Lassi Kesäläinen

Lisätyn todellisuuden kehitys

Magic Leap -laiteympäristössä

Insinööri (AMK)

Tieto- ja viestintätekniikka

Syksy 2019

KAMK • University of Applied Sciences

Tiivistelmä

Tekijä(t): Kesäläinen Lassi

Työn nimi: Lisätyn todellisuuden kehitys Magic Leap -laiteympäristössä

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: Lisätty todellisuus, Magic Leap, Unreal Engine 4

Opinnäytetyö toteutettiin osana Clever Simulation Entertainment -kehitystiimiltä tilatuttua raitiovaunun vika- ja huoltotietojen visualisointidemoa. Demo on osa lisätyn todellisuuden soveltamista teollisuuteen edistävää InduSTAR-hanketta, sen tilaajana toimii Transtech Oy. Clever Simulation Entertainment on Kajaanin ammattikorkeakoulun alaisuudessa toimiva uuden teknologian ja virtuaalitodellisuuden kehitystiimi.

Työssä perehdyttiin uusimpiin lisätyn todellisuuden ratkaisuihin sekä toteutettiin sovellus valitulle lisätyn todellisuuden ratkaisulle. Teknologisten ominaisuuksien pohjalta toteutusympäristöksi valittiin Magic Leap -laiteympäristö. Työn lopullisena tavoitteena on tuottaa esittelyversio Transtech Oy:lle raitiovaunun huoltotöitä helpottavasta lisätyn todellisuuden sovelluksesta. Sen tavoitteena on havainnollistaa tulevaisuuden teknologioita ja sovelluskohteita, jotka mahdollistavat mekaanikon tehokkaan työn ilman vaunukohtaista perehdytystä.

Lisätty todellisuus on ollut kehityksessä eri muodoissa jo yli kolmekymmentä vuotta. Viimeaikaiset teknologiset edistysaskeleet ovat mahdollistaneet sen, että lisätyn todellisuuden käyttöönotto jokapäiväisessä toiminnassa on mahdollista. Laitteet itsessään eivät ole vielä suunnittelultaan käytännöllisiä arkikäyttöön. Lisättyä todellisuutta käytetään jo eri tavoilla mobiilipeleissä, lisäksi teknologiaa on kokeiltu aseteollisuudessa tiedon tuontiin sotilaille. Kuitenkin vasta viime aikoina lisättyä todellisuutta on tuotu teollisuuteen vakavasti otettavana informaatio työkaluna. Lähitulevaisuudessa jalansijan saatuaan lisätty todellisuus tulee lisääntymään räjähdysmäisesti.

Opinnäytetyössä perehdytään lisättyyn todellisuuteen, sen teknologioihin ja sovellusratkaisuihin sekä tuotettiin ensimmäinen versio raitiovaunun vika- ja huoltotiedon visualisoinnin sovelluksen esittelyversiosta käyttäen Magic Leap One -laitetta ja Unreal Engine 4 -pelimoottoria. Demoon tehtiin mahdollisuus seurata raitiovaunun tietoverkon vikalokia visualisoituna kolmeulotteisessa lisätyssä todellisuudessa. Tulevaisuudessa sitä kehitetään edelleen toimimaan useammassa raitiovaunun osakategoriassa. Datan visualisoinnin hallintaan kehitetään myös graafinen käyttöliittymä.

KAMK • University of Applied Sciences

Abstract

Author(s): Kesäläinen Lassi

Title of the Publication: Augmented Reality Development with Magic Leap Hardware Environment

Degree Title: Bachelor of Engineering, Information and Communication Technology

Keywords: Augmented reality, Magic Leap, Unreal Engine 4

This thesis has been made as part of a tram car's fault and maintenance application demo ordered from the development team Clever Simulation Entertainment. The demo is part of InduSTAR program to increase the adaptation of augmented reality in the industry and it has been ordered by Transtech Oy. Clever Simulation Entertainment is a development team that specializes in new technology and virtual reality. It is a subordinate of Kajaani University of Applied Sciences.

The main goal of the thesis was to explore new solutions of augmented reality and to produce a demo application as per the selected augmented reality solution. Magic Leap One was chosen as the hardware environment because of its technological capabilities. The demo's goal is to produce a showcase for Transtech Oy of an augmented reality application that eases the maintenance of tram cars. The showcase demonstrates future technologies and solutions that enable the mechanic's work without tram-specific familiarization.

Augmented reality has been developed in different forms for over thirty years. The latest technological advances have made it possible to add its implementation into everyday life. Hardware used for augmented reality is yet to be designed to be suitable for conventional life. Augmented reality is already in use in different implementations of games and it has been tested even in the arms industry for bringing information for soldiers. Only lately has augmented reality seen serious adaptations in industry as an information tool. In near future, once augmented reality's foothold is established, it is going to grow exponentially.

The thesis includes an introduction to augmented reality, augmented reality technologies and solutions and the first iteration of the tram car's fault and maintenance application demo made with Magic Leap One hardware and Unreal Engine 4 game engine. The demo enables viewing the tram car's data network in visualized three-dimensional augmented reality. In the follow-up development, the demo will be complemented with more part categories and a graphical interface for managing the visualization of said categories.

Sisällys

Symboliluettelo

AR, Augmented reality	Lisätty todellisuus
Demo	Esittelyversio
GPS	Maailmanlaajuinen paikallistamisjärjestelmä
Graafinen	Kuvaannollisesti havainnollistettu
Immersiivinen	Mukaansa tempaava, kontekstiin sopiva, todentuntuinen elämys
Luokka	Ohjelmoinnissa määritelty objekti
Magic Leap	Lisätyn todellisuuden käyttöön suunnitellun laitejärjestelmän valmistaja
Magic Leap One	Ensimmäinen kaupallinen Magic Leap -laitejärjestelmä
Plugin	Lisäosa
QR koodi	Kaksiulotteinen kuvakoodi, joka voidaan muuttaa informaatioksi
Renderöidä	Piirtää näytölle tai kuvalle, kuvantaa, hahmontaa

1 Johdanto

Lisätyn todellisuuden teknologioita on jo vuosien ajan ilmaantunut jokapäiväiseen elämään eri muodoissa. Termin alaisuuteen kuuluu niin autojen tuulilasiin heijastettu mittaristo kuin urheiluohjelman lähetyksen yhteydessä tehty pelaajien ja taktiikoiden visualisointi.

Peliteknologiassa on olemassa useita esimerkkejä siitä, miten puhelimen kameran tuottamaa kuvaa ja laitteen sijaintia voidaan käyttää hyväksi pelin sisällössä. Nyt laiteteknologia on saavuttanut tason, joka mahdollistaa lisätyn todellisuuden tuomisen työympäristöihin. Laitteiden keventyessä lisätty todellisuus tulee suuntautumaan enemmän myös arkielämään. Lisätty todellisuus tulee olemaan seuraava iso muutos ihmisten elämässä. Teknologian kehittyessä ja jalansijan saatuaan lisätyn todellisuuden käyttö tulee kasvamaan räjähdysmäisesti.

Tässä opinnäytetyössä tuodaan esille kokemuksia ja hyväksi todettuja käytänteitä lisätyn todellisuuden kehittämisestä viimeisimmillä laitteilla, keskittyen erityisesti Magic Leap -laitejärjestelmään, joka valittiin työn toteutuksen alustaksi. Itse kohdetyö on raitiovaunun vika- ja huoltotiedon visualisoinnin demonstraatio. Se on toteutettu Clever Simulation Entertainmentillä [1]. Demo on osa InduSTAR -hanketta, jossa pyritään edistämään uuden teknologian ja lisätyn todellisuuden käyttöä teollisuudessa [2]. Työn tilaaja on kiskokaluston valmistamiseen erikoistunut Transtech Oy [3]. Tässä opinnäytetyössä kehitettiin ja esitetään ensimmäinen versio demosta, jolla vika- ja huoltotiedot voidaan tuoda vaunun lokista käyttäjälle. Lisätty todellisuus, jossa on vikaloki ja huoltotiedot visualisoituna, mahdollistaa työkohteen tietojen lukemisen ja huollon samanaikaisesti.

Opinnäytetyössä käydään läpi lisätty todellisuus käsitteenä, perehdytään sen historiaan ja sovelluskohteisiin sekä pohditaan lisätyn todellisuuden tulevaisuuden näkymiä. Lisäksi opinnäytetyössä perehdytään Magic Leap -laitejärjestelmään sekä käytetään raitiovaunudemoa esimerkkinä järjestelmän hyödyntämisestä lisätyn todellisuuden luomiseen ja siihen vuorovaikuttamiseen.

2 Lisätty todellisuus

Lisätyn todellisuuden kiteytetty idea on, että käyttäjä kokee saman fyysisen maailman, oli käytössä lisättyä todellisuutta tai ei. Lisätty todellisuus lisää digitaalista tietoa, jota voi käsitellä samaan tapaan kuin käyttäjä olisi vuorovaikutuksessa fyysiseen maailmaan. [4, s. 2.]

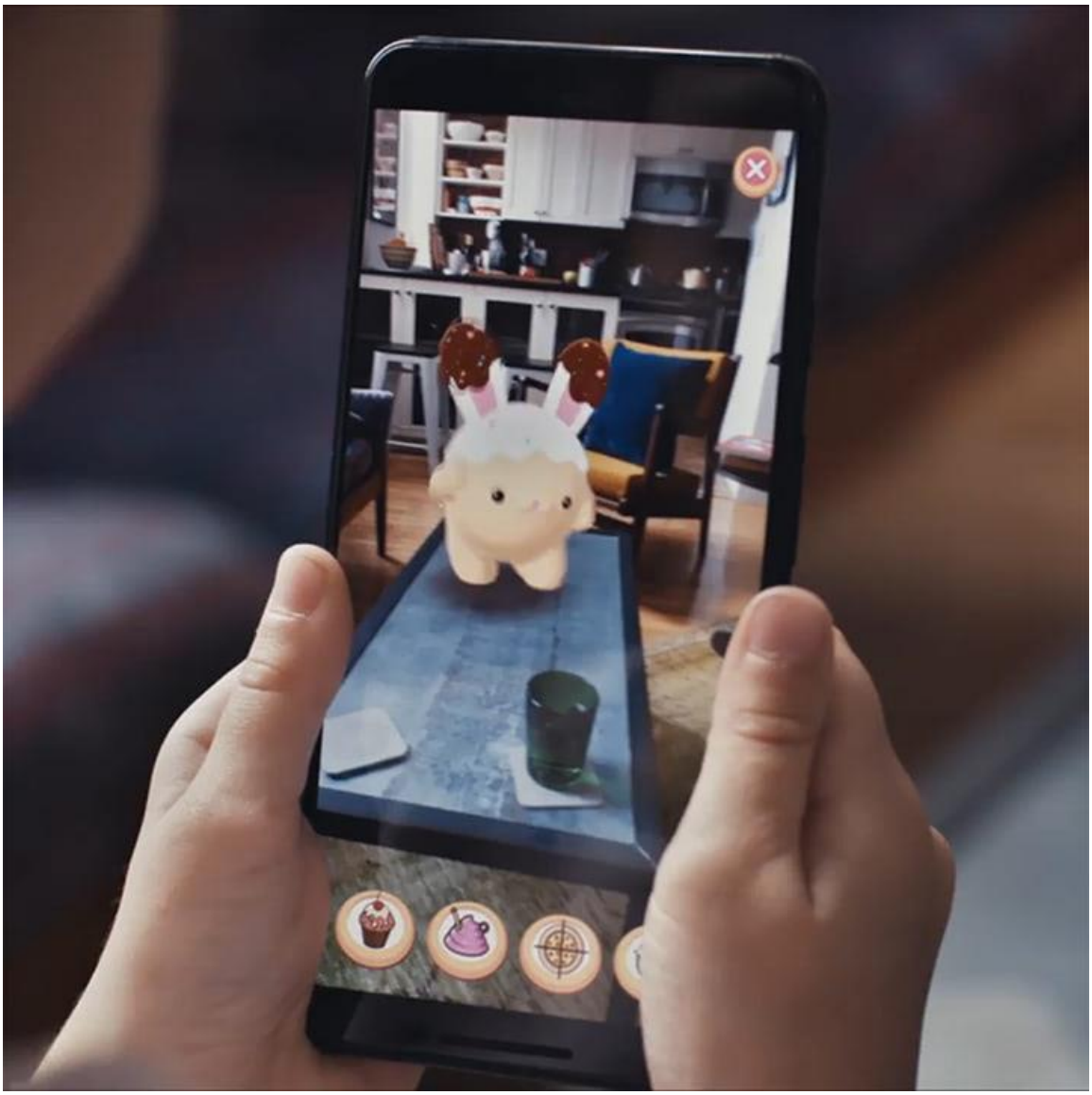
Tämä tarkoittaa, että lisätty todellisuus ei ole teknologia vaan pikemminkin tapa tuoda informaatiota fyysiseen maailmaan teknologian avulla. Käsite sisältää digitaalisen informaation esittämisen yksinkertaisesta tekstistä tai äänestä aina monimutkaisiin ja interaktiivisiin ohjelmiin. Yhteistä kaikille lisätyn todellisuuden ratkaisuille on se, että informaation pyritään tuomaan esille huomaamattomasti ja dynaamisesti. Usein esitettävä informaatio tuodaan käyttäjille joko kameran kuvaan lisättynä tai läpikuultavalla näyttölaitteistolla.

