen läpinäkyvä shaderi, jota käytettiin myös tekstien taustana. Alustan taustakuva vaihdettiin ympyrämäiseen ruutukuvioon. Alustan viereen luotiin nappulat, joiden painallukset ohjelmoitiin

vaihtamaan katseltava komponentti ja siihen liittyvä teksti. Valmis versio mallienkatseluterminaalista on kuvattuna kuviossa 16.



KUVIO 16. Lopputulos mallienkatseluterminaalista.

5.2 Hologrammien asettelu ja anchorit

Hologrammien asetteluun valittiin aluksi AR-kehitykseen keskittynyt alusta Vuforia. Vuforialla voidaan asettaa 3D-malli kuvan päälle. Näitä kuvia kutsutaan image targeteiksi.

Vuforiassa on myös Unityn ja Hololensin tuki. Image targetit määritellään selaimen kautta Vuforian kehittäjäportaalissa. Unity-paketti, josta tunnistettavat kuvat ja niiden koot löytyvät, ladataan Unityyn. Vuforialla ja sen kuvantunnistuksella pystyttiin asettamaan tulostettu kuva siihen paikkaan oikeassa maailmassa, johon hologrammin haluttiin asettuvan.

Hologrammien asettelua varten kehitettiin asettelumoodi, joka tuli laittaa päälle aina kun hologrammeja haluttiin liikuttaa. Aluksi ideana oli lähettää

Hololensille WLANin kautta virtuaalisyötteitä Hololensin omasta laiteportaalista (Microsoft 2017g), mutta koska WLAN nähtiin riskialttiina sen epävarmuuden vuoksi messutilanteissa, päätettiin asettelumoodin laukaisu sisällyttää itse ohjelmaan jollain piilotetulla tavalla.

Koska itse Hololensissä on rajattu määrä interaktiotapoja, piti ohjelmoida monta erilaista tap-elettä. Jokaiseen painallukseen ohjelmoitiin 250:n millisekunnin viive, jonka aikana voitiin tehdä uusi painallus. Niin kauan kuin painalluksia tapahtui, painallusten määrää kasvatettiin ja kyseisen painallusmäärän funktio laukaistiin, jos 250:n millisekunnin kynnys ylitettiin.

Tällä tavoin toteutettiin muun muassa 3-klikkauksen tapahtuma, johon aluksi asettelumoodin laukaisu asetettiin. Kun asettelumoodi oli päällä, liikuteltava hologrammi valittu ja Hololens tunnisti kuvan, asetettiin valittu hologrammi kuvan kohdalle. 250:n millisekunnin viive jokaisessa klikkauksessa johti kuitenkin siihen, että kyseiset 250 millisekuntia jouduttiin odottamaan jopa normaalissa demokäytössä, esimerkiksi kun näpäytettiin Clickeriä painaakseen nappulaa. Se häiritsi demon sulavuutta ja käyttökokemusta.

Alle 250:n millisekunnin mittaisella kynnyksellä klikkaaminen useita kertoja oli vaikeaa. Täten kynnyksestä luovuttiin ja tilalle kehitettiin kymmenen sekuntia kestävä pitkä painallus. Jos painallus kesti yli kymmenen sekuntia, asettelumoodi laukaistiin. Kaikki alle 10 sekuntia kestäneet painallukset olivat normaaleja painalluksia.

Kun demoa kehitettiin, Vuforia päätettiin ottaa pois, koska lisenssin saaminen ajoissa oli epävarmaa. Tilalle kehitettiin asetteluvalikko, josta valittu hologrammi voitiin liittää Hololensiin, pyöräyttää se Y-akselin mukaan osoittamaan Hololensiä, asettaa hologrammi maahan ja liikutella ja kääntää sitä manuaalisesti. Pikainen hologrammien asettelu voitiin näillä työkaluilla toteuttaa esimerkiksi kiinnittämällä ensin hologrammi Hololensiin, asettamalla Hololens haluttuun paikkaa, esimerkiksi: pöydälle tai lattialle, ja sen jälkeen irroittamalla hologrammi Hololensistä