Monille tutuin ja käytetyin tapa tuoda digitaalista informaatiota esille on projisoitu lisätty todellisuus. Uutislähetyksien sääennusteiden ja urheilulähetyksien videokuvan päälle renderöidyt graafiset elementit ovat hyviä esimerkkejä jokaiselle saatavilla olevasta arkikäytöstä. Yksi yleisesti helposti tunnistettava projisoitu lisätty todellisuus on esimerkiksi auton tuulilasiin heijastettu näyttö (kuva 1). Heijastusnäyttö on hyvä esimerkki yksinkertaisesta ja kuluttajakäytössä olevasta teknologiasta, jolla voidaan tuoda dynaamista tietoa oikeaan maailmaan.

Kuva 1. Hudway OBD 2 heijastusnäyttö. [5]

Toinen koko ajan lisääntyvä televisio- ja elokuva-alan jälkeen yleisin lisätyn todellisuuden käyttömuoto on kädessä pidettävä lisätty todellisuus, kuten esimerkiksi mobiililaitteen avulla (kuva 2). Lisätyn todellisuuden ratkaisujen käyttöä mobiilisovelluksissa tukee erityisesti se, että matkapuhelimet ovat kaikkien saatavilla ja pääsääntöisesti jokainen ihminen omistaakin mobiililaitteen. Nykyaikaiset mobiililaitteet ovat suoritusteholtaan riittävän tehokkaita suorittamaan virtuaalisten ympäristöjen vaatimaa raskasta laskentaa.

Mobiililaitteet sisältävät lisäksi lähes kaikki immersiiviseen ja vuorovaikutettavaan lisättyyn todellisuuteen tarvittavat teknologiat, kuten kameran, magnetometrin, kiihtyvyysanturin ja gyroskoopin.



Kuva 2. Digitaalinen hahmo asetettuna fyysisestä maailmasta tunnistettuun tasoon. [6]

Kamera tuottaa hyödynnettävää kuvaa fyysisestä maailmasta, lisätyn todellisuuden käytössä siitä saadaan kuvatunnistuksella tietoa ympäristöstä. Näytöllisellä laitteella kameran tuottaman kuvan tai videon päälle myös renderöidään eli kuvan päälle piirretään informaatio, joka halutaan näyttää ennen kuin se tuodaan käyttäjälle. Gyroskooppi mahdollistaa sen, että näytöllä esitettävä informaatio pysyy samassa paikassa fyysiseen maailmaan nähden mobiililaitteen liikkeestä huolimatta. Magnetometri toimii puhelimen kompassina, kiihtyvyysanturin kanssa se pystyy kertomaan puhelimen asennon maapalloon verrattuna. Tämä mahdollistaa sen, että digitaalisen informaation sijainti on sama, vaikka ohjelma käynnistettäisiin uudelleen.

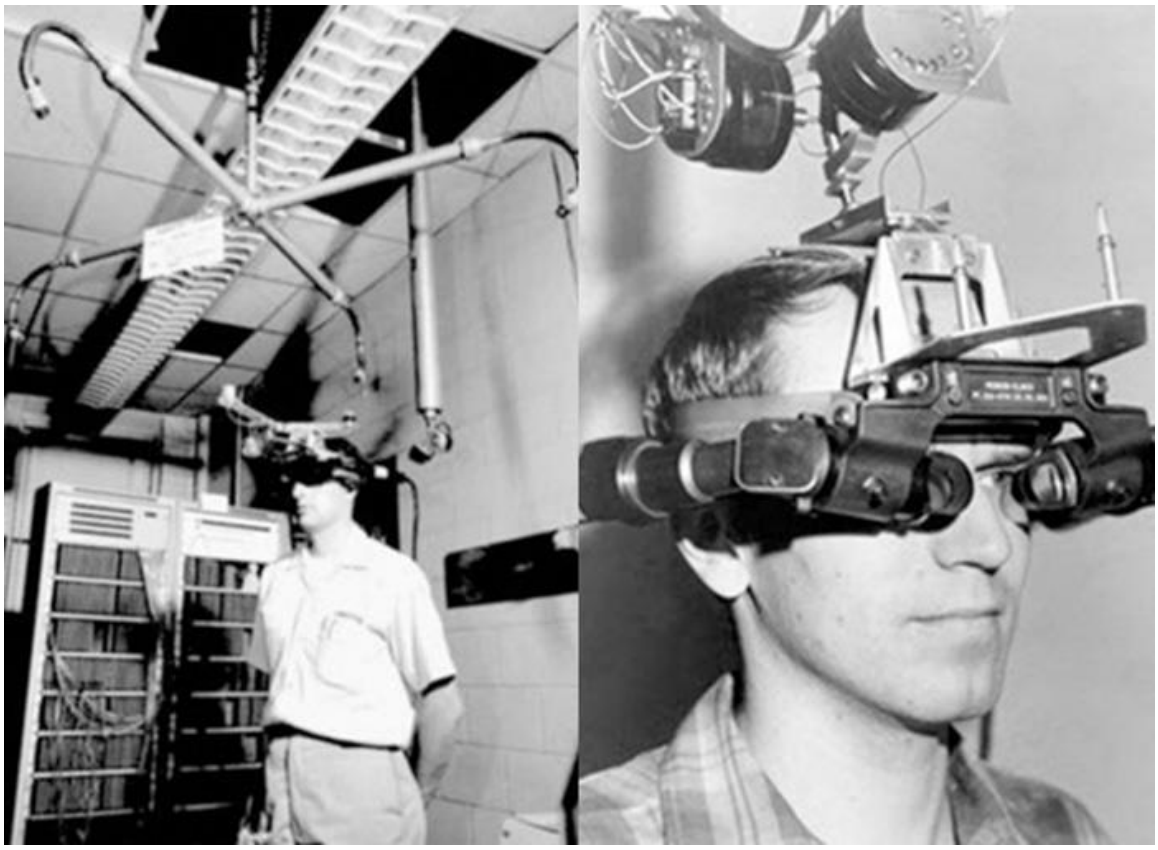
Seuraava iso askel lisätyn todellisuuden monipuollistumisessa ja sulauttamisessa fyysiseen maailmaan on käynnistynyt ja jo osittain käytössä teollisuudessa. Puettava tai tarkemmin määriteltynä päähän puettava lisätty todellisuus mahdollistaa sen, että virtuaalinen maailma voidaan tuoda käyttäjän näkökulmasta fyysisen maailman päälle ja laajalla tiedolla fyysisestä maailmasta se pystytään toteuttamaan hyvinkin saumattomasti (kuva 3). Tällä hetkellä laitteet ovat päähinemäisiä visiireitä, mutta teknologian kehittyessä ne muuttuvat lähemmäs ja lähemmäs silmä- tai aurinkolasien kaltaisuutta.

Kuva 3. Magic Leap demokuva, jossa on jälkirenderöity käyttäjän näkymä. [7]

2.1 Historia

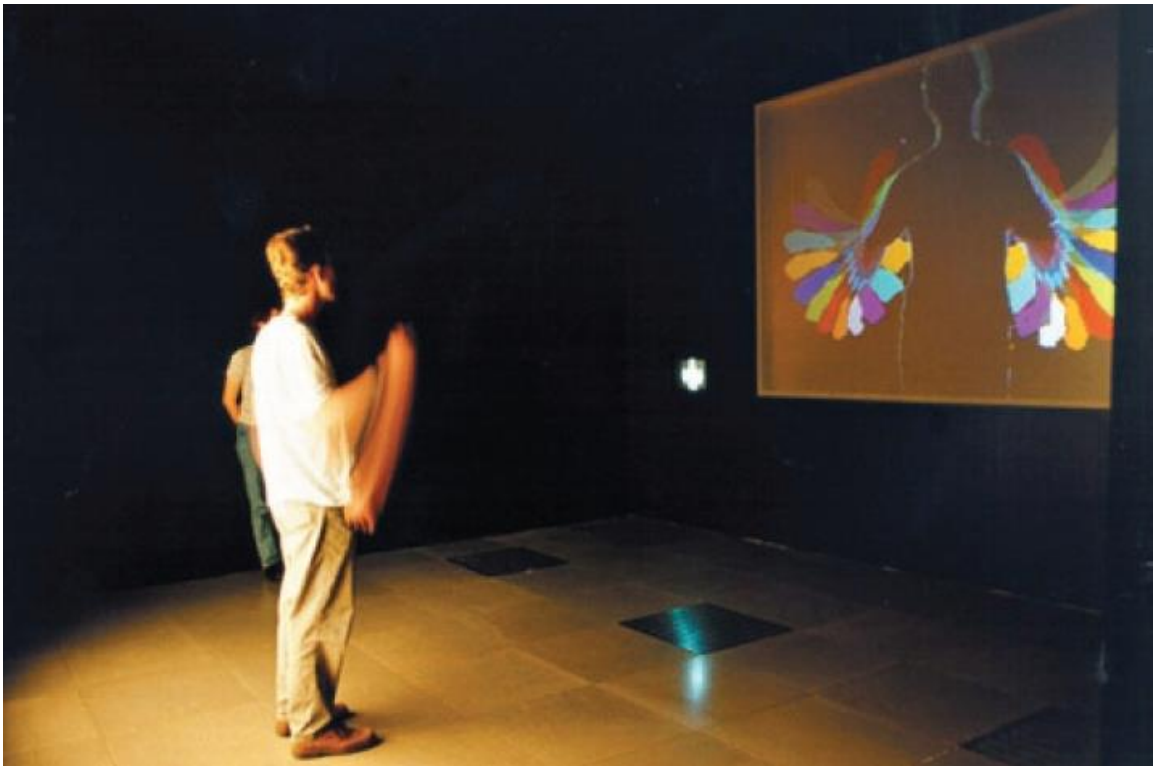
On ollut monia taitavia ja määrätietoisia ihmisiä, jotka ovat tehneet mahtavia asioita lisätyllä todellisuudella, paljon vähemmällä teknologialla kuin mihin ihmiset ovat tottuneet nykyään. [8, s. 7]

Vuonna 1968 Ivan Sutherland kehitti laitejärjestelmän The Sword of Damocles, jota pidetään sekä ensimmäisenä lisätyn todellisuuden että virtuaalitodellisuuden laitejärjestelmänä (kuva 4). Siinä oli syvyysnäköön kykenevä läpikatsottava näyttö ja käyttäjän liikettä seuraava järjestelmä. [8, s. 8]



Kuva 4. The Sword of Damocles. [9]

Vuonna 1975 Myron Kruegerin omistama videotalo kehitti aktiivisen varjokuvataideteoksen, jossa käyttäjä pystyi ensimmäistä kertaa vuorovaikuttamaan virtuaalisiin objekteihin (kuva 5). [8, s. 8]



Kuva 5. Videotalo, Myron Krueger. Jälkiväritetty. [10]

Vuonna 1996 Jun Rekimoto kehitti prototyypin merkatusta lisätystä todellisuudesta nimeltään NaviCam, joka käytti kaksiulotteista matriisimerkintää. Sen avulla voitiin päättää paikka, jossa digitaalista informaatiota esitetään. Se oli ensimmäisiä lisätyn todellisuuden teknologioita, joka käytti merkattua ympäristöä. [8, s. 10]

Vuonna 1999. Hirokazu Kato julkaisi ARToolKit kehitystyökalut avoimeksi lähdekoodiksi. Kehitystyökalua päivitetään edelleen ja suurin osa Flash-pohjaisista lisätyn todellisuuden sovelluksista onkin kehitetty ARToolKittiä hyväksi käyttäen. [8, s. 10]

Vuonna 2016 Microsoft julkaisi HoloLens-kehittäjäversion. Se on yksi ensimmäisistä uuden sukupolven lisätyn todellisuuden laseista [11]. Samana vuonna julkaistiin Pokemon Go, lisätyn todellisuuden mobiilipeli, joka on menestynyt ympäri maailmaa [12].