näpäyttämällä Clickeriä ja rotatoimalla se haluttuun katselukulmaan liikkumalla itse Hololensin kanssa kulmaan, josta hologrammia haluttiin katseltavan ja painamalla toimintoa, joka käänsi hologrammin osoittamaan kohti Hololensiä. Hienosäätöä varten voitiin käyttää manuaalisia siirto-ominaisuuksia, joilla hologrammia voitiin liikuttaa tai pyörittää X-, Y- tai Z-koordinaattien tai akseleiden myötäisesti.

Jottei hologrammeja tarvittu jokaisella käynnistyskerralla asetella uudestaan, lisättiin objektiin spatial anchor -komponentti ja se tallennettiin spatial anchor storeen, josta applikaation uudestaan käynnistyttyä ne voitiin taas ladata.

5.3 Demon tarinankerronta ja käyttäjäystävällisyys

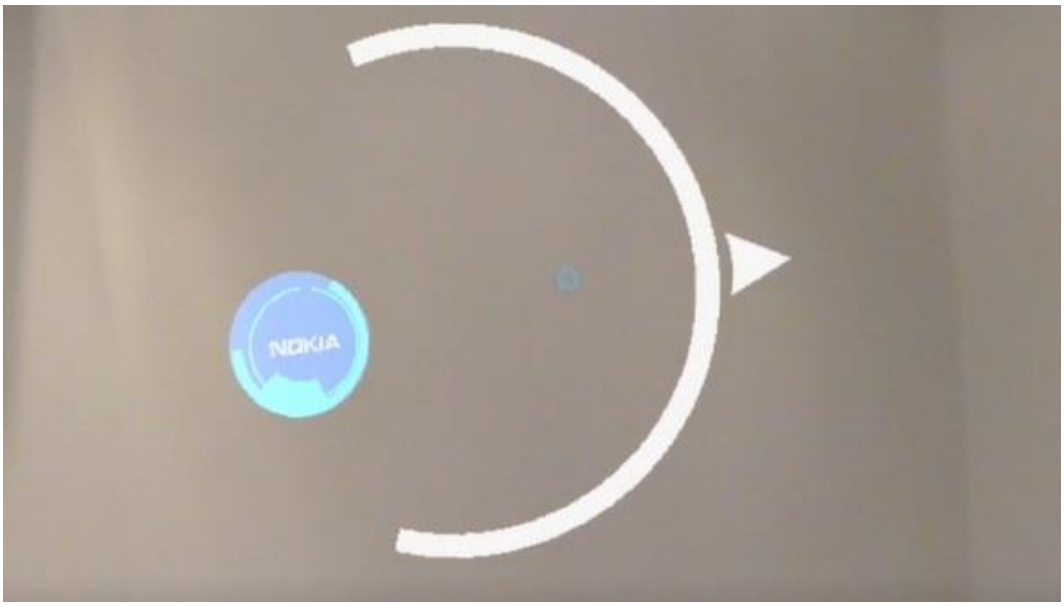
Demon tarina jakautui kolmeen osaan. Ensimmäisessä osassa näytettiin ja kerrottiin palvelinsalista, jossa demonkäyttäjä kuvitteellisesti oli. Toisessa osassa esitettiin yksi palvelimista, jonka vaihdettavasta kovalevystä tuli tehdä ilmoitus ja jonka yksi palvelimen komponenteistä tuli vaihtaa. Kun kovalevyn ilmoitus oli lähetetty ja komponentti vaihdettu, siirryttiin kolmanteen osaan, jossa esiteltiin OCP-räkin komponentteja. Demossa haluttiin käyttää mahdollisimman paljon Hololensin ominaisuuksia, kuten liikkumista virtuaaliympäristössä. Siksi demon kaikki kolme osaa levitettiin eri puolille esitystilaa, minkä vuoksi käyttäjän piti liikkua. Demosta piti pystyä myös suoriutumaan noin kahdessa minuutissa, sillä kokeilijoita saattoi olla jopa satoja päivässä. Demoa ohjaamaan tehtiin kuvion 17 mukainen tekoälyhahmo, joka johdatti käyttäjän vaadittuun paikkaan ja kertoi tälle, mitä piti tehdä.

KUVIO 17. Kuvakaappaus käyttäjää ohjaavasta tekoälyhahmosta

Kun demoa näytettiin pilotoinnin aikana, huomattiin, että Hololens-älylasit oli vaikea asettaa päähän. Hololensin pienen näkökentän vuoksi ensikertalaisten oli myös vaikea löytää oikea asento nähdäkseen hologrammit. Siksi demon alkuun päätettiin tehdä koko näytön peittävä ruutu, jonka keskellä luki kirkkain kirjaimin teksti, joka kehotti käyttäjää asettamaan Hololensin silmille niin, että hän näkisi kaikki neljä kulmaa. Alkuruutu takasi, että käyttäjä oli jo alusta alkaen asettanut Hololensin päähänsä oikein.

Ensimmäisillä pilotointikierroksilla huomattiin, että käyttäjät hukkasivat tekoälyn usein eivätkä täten enää tienneet, minne heidän oli tarve katsoa tai hakeutua milläkin hetkellä. Ratkaisuksi ongelmaan kehitettiin kuvion 18 mukainen suuntaindikaattori. Se kiinnitettiin keskelle käyttäjän ruutua ja sen muodoksi valittiin ympyrä, jotta se ei peittäisi näkökenttää. Ympyrän reunaa kiersi nuoli, joka indikoi suuntaa, johon käyttäjän tuli katsoa. Kun katsottava piste oli Hololensin näkökentässä, suuntaindikaattori poistettiin. Kun suuntaindikaattori oli lisätty, ensikertalaisten läpipääsy demossa ilman

ulkopuolisen apua parani huomattavasti.



KUVIO 18. Kuvakaappaus suuntaindikaattorista Hololensin näkökulmasta.

Demossa oli useita painettavia nappuloita. Hankaluudeksi muodostui pilotoinnin ensikertalaisten tietämättömyys siitä, kuinka nappuloita oli tarkoitus painaa. Demon nappulat oli kehitetty painettavaksi samalla tavalla kuin Hololensin omassa käyttöjärjestelmässä, eli kursori kiinnitettiin keskelle näyttöä ja sitä ohjattiin katseella. Painallus tapahtui liikuttamalla kursoria katseen avulla painettavan nappulan päälle ja näpäyttämällä Clickeriä kädellä. Jotkut ensikertalaiset yrittivät käyttää kättään kursorin liikuttamiseen. Tämän välttämiseksi todettiin parhaaksi selittää oikea käyttöperiaate demon alussa. Demon alkuun tehtiin aloitusruudun jälkeen tutoriaali, joka oli naamioitu kyselyksi. Tekoäly pyysi heti demon alettua käyttäjää seuraamaan suuntaindikaattoria ja vastaamaan kysymykseen. Kysymys ohjelmoitiin siten, että siihen vastatakseen käyttäjän oli aina käännyttävä oikealle. Näin käyttäjä ymmärsi, kuinka suuntaindikaattoria seurattiin ja löysi ensimmäisen suoritettavan tehtävän.

Ensimmäisenä tehtävänä oli vastata kysymykseen oliko kyseessä käyttäjän ensimmäinen kerta Hololensin parissa. Teksti opasti, kuinka kysymykseen vastattaisiin: kysymyksen alapuolella oli kaksi nappulaa:

“yes” ja “no”. Kun käyttäjä oli painannut jompaa kumpaa näistä nappuloista, hänen katseensa ohjattiin takaisin tekoälyyn ja itse kierros alkoi. Tämän pikaisen tutoriaalin avulla miltei jokainen testikäyttäjä läpäisi koko demon ilman lisäapua.