2.2 Sovelluskohteet

Lisättyä todellisuutta voidaan käyttää hyvin erilaisiin tarkoituksiin, vaikka toteutukset voivatkin olla lähtökohtaisesti samankaltaisia. Lisätyn todellisuuden ympäristön käyttötarkoitus määrittelee kuitenkin erilaiset ehdot toteutukselle. Sovelluskohteet voidaan kuitenkin helposti luokitella työympäristökäytöksi, hyötykäytöksi ja viihdekäytöksi.

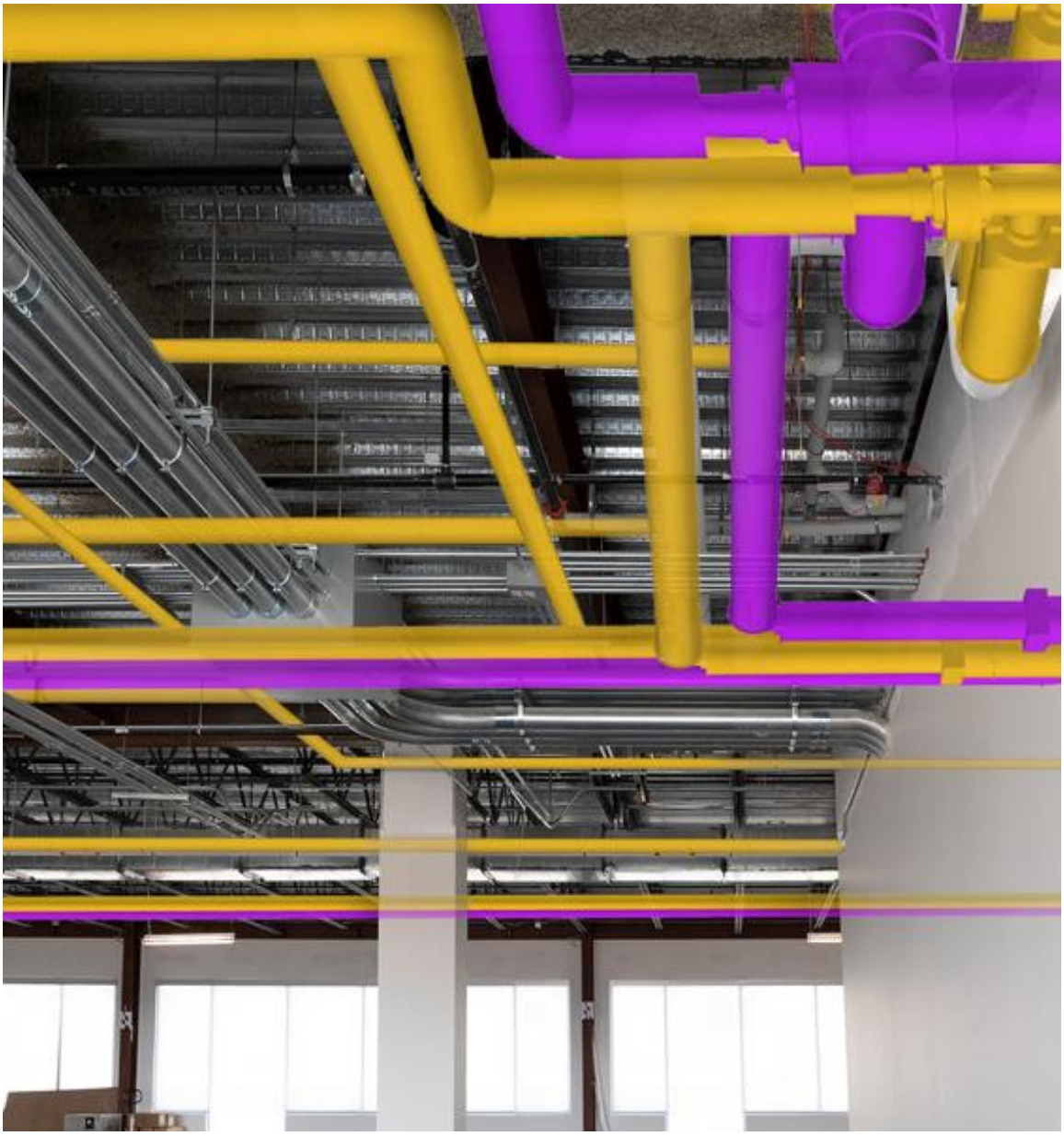
Työympäristö voi olla hyvinkin vaativa informaation tarkkuuden ja helppolukuisuuden kannalta. Samalla usein edellytetään, ettei ohjelma häiritse työntekijän ympäristön havainnointia. Hyötykäytössä on selkeä tavoite, arjen tehtävien helpottaminen ja informaation nopea saatavuus. Viihdekäytössä suurimmassa osassa on käyttäjän viihdyttäminen ja aktivoiminen.

2.2.1 Työympäristö

Työympäristöissä lisätty todellisuus mahdollistaa kriittisen informaation tuonnin ja visualisoinnin käyttäjälle ilman erillisiä ohjeita ja selvityksiä. Tämä mahdollistaa sen, että uusikin työntekijä voi olla ajan tasalla työvaiheista ja nähdä tehtävän suorituspaikka ja lopputulos fyysisessä maailmassa jo ennen työn toteutusta (kuva 6).

Moni rakennus- ja teollisuusalan yhtiö onkin jo ottanut lisätyn todellisuuden osittain käyttöön, ja eri alojen toimijat testaavat lisätyn todellisuuden ratkaisuja ja mahdollisuuksia niin työtehon kuin myös työturvallisuuden parantamiseen liittyen.

Esimerkiksi Bechtel suuri yhdysvaltalainen rakennus- ja teollisuusalan yritys, on ottanut lisätyn todellisuuden työntekijöiden koulutuskäyttöön vaarallisten kohteiden kanssa työskentelyssä. Lisätty todellisuus mahdollistaa mm. vaaratilanteita aiheuttavien kohteiden tunnistamisen ja videopuhelun kouluttajiin. [13.]



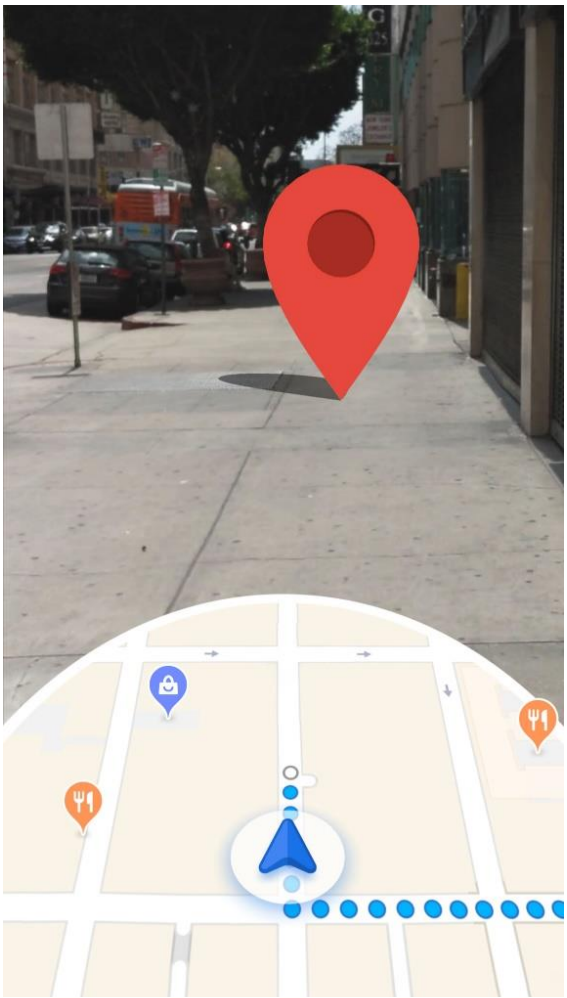
Kuva 6. Putkiston pohjapiirustus visualisoituna rakennustyömaalla. [14]

2.2.2 Hyötykäyttö

Arjen hyötykäytössä lisätty todellisuus on vielä varhaisiässä, mutta yleistyy tasaiseen tahtiin. Yritykset ja palveluntuottajat tekevät lisätyn todellisuuden ratkaisuja monesti helpottamaan jo olemassa olevia palveluja.

Hyvänä esimerkkinä toimii IKEA Place, joka mahdollistaa huonekalu ostoksien katselun ostokohteessa. [15.] Se on yksinkertainen, mutta toimiva lisätyn todellisuuden ratkaisu. Käyttäjä voi katsella huonekalujen 3D-malleja ja asettaa ne manuaalisesti kolmeulotteisella akselilla mahdollistaen huonekalun katselun puhelimella sen tulevassa paikassa.

Palveluntuottajista Googlella on aivan oma asema lisätyn todellisuuden kehityksessä. Sillä ei ole ainoastaan iso määrä jo olemassa olevia palveluja, joita voi parantaa lisätyllä todellisuudella (kuva 7). Googlen ARCore on myös yksi suosituimmista lisätyn todellisuuden ohjelmointirajapinnoista. Lisäksi, vaikka Googlen tuotteet on usein suunnattu heidän omistamalleen Android käyttöjärjestelmälle, ARCore rajapinta mahdollistaa IOS -käyttöjärjestelmän kanssa yhteensopivan lisätyn todellisuuden kehityksen. [6.]



Kuva 7. Google Maps lisätyssä todellisuudessa. [6]

2.2.3 Viihde

Viihdekäytössä lisätty todellisuus on jo käytössä useissa mobiilipeleissä, joissa pelin tapahtumat on sidottu käyttäjän sijaintiin ja pelimaailma piirretään fyysisen maailman päälle. Tunnetuin esimerkki on Pokemon Go ja kehitteillä oleva Minecraft Earth. [16.] Microsoft toteutti Minecraft Earth pelistä puettaville lisätyn todellisuuden laitteille suunnatun konseptidemon, joka myöhemmin suunnattiin mobiililaitteille niiden paremman saatavuuden takia.

Lisätyn todellisuuden pelit ovat hyvä suunnannäyttäjä siitä, mitä kaikkea lisätyn todellisuuden avulla voidaan toteuttamaan nykyteknologian avulla. Tulevaisuus tuo mukanaan paitsi uutta teknologiaa, myös parempia ratkaisuja ohjelmistokehitykseen. Teknologian ja kehitysalustojen kehittyessä kaikissa lisätyn todellisuuden sovelluskohteissa tullaan näkemään suuria loikkia kehityksen osalta.