Koska Hololensin syötetapoja oli vähän ja koska WiFiin ei haluttu tukeutua, eikä sen vuoksi voitu käyttää Hololensin REST-rajapintaa, demon resetointi jouduttiin improvisoimaan. Resetoinnissa ei haluttu luottaa siihen, että käyttäjä painaisi viimeistä nappulaa demon lopussa ja koska esityspituuden epävarmuuden vuoksi ei voitu käyttää myöskään ajastinta, resetoiminen toteutettiin kääntämällä Hololens ylösalaisin.

Unityssä tunnistus siitä milloin Hololens oli väärinpäin, saatiin laskemalla pistetulo kameran maailmankoordinaatiston ylöspäin suuntaavaan vektorin ja alavektorin 0,-1,0 välillä. Pistetulolla voitiin selvittää vektoreiden yhdensuuntaisuus. Jos ylä ja alavektorin pistetulo oli lähes 1, voitiin todeta, että Hololens oli väärinpäin. Sama ominaisuus lisättiin myös hologrammien asettelumoodiin hätävaraksi, jossa kaikki hologrammit asettuivat Hololensin kohdalle, kun se käännettiin ylösalaisin. Headsetin kääntö ylösalaisin demon uudelleenkäynnistämiseksi oli nopea ja melko huomaamaton toimenpide.

5.4 Havaintoja lopputuotteesta

Hololens-demo esiteltiin Mobile World Congress 2017 -messuilla rajatulle yleisölle Nokian suljetulla näyttelyalueella. Demoa kokeili jopa sata henkilöä päivässä.

Heti ensimmäisenä päivänä huomattiin, että demo kesti liian kauan. Ensikertalaiselle kierros demossa kesti noin 4 minuuttia. Jos esimerkiksi viiden henkilön ryhmä halusi katsoa demon, kokonaisajaksi tuli noin 20 minuuttia, mikä oli liian pitkä aika. Siksi demosta päätettiin poistaa ensimmäinen osio, jossa esitettiin tutoriaali ja näytettiin palvelinhalli. Päivityksen tekeminen oli helppoa, sillä applikaatioon oli jätetty ominaisuus, jolla pystyttiin hyppäämään kolmeen eri vaiheeseen demon

kulussa.

Alun perin tämä toiminto oli kehitetty turvaamaan tapahtuma, jossa demo tuntemattomasta syystä olisi kaatunut, tai Hololensin akku olisi loppunut. Toimintoa käytettiin Hololensin puheentunnistuksella ja se implementoitiin Unityssä käyttäen HoloToolkitin mukana tullutta KeywordRecognizer-komponenttia. Hololens tunnisti demon alussa lausahdukset: “Stage one” “-two” ja “-three”. Näillä voitiin ladata demo halutusta kohdasta. Demon aloitus palvelinräkin, eli stage threen, kohdalta pudotti kierroksen pituuden noin kahteen minuuttiin.

Ensikertalaisten kanssa ongelmaksi ilmaantui myös käyttäjän ohjaaminen demon maailmassa. Hololensin suoratoistoa ei haluttu pitää päällä siitä syystä, että tämä laski Hololensin kuvanlaatua. Kuvattaessa Hololensin näkymää sen REST-rajapinnan kautta Hololens yrittää kompensoida vaaditun suoritustehon poistamalla esimerkiksi reunanpehmennykset kuvasta. Ilman reunanpehmennyksiä tekstistä tuli vaikealukuista.

Koska suoratoistoa ei haluttu käyttää, ei voitu helposti tietää, missä vaiheessa demoa kävijä liikkui. Tämän vuoksi esittelijän oli vaikea kertoa oikeasta asiasta oikeaan aikaan demon kuluessa. Kun asiaa mietittiin, ymmärrettiin, että puhetta ja eri audioklippejä, joita demossa soitettiin, esimerkiksi nappulan painalluksen seurauksena, voitiin kuulla ulkopuolelta. Täten demon esittelijä oppi kuuntelemaan vierestä, missä kohdassa demoa käyttäjä oli menossa.