2.3 Teknologia

Lisätyn todellisuuden käsitteeksi riittää, että tuodaan interaktiivista fyysiseen maailmaan sidonnaista digitaalista informaatiota käyttäjälle. Odotukset lisätystä todellisuudesta ovat kuitenkin usein suuntautuneet viimeisimmän teknologian saralle immersiiviseen ja sulavaan kokemukseen, jossa virtuaalinen ja fyysinen maailma ovat käyttäjän näkökulmasta sama kokemus (kuva 8).

Kuva 8. Hyper Reality-lyhytelokuvassa esitellään dystopiamaista lisättyä todellisuutta. [17]

Fyysiseen maailmaan saumattomasti liitetty lisätty todellisuus vaatii paljon teknologiaa. Sen mukanaan tuomat haasteet on ratkaistava laitejärjestelmän ominaisuudet ja rajoitteet huomioon ottaen.

2.3.1 Keinotekoisen maailman tuonti oikean rinnalle

Digitaalinen informaatio voidaan tuoda käyttäjälle missä tahansa muodossa. Suosituin tapa riippumatta siitä, onko informaatio sidottu kaksi- vai kolmiulotteiseen avaruuteen, on tuoda se omana instanssinaan, jota voi verrata peleistä tuttuihin maailmoihin. Virtuaalinen maailma voidaan siten sovittaa näyttölaitteeseen haluttuun kohtaan ja halutussa perspektiivissä.

Informaation sijainti suhteessa fyysiseen maailmaan voidaan toteuttaa manuaalisesti käyttäjän asettamana tai automaattisesti, kuten sijainnin määrittäminen kameran kuvaa tunnistamalla tai paikallistamisjärjestelmän avulla.

2.3.2 Näyttöratkaisut

Lisätyn todellisuuden näyttöratkaisut vaihtelevat riippuen lopullisesta käyttölaitteesta. Mobiililaitteessa voidaan käyttää näyttöä kuvan esittämiseen käyttäjälle. Näytettävä kuva luodaan renderöimällä kameran tuottaman kuvan päälle haluttu informaatio ennen kuin se esitetään näytöllä käyttäjälle.

Puetussa lisätyssä todellisuudessa suosittu vaihtoehto on joko käyttää läpinäkyvää näyttöä tai katselukohtaan heijastettavaa kuvaa. Tämä mahdollistaa sen, että käyttäjä näkee fyysisen maailman omilla silmillään samalla, kun virtuaalinen maailma tuodaan luontevasti sen päälle täydentämään kokemusta tarkoituksen mukaan lisätyllä informaatiolla.

2.3.3 Kameroiden käyttömahdollisuudet

Perinteisen näytöllisen laitteen tapauksessa kameran kuvaa käytetään muokatun maailman piirtämiseen, mutta myös läpinäkyvällä näytöllä on tarve kameralle.

Kameran kuvasta voidaan tunnistaa pienelläkin suorituskyvyllä merkkejä, kuten QR-koodeja. QR-koodi mahdollistaa yksilöidyn informaation esittämisen käyttäjille ympäristöön sijoitettavilla ku-

vakoodeilla. Verrattaessa normaaliin sijainnin tunnistukseen kuvasta QR-koodi sisältää informaatiota, joka puolestaan mahdollistaa tunnistuskohteeseen sidotun tiedon yksilöimisen ohjelman ulkopuolisesti jo fyysisessä maailmassa. QR-koodit ovat varsin yleisesti yrityksien ja kaupunkien käytössä. QR-koodin käyttö kuitenkin edellyttää QR-Koodit sisällään pitävien markkereiden sijoittamista oikeaan maailmaan.

Raskaampaan laskentaan kykenevällä laitteella voidaan niin ikään tunnistaa fyysisen maailman tasoja, paikkoja, ihmisiä ja kädenliikkeitä. Lähes mitä tahansa, josta voidaan laskea toistuva geometria. Monimutkainen kuvatunnistus itsessään vaatii ison määrän dataa ennen kuin sitä voidaan hyödyntää luetettavasti, mutta tekoälyn ja tekniikoiden kehittyessä se on koko ajan helpompaa mahdollistaen parempia ja älykkäämpiä ohjelmia lisätyssä todellisuudessa.

Kahdella kameralla saadaan tarkemmat tulokset, mutta ennen kaikkea se mahdollistaa syvyysnäön. Tämä mahdollistaa sen, että jos tunnistetaan jotain kuvasta, niin pystytään myös laskemaan sen tarkka etäisyys.

2.3.4 Ohjainvaihtoehdot

Virtuaaliseen maailmaan voidaan vaikuttaa monella eri tavalla.

Kosketusnäytöllä voidaan käsitellä informaatiota samaan tapaan kuin missä tahansa mobiiliapplikaatiossa. Gyroskoopilla varustetussa laitteessa voidaan käyttää myös laitteen asentoa manipuloimaan virtuaalisen maailman asentoa.

Lisäksi voidaan käyttää erillistä ohjainta, joka toimii osoituskynän tapaan ja mahdollistaa ylimääräisien näppäimien käytön. Tällaisella ohjaimella on lähes samat vaatimukset kuin itse päälaitteella, koska sen sijainnin ja asennon pitää olla koko ajan tiedossa.

Kameraa hyödyntämällä voidaan myös käyttää käsiä ohjainten sijaan ja kahdella kameralla varustetulla laitteella syvyysnäön ansiosta voidaan saada hyvinkin luonnollinen käyttökokemus, jossa pystytään manipuloimaan virtuaalista maailmaa luonnollisilla käden liikkeillä.

2.3.5 Sensorit

Lisätyn todellisuuden saumaton sulauttaminen fyysiseen maailmaan vaatii paljon sensoriteknologiaa, jotta virtuaalinen maailma pysyy oikeassa paikassa relatiivisesti fyysiseen maailmaan.

Gyroskooppi on sensoreista tärkein, koska sen avulla määritetään laitteen asento. Tämän tiedon perusteella virtuaalista maailmaa voidaan liikuttaa siten, että se pysyy paikallaan relatiivisesti fyysiseen maailmaan nähden.

Magnetometri toimii elektronisten laitteiden kompassina. Sen avulla voidaan mitata voimakkaimman magneettikentän sijainti, joka mahdollistaa tiedon pohjoisen sijainnista.

Kiihtyvyysanturi mahdollistaa laitteen liikkeen käytön osana sovelluksia, ja sitä käytetään erityisesti viihdesovelluksissa, mutta sen tärkein ominaisuus on mahdollisuus painovoiman suunnan laskentaan.

Magnetometri yhdessä kiihtyvyysanturin kanssa mahdollistaa puhelimen tarkan asennon tietämisen. Asennon tiedon avulla pystytään pitämään virtuaalinen maailma samassa paikassa, vaikka laite välillä sammutetaan.

GPS (Maailmanlaajuinen paikallistamisjärjestelmä) mahdollistaa laitteen sijainnin löytämisen. GPS on vaihtoehtoinen tapa kuvatunnistukselle sijoittaa virtuaalimaailma sijainnin mukaan. Jo pelkästään laitteen sijainti mahdollistaa paikkariippuvaisen informaation näyttämisen käyttäjälle. GPS-tiedon yhdistäminen magnetometrin ja kiihtyvyysanturin kanssa mahdollistaa virtuaalisen maailman kohdistamisen kolmiulotteisessa avaruudessa kuuden vapauden asteen tarkkuudella. Kun tiedossa on laitteen asennon ja sijainnin lisäksi kohteen sijainti ja korkeus, voidaan laskea ja sijoittaa virtuaalinen maailma päällekkäin fyysisen maailman kanssa. Tämä vaatii laitteelta hyvää laskentakykyä. Lopputuloksen tarkkuus on täysin laitteen sisältämien sensoreiden tarkkuuden varassa.

Hajavalomittain mahdollistaa kuvatunnistuksen laskennan säätämisen valon mukaan luotettavimpien tuloksien saamiseksi.

Mikrofoni puolestaan mahdollistaa äänikomennot ja ääniriippuvaisten tapahtumien lisäämiseen virtuaaliseen maailmaan.

2.3.6 Lisätyn todellisuuden eri muodot

Lisätty todellisuus jaetaan yleensä laitteesta riippumatta merkattuun, merkkaamattomaan ja sijaintipohjaiseen. Siitä huolimatta sama sovellus voi hyväksikäyttää kaikkia muotoja samanaikaisesti.

Merkattu lisätty todellisuus tarkoittaa sitä, että digitaalinen informaatio tuodaan käyttäjälle paikkasidonnaisesti fyysiseen maailmaan sijoitetuilla asioilla. Esimerkiksi informaatiota tuodaan käyttäjälle, kun laite näkee QR koodin. Toinen tapa on esimerkiksi sijoittaa virtuaalinen maailma laitejärjestelmään yhdistetyillä ulkopuolisilla kameroilla tai seurainlaitteilla, joilla kartoitetaan laitteen sijainti. Merkattua lisättyä todellisuutta käytetään yleensä silloin, kun fyysinen maailma on vakio. Esimerkiksi museo tai esittelytila on ympäristönä omiaan merkattuun lisättyyn todellisuuteen.

Merkkaamaton lisätty todellisuus tarkoittaa sitä, ettei laitteen tarvitse tietää sijaintia. Tämä yleensä tarkoittaa sitä, että virtuaalinen maailma sijoitetaan fyysisen maailman päälle reaaliajassa kameran kuvan avulla tunnistetuille tasoille. Tämä myös yleensä tarkoittaa sitä, että jos sovelluksessa on paikka- tai kohdesidonnaista dataa, käyttäjän on sijoitettava se itse tai se voidaan esittää vasta, kun kameralla on näköyhteys kohteeseen ja se on onnistuneesti kuvatunnistettu.

Sijaintipohjainen lisätty todellisuus viittaa paikannusjärjestelmää hyväksikäyttävään sovellukseen. Tämä vaatii laitteelta enemmän sensoreita, mutta se mahdollistaa tiedon ja maailman sijoittamisen kontekstilla ilman ulkopuolisia lisälaitteita tai merkkejä. Sijainnin saaminen, kuitenkin edellyttää satelliittiyhteyden ja on tarkkuudeltaan riippuvainen laitteesta ja magneettikentistä, jotka voivat häiritä niin GPS-signaalia kuin magnetometriäkin.

3 Magic Leap

Magic Leap -laitejärjestelmä valittiin käyttöön, koska se oli työn aloituksen ajankohdalla tehokkain ja monipuolisin lisätyn todellisuuden laitejärjestelmä.

Magic Leap One on avaruudellinen tietokone, joka antaa nähdä ja vuorovaikuttaa digitaalista substanssia ympäröivässä maailmassa [18].

3.1 Laitteisto

Magic Leap -laitejärjestelmä sisältää lasit, vyöllä kannettavan tietokoneen ja ohjaimen (kuva 9). Laseissa on stereoskooppisen näön mahdollistavat läpinäkyvät näytöt, käyttäjän silmien seuranta, neljä mikrofonia ja tilaääneen pystyvät kaiuttimet. Lisäksi siinä on normaalin kameran lisäksi neljä sensorikameraa, jotka mahdollistavat tarkan syvyysnäön ja fyysisen maailman digitaalisen uudelleenluonnin. Laitteen gyroskoopit, kiihtyvyysanturit ja magnetometri mahdollistavat tarkan kuuden asteen vapauden, joka tarkoittaa sijainnin seurantaa kolmiulotteisessa vektoriavaruudessa laitteen asento ja kulma mukaan lukien. Lasit sisältävät myös ohjainlaitteen seurannan. [18.]