Jatkokehityksen kannalta tämä voitaisiin ottaa huomioon ja esimerkiksi toistaa demon esittelijälle käyttäjän audio vaikkapa nappikuulokkeeseen. Parhaana ratkaisuna olisi kuitenkin esittelijän oma Hololens, jolla tämä näkisi saman maailman, jonka käyttäjäkin. Se vaatisi kuitenkin tiedon välittämistä yhdeltä demon instanssilta toiselle. Tämä taas vaatisi LAN-verkkoa, johon ei haluttu demossa turvautua.

Myös tapahtumien kertominen instanssilta toiselle, olisi vaatinut paljon enemmän aikaa demon kehittämiseen ja sitä ei ollut.

Demoa esitettäessä keksittiin jatkokehitysidea, jossa esittelijä pitää Hololensin Clickerin itsellään koko ajan. Esittelijä voi Clickeriä painamalla siirtyä demossa seuraavaan kohtaan. Tämän avulla esittelijä tietää tarkasti, mikä tapahtuma on kyseisellä hetkellä menossa.

Hololens-demo onnistui odotettua paremmin. Ennen messuja oli pelätty mahdollisuutta, että Hololens saattaisi hukata tiedon skannatusta tilasta väenpaljouden takia. Jos kyseinen skenaario olisi toteutunut, olisivat hologrammien paikat nollautuneet. Näin ei kuitenkaan käynyt, vaikka Hololensin ympärillä saattoi olla kymmeniä ihmisiä.

Toinen skenaario, jota pohdittiin, oli spatial anchoreiden paikkojen muuttuminen radikaalisti. 3D-mallien ja palvelinhallin hologrammien kanssa tämä ei olisi haitannut, mutta palvelimen päälle sijoitetun UI:n piti olla sentilleen oikeassa kohdassa, jotta se näytti uskottavalta. Hologrammit pysyivät yllättävän hyvin paikallaan koko messujen ajan.

Hologrammien paikat tarkistettiin päivittäin. Useimmiten vain yhtä kolmesta jouduttiin säätämään.

6 YHTEENVETO

Hololens-älylasit ovat tämän tutkimuksen kirjoitushetkellä vasta ottamassa ensiaskeliaan suuren yleisön tietoisuuteen. Minimissään 3 000:n dollarin hinta pitää ne toistaiseksi vain harvojen saatavilla.

Messujen aikana esitetty demo osoitti, että kiinnostusta Hololens-älylaseja kohtaan on paljon ja käyttömahdollisuuksia teknologialle on monia. Kävijäpalaute oli hyvin positiivista, ja demo sai aikaan ilmiselvän wow-efektin. Koko messuilla Hololens-demoja oli vain puolisen tusinaa. Tulevaisuudessa tämä tulee kuitenkin muuttumaan, koska markkinoille on tulossa kilpailevia valmistajia, jolloin hintakin tulee laskemaan.

Hololens-älylasien ominaisuudet vaativat tietysti vielä kehittämistä ollakseen todella käyttäjäystävälliset. Esiteltäessä Hololens-älylaseja suurelle yleisölle messutapahtuman aikana huomattiin, että niiden fyysinen käyttö vaatii totuttelua. Älylasit puetaan ylle kiristettävän pannan avulla ja katseen oikea kohdistaminen vie hiukan aikaa.

AR:n ja sen myötä Hololens-tyyppisten älylasien käyttökohteet ovat rajattomat. AR tulee tekemään disruptiivisen muutoksen esimerkiksi navigointiin. Vaikka jo nyt kartan lukeminen ja navigointiohjeiden vastaanottaminen puhelimeen on arkipäivää, olisi se silti helpompaa suoraan katseella lasien läpi.

Esimerkiksi metrotunneleissa navigointi voi olla jopa mahdotonta älypuhelimella, koska sen paikannus perustuu GPS:ään, jonka signaali ei välttämättä kulje paksun kallion läpi. Toisaalta myöskään sisätiloja ei ole kovinkaan tarkasti kartoitettu. Näihin tapauksiin Hololensin koko ajan reaaliajassa ympäristöä tarkkaileva ja ympäristön tunnistusta päivittävä teknologia sopisi täydellisesti.

Nykyisenkaltaisen Hololensin korkean hinnan ja suuren koon takia Hololens itsessään tuskin tulee olemaan läpimurto, mutta sen toteuttama ensiaskel luultavasti sytyttää kipinän, jonka kautta AR-headsettien päätyminen suuren yleisön hyödynnettäväksi tulee tapahtumaan

LÄHTEET

Painetut lähteet:

Barfield, W. 2015. Fundamentals of Wearable Computers and Augmented Reality, Second Edition. Boca Raton: CRC Press.

Chrysanthou, Y., Slater, M. & Steed, A. 2002. Computer Graphics and Virtual Environments. Harlow: Pearson Education Limited.

Hughes, L. M. & Wright, J. S. 2010. Computer Animation. New York: Nova Science Publishers, Inc.

Kipper, G. & Rampolla, J. 2012. Augmented Reality An Emerging Technologies Guide to AR. Waltham: Elsevier.

Steinicke, F. 2016. Beign Really Virtual: Immersive Natives and the Future of Virtual Reality. Cham: Springer Nature.

Elektroniset lähteet:

ARToolKit 2015. ARToolKit Documentation [viitattu 10.3.2017]. Saatavissa: https://artoolkit.org/documentation/

Augmented Reality 2017 [viitattu 12.2.2017]. Saatavissa: http://www.sweethaven02.com/PDF_Lifelong/Augmented%20reality.pdf

Autodesk 2016. TurboSmooth Modifier [viitattu 14.4.2017]. Saatavissa: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-EA8FF838-B197-4565-9A85-71CE93DA4F68-htm.html

Brownlee Marques. 2013. Google Glass Explorer Edition: Explained! [viitattu 7.2.2017]. Saatavissa: https://www.youtube.com/watch?v=elXk87IKgCo

CB Insights. 2017. AR/VR Sets new records for deals and dollars in 2016 [viitattu 1.3.2017]. Saatavissa: https://www.cbinsights.com/blog/ar-vr-startup-funding/

CloudTweaks. 2014. Will virtual reality replace the virtual desktop? [viitattu 5.4.2017]. Saatavissa: https://cloudtweaks.com/2014/10/will-virtual-reality-replace-virtual-desktop/

Digi-Capital. 2016. AR/VR investment hits $1.7 billion in last 12 months [viitattu 1.3.2017]. Saatavissa: http://www.digi-capital.com/news/2016/04/arvr-investment-hits-1-7-billion-in-last-12-months/#.WJorxBuLREY

Fitzsimmons, M. 2015. Microsoft HoloLens detailed: see-through screen, spatial sound and sensors [viitattu 5.3.2017]. Saatavissa: http://www.techradar.com/news/wearables/microsoft-hololens-detailed-see-through-screen-spatial-sound-and-sensors-1292602

Freese, P. 2016. Introducing holographic emulation [viitattu 18.3.2017]. Saatavissa: https://blogs.unity3d.com/2016/09/29/introducing-holographic-emulation/

Guttag, K. 2016. AR/MR Combiners Part 2 – Hololens [viitattu 20.3.2017]. Saatavissa: http://www.kguttag.com/2016/10/27/armr-combiners-part-2-hololens/

Holmdahl, T. 2015. BUILD 2015: A closer look at the Microsoft Hololens hardware [viitattu 5.3.2017]. Saatavissa: https://blogs.windows.com/devices/2015/04/30/build-2015-a-closer-look-at-the-microsoft-hololens-hardware/#cidWdTsrpg5w5Bbr.97