Kuva 9. Magic Leap One -laitejärjestelmä. [19]

Ohjain pitää sisällään myös gyroskoopit, kiihtyvyysanturin ja magnetometrin, joten sen sijainti pysyy tarkkana virtuaalisessa maailmassa. Lisäksi se sisältää haptisen palautteen värinän muodossa. Ohjaimessa on kaksi nappia, yksi liipaisin ja kosketuslevy. Kosketuslevy toimii tietokoneen hiiren tapaan osoittimen ohjaajana, mutta sitä voidaan myös käyttää ohjaussauvana. [18.]

Kevytpaketti eli Magic Leap Onen suorituskykyinen tietokone hoitaa kaiken laskennan. Se pitää sisällään LuminOS-käyttöjärjestelmän, joka tukee Bluetooth 4.2, WiFi 802.11a/b/g/n/ac, USB-C yhteyksiä ja sisältää myös kaiuttimen. [18.]

3.2 Käyttömahdollisuudet

Magic Leap One -laitteiston kattava komponentti kokoonpano mahdollistaa virtuaalitodellisuuden kaltaisen kokemuksen tuonnin fyysiseen maailmaan. Eri teknologiat sijainnin seurantaan mahdollistavat pysyvän kokemuksen luomisen, tarkoittaen että laitteen uudelleen käynnistämisen jälkeen maailma sijaitsee samassa paikassa kuin se oli ennen sovelluksen sulkemista. Lisäksi vuorovaikutuksen virtuaalimaailman kanssa pystyy toteuttamaan omilla käsillä ilman tarvetta erilliselle ohjaimelle. Ohjainta käyttäessä sen sijainti ja suunta on koko ajan tiedossa virtuaalimaailmassa ja sitä on luonteva käyttää osoittimena tai peliohjaimena. Ohjainyhdistelmän käyttö on kokemukseltaan samalla tasolla kuin vastaavissa virtuaalitodellisuuslaitteistoissa.

Laitejärjestelmästä puuttuu GPS-paikannin, mikä pois sulkee täydellisen sijaintiin perustuvan lisätyn todellisuuden. Osittainen sijainti on kuitenkin mahdollista saada WiFi-verkon kautta IP-osoitteen sijaintina. Kamerateknologian avulla laitejärjestelmä pystyy nopeaan ja tarkkaan kuvatunnistukseen, joka on hyödynnettävissä paikkaan sidotun lisätyn todellisuuden tekoon.

3.3 Kehitysympäristöt

Ohjelmakehitys Magic Leap -laitejärjestelmälle tapahtuu tällä hetkellä joko Windows tai Mac käyttöjärjestelmillä. Kirjoitushetkellä Linux-järjestelmille ei ole omaa kehitysympäristöä.

Lumin SDK eli Lumin ohjelmakehitystyökalut pitävät sisällään tarvittavat ajurit ja työkalut yhteydenpitoon Magic Leap One -laitteistoon sekä sovelluksien tekemiseen. C-API kirjasto eli C ohjelmointikielipohjaisen sovellusohjelmointi rajapinta sisältyy sekin Lumin SDK pakettiin ja sillä mahdollistetaan keskustelu sovelluksen ja Magic Leap -laitejärjestelmän kanssa.

Ohjelmankehitysympäristönä voi käyttää mitä tahansa kolmannen osapuolen ohjelmankehitysympäristöä. Magic Leap tarjoaa myös oman Lumin Runtime editorin C-ohjelmoijia varten, MagicScript-työkalut Java-kehittäjiä varten sekä Lumin Web Platfrom -työkalut selainsovelluksien tekoon. Magic Leap:n työkaluihin kuuluu lisäksi rajapinnat Unreal- ja Unity-pelimoottoreille.

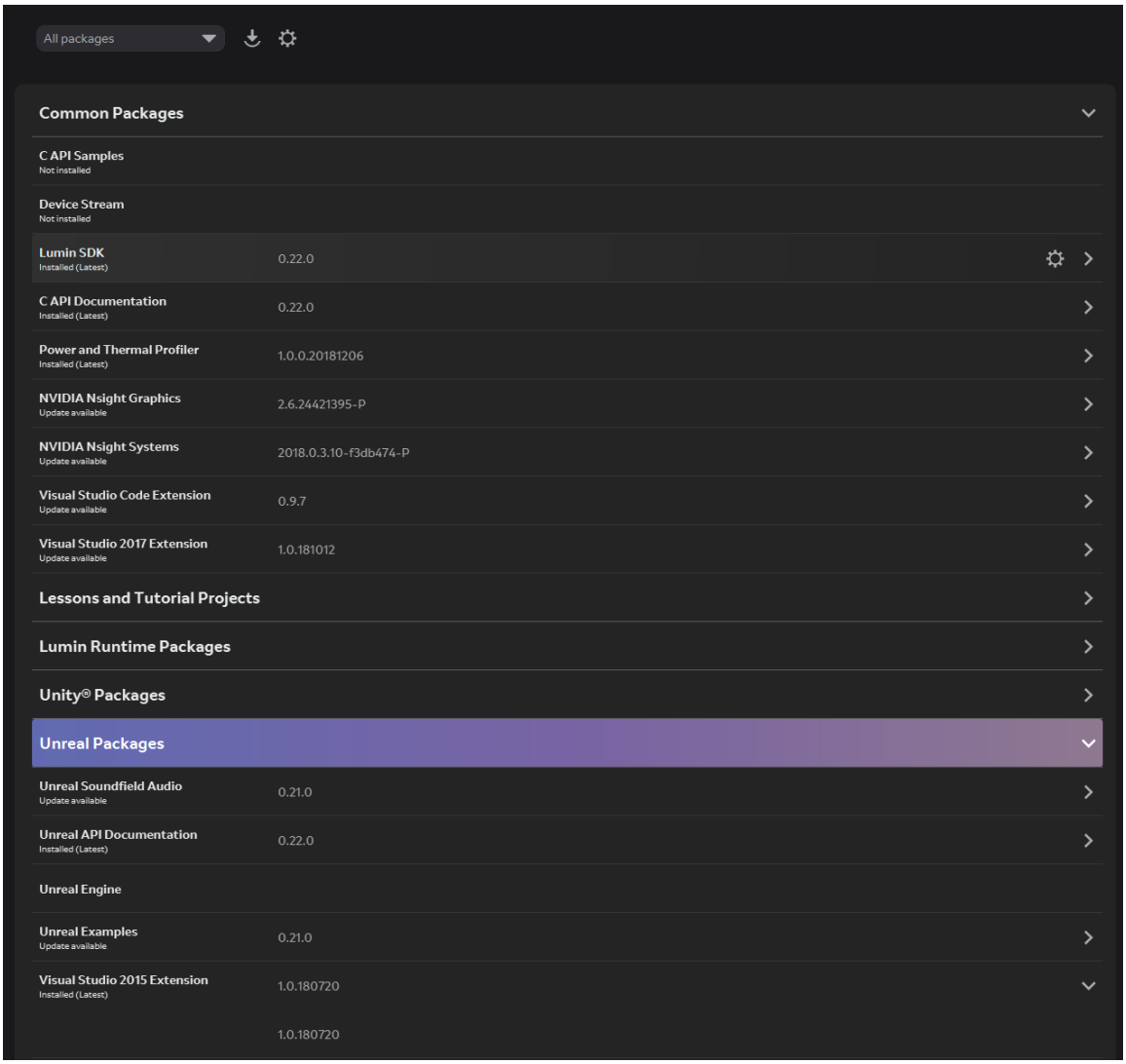
4 Magic Leap & Unreal

Opinnäytetyöhön valittiin kehitysympäristöksi Unreal Engine 4 -rajapinta, koska kehitysrajapinta oli entuudestaan tuttu, ja se tarjoaa kehitystyön käyttöön pelimoottorin käyttäjäystävällisen käyttöliittymän ja monipuolisen valikoiman kehitystyökaluja, mutta myös mahdollisuuden puhtaaseen C ja C++ pohjaiseen ohjelmointiin.

Unreal-pelimoottorille löytyy Magic Leapin oma versio GitHub-versionhallinta pilvipalvelusta, jossa on viimeisimmät päivitykset sekä binaariversio Epic Launcher hallinta sovelluksesta yksinkertaisempaan ympäristön pystytykseen. Molemmat kehitystyökalut sisältävät pelimoottorirajapinnan plugin-lisäosina. Tähän opinnäytetyöhön valittiin versionhallinnasta päivitetty version.

Ensimmäinen vaihe Magic Leap -kehitysympäristön käyttöönotossa alustasta riippumatta on ladata Magic Leapin sivuilta Package Manager latausten hallintasovellus, jonka kautta voidaan asentaa kehittäjätyökalut (kuva 10).

Package Managerista ladataan Lumin SDK ja Unreal Soundfield Audio. Package Managerissa löytyy myös linkki Unreal Enginen GitHub -säiliön Magic Leap kehityshaaraan. Sen käyttöönotto vaatii GitHub käyttäjän linkityksen Epic Gamesin Unreal GitHub -pilvipalveluun.

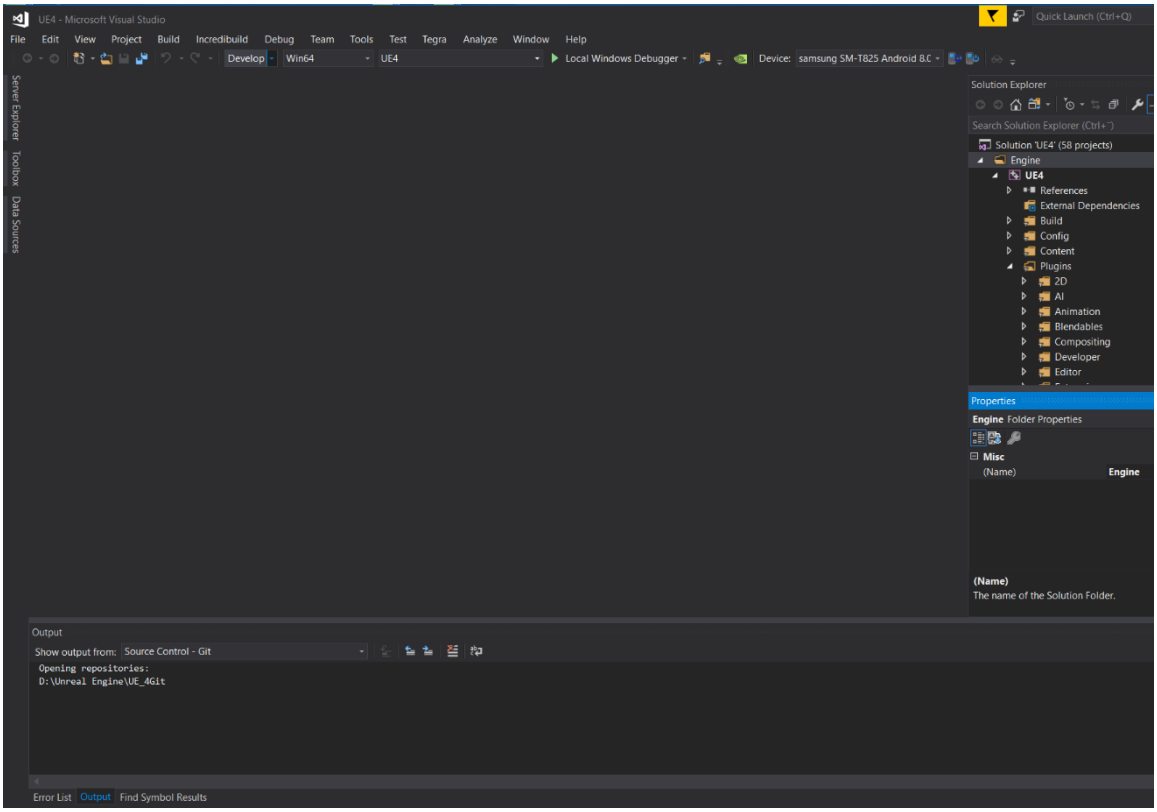


Kuva 10. Magic Leap Package Manager lataustenhallintatyökalu.