Kohler, C. 2010. Virtual Boy, Nintendo’s big 3-D flop, turns 15. [viitattu 29.1.2017]. Saatavissa: https://www.wired.com/2010/08/virtual-boy/

Kumparak, G. 2014. A brief history of Oculus [viitattu 5.2.2017]. Saatavissa: https://www.wired.com/2010/08/virtual-boy/

Microsoft 2017a. Mixed reality: Your world is the canvas [viitattu 1.4.2017]. Saatavissa: https://www.microsoft.com/en-us/hololens

Microsoft 2017b. Coordinate Systems [viitattu 1.4.2017]. Saatavissa: https://developer.microsoft.com/en-us/windows/mixed-reality/coordinate_systems

Microsoft 2017c. Gestures [viitattu 1.4.2017]. Saatavissa: https://developer.microsoft.com/en-us/windows/mixed-reality/gestures

Microsoft 2017d. Hologram Stability [viitattu 1.4.2017]. Saatavissa: https://developer.microsoft.com/en-us/windows/mixed-reality/hologram_stability

Microsoft 2017e. Device Portal API reference [viitattu 1.4.2017]. Saatavissa: https://developer.microsoft.com/en-us/windows/mixed-reality/device_portal_api_reference

Microsoft 2017f. Holographic Academy [viitattu 1.4.2017]. Saatavissa: https://developer.microsoft.com/en-us/windows/mixed-reality/academy

Microsoft 2017g. Using the Windows Device Portal [viitattu 1.4.2017]. Saatavissa: https://developer.microsoft.com/en-us/windows/mixed-reality/using_the_windows_device_portal

Microsoft 2017h. Hololens Hardware details [viitattu 1.4.2017]. Saatavissa: https://developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details

Microsoft 2017i. Persistence in Unity [viitattu 1.4.2017]. Saatavissa: https://developer.microsoft.com/en-us/windows/mixed-reality/persistence_in_unity

Milgram, P., Takemura, H., Utsumi, A. & Kishino, F. 1994 [viitattu 1.4.2017]. Saatavissa: http://etclab.mie.utoronto.ca/publication/1994/Milgram_Takemura_SPIE1994.pdf

Open Compute Project. 2017. About OCP [viitattu 7.4.2017]. Saatavissa: http://www.opencompute.org/about/

Rieder David. 2011 [viitattu 10.2.2017]. Saatavissa: http://swimdr549.blogspot.fi/2011/07/

Roberson Museum and Science Center. 2000. The Link Flight Trainer [viitattu 28.1.2017]. Saatavissa: https://www.asme.org/getmedia/d75b81fd-83e8-4458-aba7-166a87d35811/210-Link-C-3-Flight-Trainer.aspx

The father of virtual reality [viitattu 9.2.2017]. Saatavissa: http://www.mortonheilig.com/

Turi, J. 2014. The sights and scents of the Sensorama simulator [viitattu 9.3.2017]. Saatavissa: https://www.engadget.com/2014/02/16/morton-heiligs-sensorama-simulator/

Unity. 2017. Unity Documentation [viitattu 1.4.2017]. Saatavissa: https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html

Unityperformance. 2016. Built in shader performance (Desktop – 750m) [viitattu 26.3.2017]. Saatavissa: http://unityperformance.com/guide/builtin-shaders-desktop-750m

We’re graduating from Google[x] labs [viitattu 7.2.2017]. Saatavissa: https://plus.google.com/+GoogleGlass/posts/9uiwXY42tvc

Wikipedia 2017a. Google Glass [viitattu 7.2.2017]. Saatavissa: https://en.wikipedia.org/wiki/Google_Glass

Wikipedia 2017b. Holography [viitattu 1.4.2017]. Saatavissa: https://en.wikipedia.org/wiki/Holography

Wikipedia 2017c Microsoft Hololens [viitattu 5.3.2017]. Saatavissa: https://en.wikipedia.org/wiki/Microsoft_HoloLens