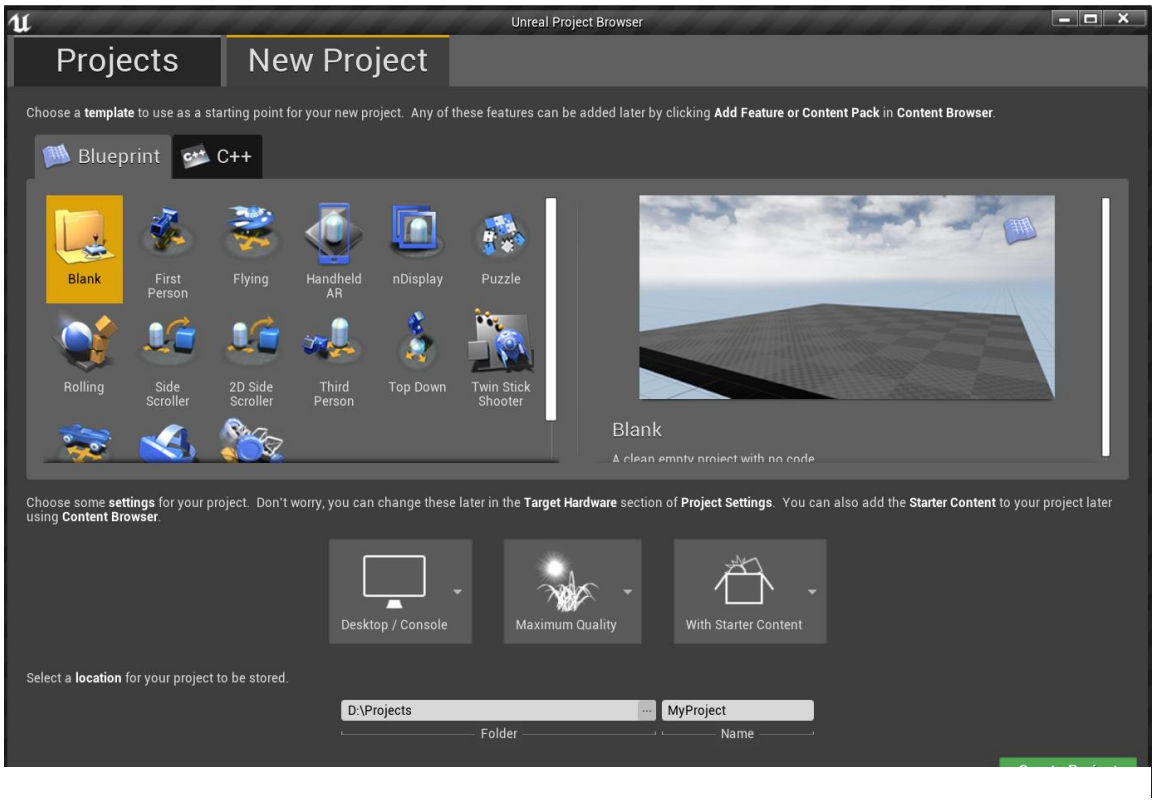
GitHub-versiohallinnan säiliöstä ladataan Unreal Engine Release-versio Magic Leap haarasta. Lumin SDK:n ja Soundfield Audion sekä Unreal Engine 4 versioiden on oltava yhteen sopivia. Ladattaessa viimeisimmät versiot tämä yleisesti ottaen pitää paikkansa.

Unreal Engine kootaan ohjelmakehitysympäristöllä (esim. Visual Studio) asetuksilla Development Editor, Käyttöympäristö ja UE4 (kuva 11). Koonnin jälkeen Unreal-pelimoottori voidaan ajaa. Pelimoottorin käynnistyessä aukeaa normaali Unreal-projektien hallintaselain, jonka avulla voidaan luoda uusi projekti. Luotu projekti sisältää automaattisesti Magic Leap -kehitykseen tarvittavat lisäosat (kuva 12).

Tämän jälkeen Unreal Soundfield Audio-lisäosa kopioidaan Unrealin Plugin-projektikansioon.



Kuva 12. Unreal Engine, Visual Studio 2015 ohjelmakehitysympäristössä.

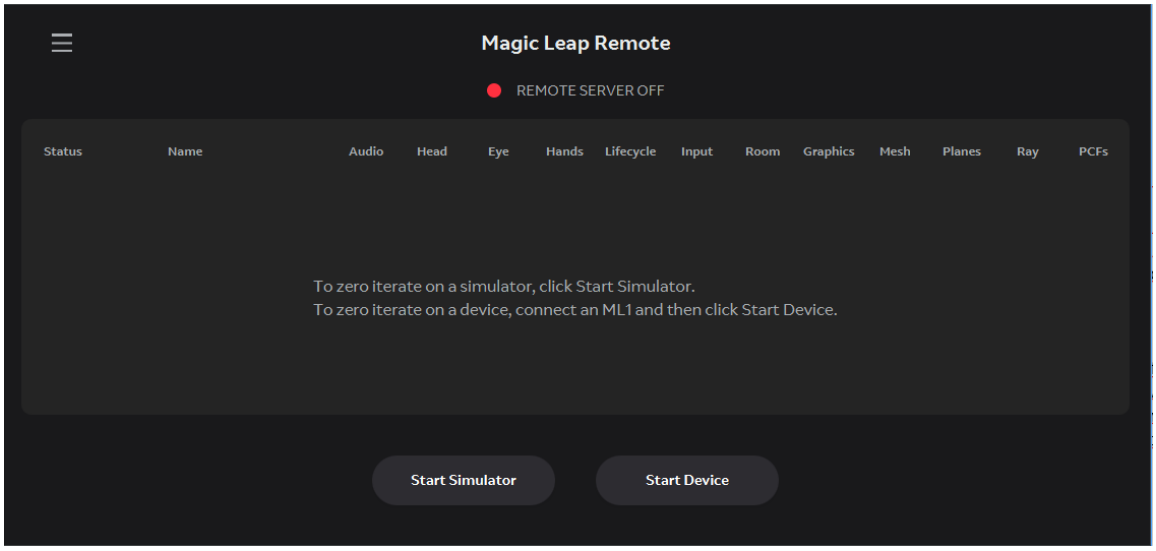


Kuva 11. Unreal Engine 4:n projekti selain.

C++ ohjelmoidessa on syytä muistaa lisätä projektin build.cs tiedostoon käytettävät moduulit, mikäli koodissa käytetään Magic Leap sovellus ohjelmointirajapintaa. Näistä yleisimmät ovat MagicLeap ja MLSDK, joista ensimmäinen sisältää pelimoottori rajapinnan ja toinen sovellusten ohjelmointi rajapinnan.

Viimeinen huomioitava asia Magic Leap kehityksessä on projektiasetuksien kuntoon laittaminen. Tärkeimmät ovat Magic Leapin verkkosivuilta ladattavan kehityssertifikaatin sekä Lumin SDK:n mukana tulevan MLSDK:n yhdistäminen projektin Magic Leap-kohtaisiin alusta-asetuksiin.

Lisäksi lisäosakategoriassa on Magic Leapillä oma kohta, jossa voi laittaa MLRemoten käyttöä varten Zero Iteraation päälle. MLRemotella voidaan nopeuttaa ja helpottaa kehittämistä mahdollistamalla sovelluksien testaaminen reaaliajassa ilman tarvetta kääntää ja asentaa sovellusta Magic Leap One -laitteen kiintolevylle (kuva 13). Tässä lähestymistavassa osa Magic Leap One-laitteen ominaisuuksista ei ole kuitenkaan käytössä. Esimerkiksi kameradatan käyttö ei ole mahdollista MLRemoten kautta.



Kuva 13. MLRemote mahdollistaa emulaattoriympäristön sekä reaaliaikaisen sovelluksen ajon Magic Leap -laitejärjestelmiin.

5 Tekninen osuus

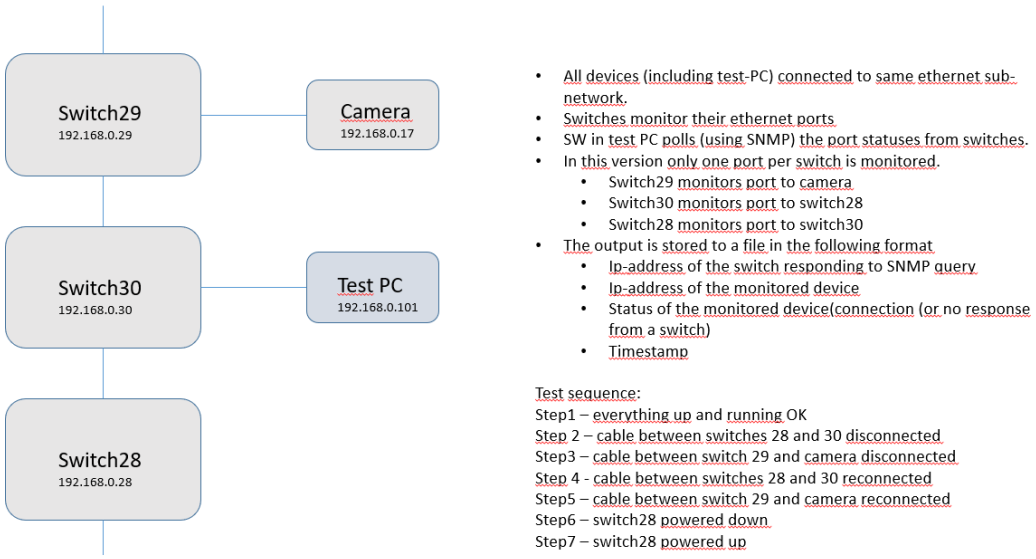
Opinnäytetyön toimeksiantona toimii InduSTAR hankkeeseen kuuluva Transtechin tekemä tilaus, joka sisältää uusiin raitiovaunuihin tulevan lisätyn todellisuuden huolto- ja vikatietojen visualisointiin keskittyvän sovellusdemon.

Demo toteutettiin ja sitä jatkokehitetään Magic Leap One -laitteelle, joka valittiin, koska siinä on viimeisimmät, laadukkaat ja monipuoliset lisättyyn todellisuuteen tarvittavat ominaisuudet ja komponentit.

Kehitysympäristönä AR-sovelluksella ovat Unreal Engine 4 ja Visual Studio, ja käyttöjärjestelmänä Windows. Nämä kyseiset kehitysympäristöt valittiin, koska Unreal Engine mahdollistaa graafisen pelimoottorin ja perinteisen ohjelmakehitysympäristön saumattoman ja helpon yhteiskehityksen. Pelimoottori sisältää visuaalisen maailman ja ikkunan hallinnan sekä nopean ohjelmoinnin ja koodin hallinnan. Visual Studio puolestaan mahdollistaa suorituskykyisen ja tehokkaan koodin kirjoittamisen ja virheenetsintätyökalujen käytön.

Kerronnassa oletetaan, että lukijalla on hallussa ohjelmoinnin perusteet sekä tietämystä Unreal Engine 4 -pohjaisesta ohjelmoinnista. Näin ollen kerronta keskittyy enemmän yleiskuvaan ja Magic Leap -kohtaiseen osuuteen, välttäen aiheesta irtautumisen ja opetusmateriaalin tarpeen, joka on helposti saatavissa muualta.

Demon ensimmäinen versio, jota käsitellään tässä opinnäytetyössä pitää sisällään paikallisen tietopankin, ja suljetun todellista vastaavan kolmen kuvitellun reitittimen, tietokoneen ja kameran tietoverkon (kuva 14). Tässä versiossa tietoverkko ja tietopankin data visualisoidaan käyttäjälle lisätyssä todellisuudessa ja mahdollistetaan vuorovaikuttaminen esitettyyn dataan. Demosovelluksen toteutukseen keskittyvässä osiossa kerrotaan ensin Magic Leap -laitejärjestelmään liittymättömistä ohjelmaosioista, koska tämä parantaa käsitystä ja yleiskuvaa Magic Leap -laitejärjestelmästä. Magic Leap -osioiden yhteydessä kerrotaan mitä Magic Leap -laiteympäristö vaatii ja tuo mukanaan kehitysprosessiin.



Kuva 14. Kuviteltu verkkokartta, jonka pohjalta demoa alettiin luomaan.

5.1 Työkohteena Transtechin raitiovaunu

Transtechin huolto- ja vikatiedon visualisoinnin demosovelluksessa on ideana, että verkossa olevasta tietopankista saadaan vaunukohtaiset vikatiedot ja huolto-ohjeet. Lisätyn todellisuuden sovelluksella voidaan tarkistaa vaunussa olevat viat, visualisoida niiden sijainnit fyysisessä maailmassa ja tarjota huolto-ohjeet käyttäjälle. Tämä mahdollistaa uuden huoltohenkilön tehokkaan ja itsenäisen työn.

Nykytilanne Transtechillä on se, että vikatiedot ovat saatavissa csv-tiedostossa luettelona. Uusilla huoltohenkilöillä on edessä vaunukohtainen perehdytys ja kokeneellakin huoltohenkilöllä menee runsaasti aikaa niin tietojen lukemiseen ulkoisesta lähteestä kuin vian etsintään. Ainoa tapa, jolla työntekoa on mahdollista tehostaa huomattavasti, on lisätty todellisuus. Tiedon tuonti suoraan käyttäjälle mahdollistaa jouhevamman työnteon ilman liikkumistarvetta ulkoisten tiedonlähteiden ja työpisteen välillä tai työkohteen sijainnin ymmärtämistä räjähdekuvien sekä tekstin avulla.

Demon päätavoite on luoda käytännön esimerkki tulevaisuuden ratkaisusta tehokkaaseen raitiovaunun huoltoprosessiin, jossa huoltohenkilön perehdytys sekä vaunun komponenttien sijainnin ja toiminnan tietämisen tarve on minimoitu.

Tavoitteena on, että se on lopullisessa muodossaan jo hyödynnettävissä huoltotöissä. Demon varsinainen tarkoitus on esitellä Transtechin johdolle lisätyn todellisuuden käyttömahdollisuuksia käytännössä ja tulevaisuudessa toimia mallipohjana varsinaiselle myöhemmin toteutettavalle käyttöön otettavalle lisätyn todellisuuden huolto- ja vikatietosovellukselle.

Ensimmäisen demoversion keskeinen toiminnallisuus on paikallinen omassa ajassa kulkeva vikatiedon keräys, joka ilmoittaa havaitut vikatiedot luettelosta ajan mukaan generoidusta vikatietoluettelosta. Järjestelmä sisältää paikkaan sidotut visuaaliset mallit verkkokartasta sekä jokaisen komponentin ja niiden yhteyksien visualisoinnin. Lisäksi käyttäjillä on mahdollisuus vikatietojen reaaliaikaiseen tarkasteluun lokitiedoista ohjelman sisällä.

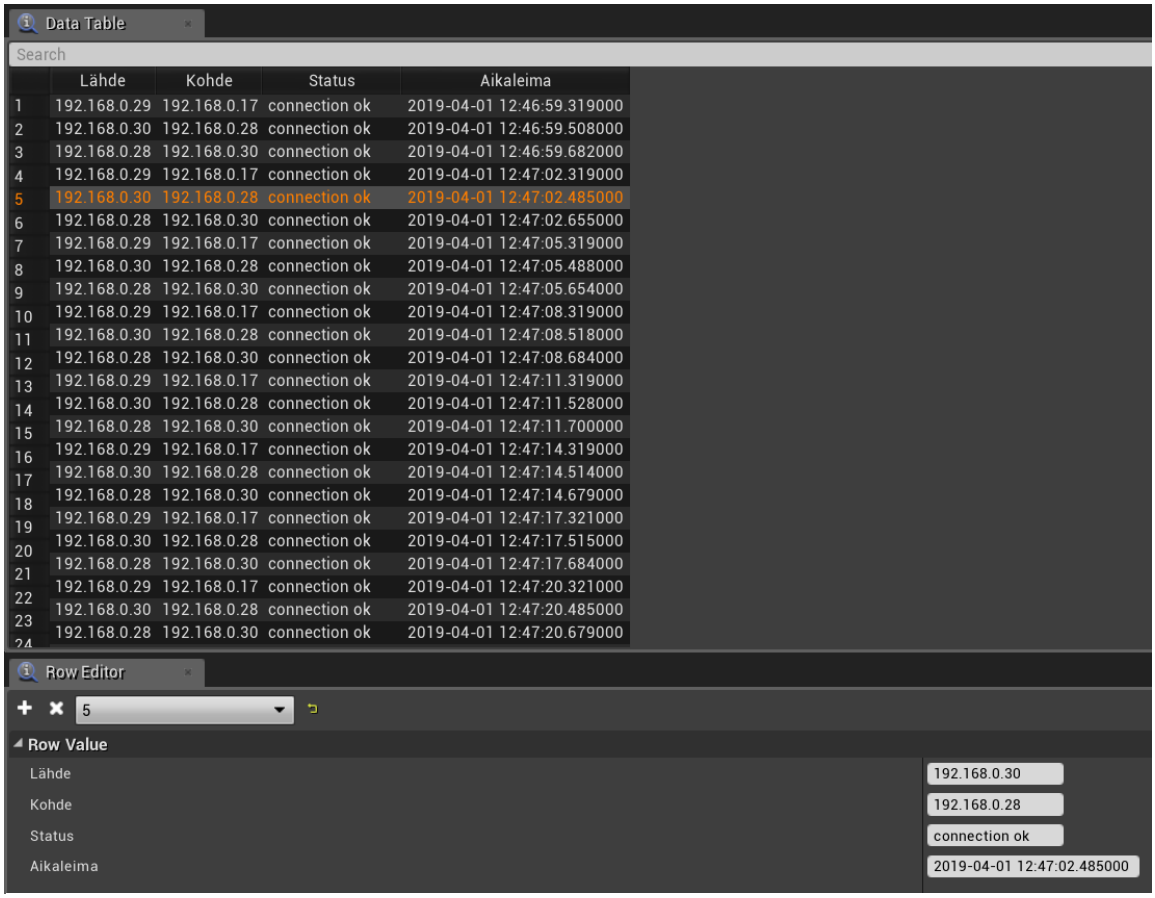
5.2 Toteutus

Sovellus pitää sisällään paikallisen tietopankin, verkkokartan virtualisoinnin, lokitietojen kirjaamisen sekä näiden hallinnoinnin. Ohjelman toiminnan kerronta on paloiteltu pienempiin osioihin. Osioita ovat paikallinen tietokanta, joka pitää sisällään vikatietojen hallinnan ja simuloidun ajan luennan. Verkkokartta, joka pitää sisällään ohjelman visuaaliset elementit ja tiedon hallinnan käsittelyn. Yhteystietojen visualisointi, jossa kerrotaan värikoodauksella toiminta verkkokartassa. Magic Leap -laitejärjestelmäkohtaisessa kehityksessä käydään läpi mitä lisättyyn todellisuuteen siirtyminen vaatii Magic Leap -laitejärjestelmän kanssa. Paikkaan sitomisessa kerrotaan, miten ohjelma on sidottu fyysiseen maailmaan ja sovelluksen ohjauksessa selitetään, miten lisätyn todellisuuden vuorovaikutus toimii. Toteutuksen lopuksi käydään läpi ensimmäisen demoversion tila opinnäytetyön kirjoitushetkellä ja miten sitä tullaan jatkokehittämään kohti lopullista versiota.

5.2.1 Paikallinen tietokanta

Kehitysprojektin ensimmäisessä vaiheessa ei käytetty oikeaa raitiovaunun vikatietokantaa, vaan kehitys toteutettiin Transtechin toimittaman paikallisen vikatietolokin perusteella. Vikatietolokin pohjalta luotiin paikallinen tietokanta Unrealin projektiin Data Table tiedostoon (kuva 15). Data Table tiedosto pitää sisällään määrittämättömän määrän Struct-tiedostoa, jolla ohjelmakehittäjä määrittää mitä säilötty data pitää sisällään. Kyseinen Struct määriteltiin sisältämään saman datan kuin alkuperäinen vikatietolista, joka pitää sisällään yhteyden lähde- ja kohdeosoitteet sekä yhteyden kuuluvuuden ja aikaleiman.

Tällä menetelmällä tietokanta saatiin siirrettyä sellaisenaan helposti käsiteltävään muotoon projektin käyttöön. Projektin sisäisenä tiedostona tietokanta on mahdollista lukea nopeasti suoraan keskusmuistista, mikä puolestaan poistaa tarpeen ylimääräiselle tiedostonlukemiselle. Tämä ominaisuus on käytössä vain demon ensimmäisessä vaiheessa, ja lopullista demoversiota varten toteutetaan suora yhteys raitiovaunun tietokantaan ja luetaan tietokannan muutokset reaaliajassa.



Kuva 15. Vikatietoluettelo tallennettuna Data Table tiedostoon.

Reaaliaikaisen tiedon päivitystä simuloimaan luotiin paikallinen BP_Listener-luokka, joka ilmoittaa sen hetkisen simuloidun ajan perusteella yhteyksien tilan (kuva 16). Tämän luokan tehtävänä

Kuva 16. BP_Listener luokan perustoiminnallisuus, jossa se simuloi aikaa ja ilmoittaa viimeisimmän tiedon yhteyksistä.

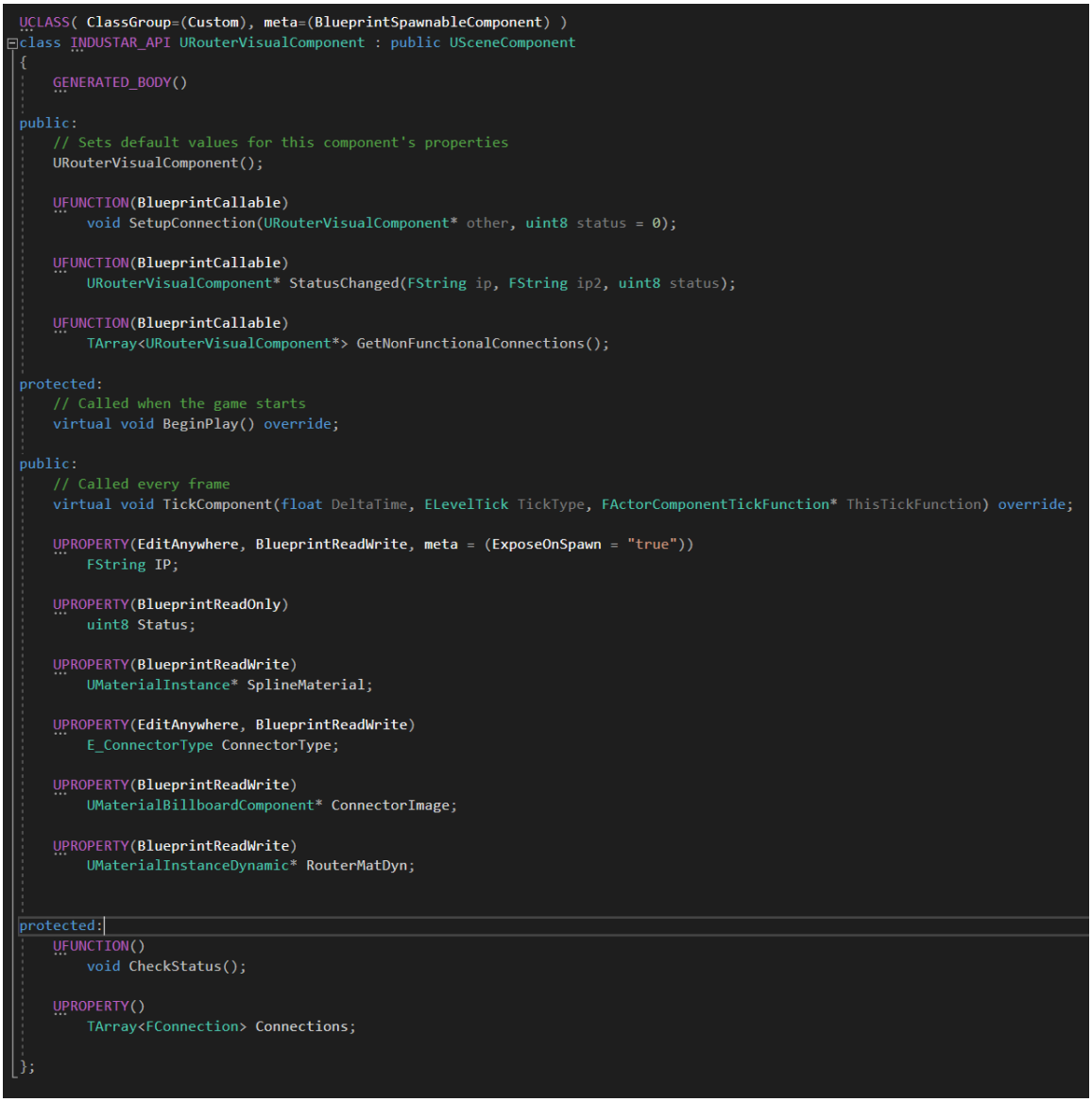
on lukea tietopankkia simuloidun ajan hetken kohdalta ja lähettää sen hetkinen tieto komponenteista ohjelman sisällä. Sen tehtävä on simuloida verkkoprotokola luokkaa tuomalla tietoa muulle sovellukselle, mutta sen tarkoituksena ei kuitenkaan ole simuloida itse verkkoliikennettä.

Koska vikatietoluettelo on pitkä, optimoitiin sen lukeminen yhteen riviin ohjelman päivityssyklin aikana. Tämä lähestymistapa poistaa lukemisesta aiheutuvan viiveen ilman tarvetta säikeistämiseen. BP_ Listenerin tehtäväksi jää siis lukea simuloidun ajan senhetkiseltä ajalta kaikkien yhteyksien tilat ja lähettää niistä ilmoitus. Tässä demoversiossa tilan lähetys toteutetaan viiden syklin aikana, jolloin datan lukeminen kaikille laitteille tapahtuu sekunnin kymmenesosan sisällä. Demon tarkoituksiin tämä lukemisnopeus on riittävä ja poistaa tarpeen monimutkaisemman järjestelmän kehitystyöltä, joka ei tule olemaan käytössä lopullisessa esittelyversiossa. Oikeassa ympäristössä tämä vastaa tietoverkosta yhteyden tilan tarkistamisen jälkeistä toimintaa ja ohjelman sisäistä keskustelua.

5.2.2 Verkkokartta

Verkkokartan virtualisointi ja visualisointi toteutettiin tekemällä kolme luokkaa: URouterVisualComponent, BP_RouterInfoWidget sekä BP_Network.

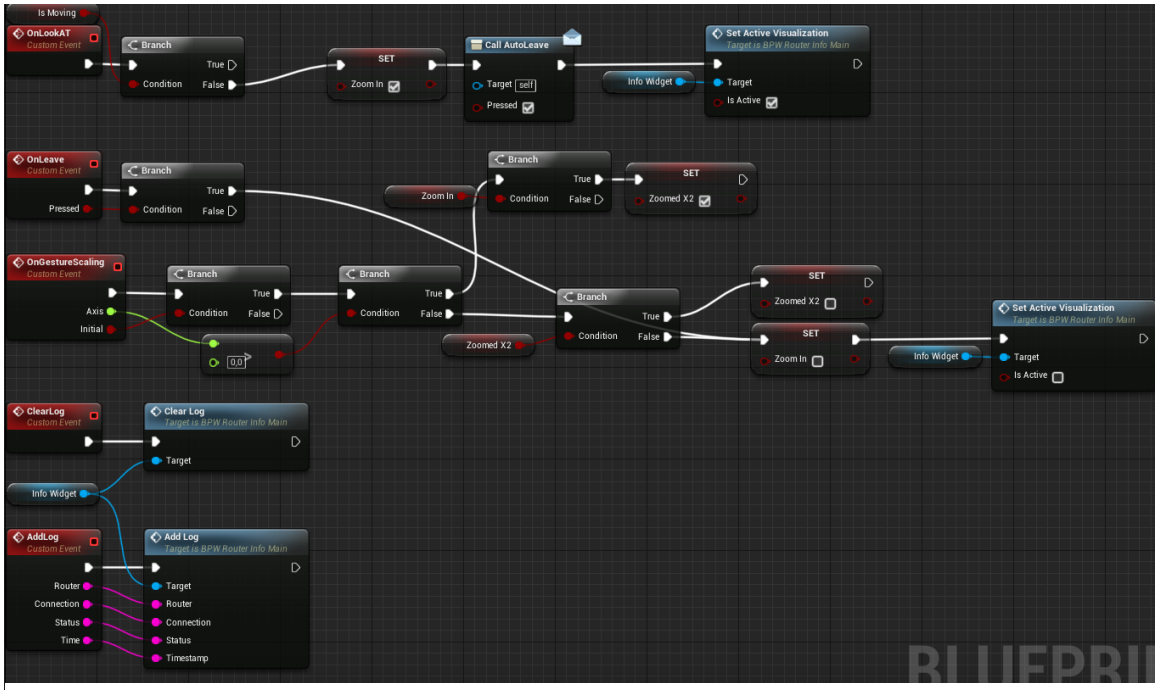
URouterVisualComponent pitää sisällään verkkokomponenttien tietojen ja visuaalisen toteutuksen sekä sijainnin virtuaalimaailmassa (kuva 17). URouterVisualComponent periytyy USceneComponentista saaden siltä avaruudenhallintatoiminnallisuuden, joka sisältää sijainnin, rotaation ja koon. Lisäksi sille on lisätty ominaisuudeksi mahdollisuus yhdistyä toiseen URouterVisualComponentiin ja luoda niiden välille visualisoitu yhteys. Sen lisäksi se pitää sisällään verkkokomponentin tiedot, materiaalit yhteyden tilan visualisointiin ja mahdollisuuden yhdistyä ulkopuoliseen luokkaan, joka hoitaa yhteyksien tilamuutoksien ilmoituksen. Paikallisessa demossa tämä luokka on BP_Listener. Myöhemmässä vaiheessa sen hoitaa erillinen luokka, joka on yhdistetty verkkoliikenteestä huolehtivaan luokkaan.



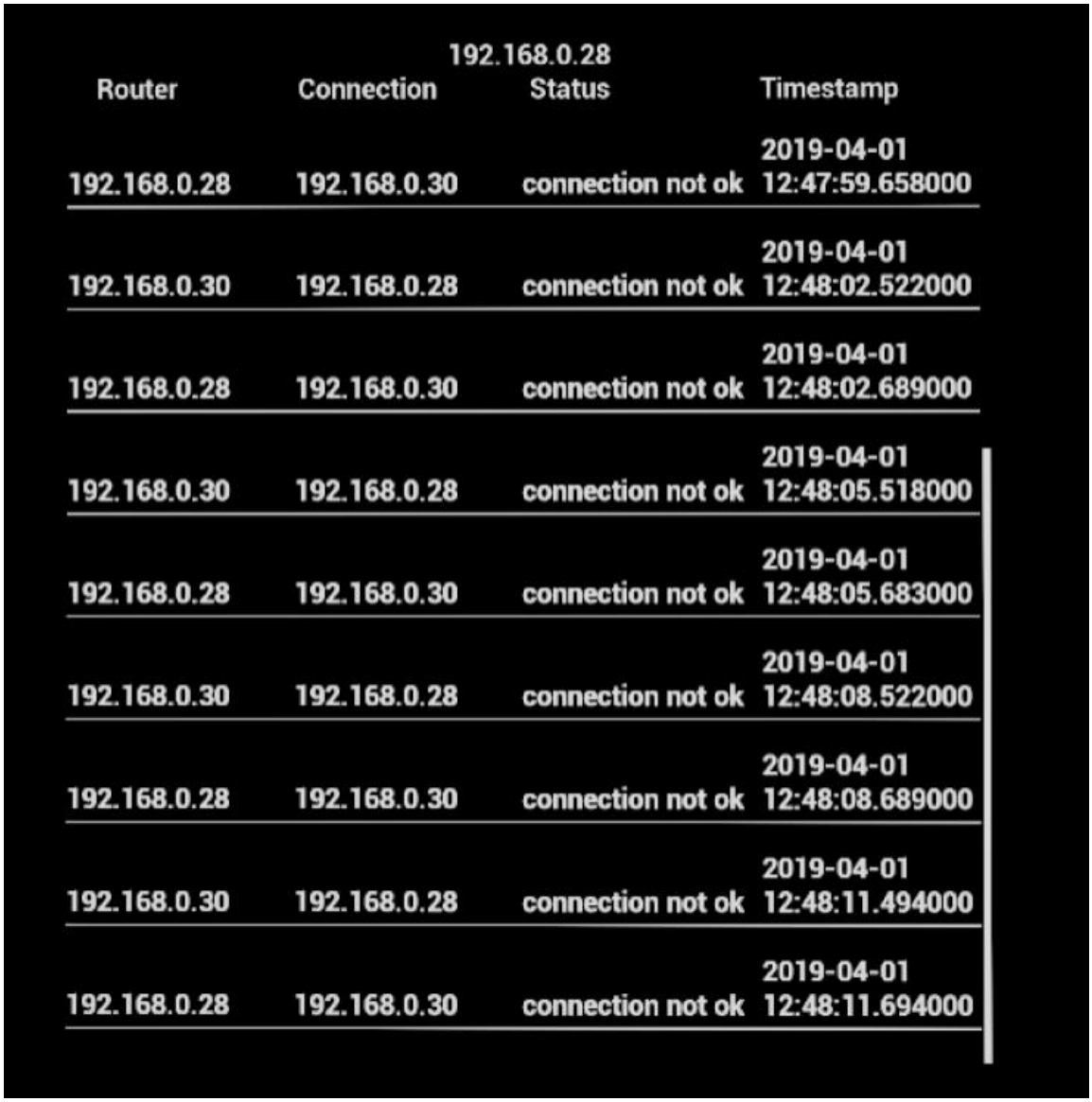
```
UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
class INDUSTAR_API URouterVisualComponent : public USceneComponent
{
    GENERATED_BODY()

public:
    // Sets default values for this component's properties
    URouterVisualComponent();

    UFUNCTION(BlueprintCallable)
        void SetupConnection(URouterVisualComponent* other, uint8 status = 0);

    UFUNCTION(BlueprintCallable)
        URouterVisualComponent* StatusChanged(FString ip, FString ip2, uint8 status);

    UFUNCTION(BlueprintCallable)
        TArray<URouterVisualComponent*> GetNonFunctionalConnections();

protected:
    // Called when the game starts
    virtual void BeginPlay() override;

public:
    // Called every frame
    virtual void TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction) override;

    UPROPERTY(EditAnywhere, BlueprintReadWrite, meta = (ExposeOnSpawn = "true"))
        FString IP;

    UPROPERTY(BlueprintReadOnly)
        uint8 Status;

    UPROPERTY(BlueprintReadWrite)
        UMaterialInstance* SplineMaterial;

    UPROPERTY(EditAnywhere, BlueprintReadWrite)
        E_ConnectorType ConnectorType;

    UPROPERTY(BlueprintReadWrite)
        UMaterialBillboardComponent* ConnectorImage;

    UPROPERTY(BlueprintReadWrite)
        UMaterialInstanceDynamic* RouterMatDyn;


protected:
    UFUNCTION()
        void CheckStatus();

    UPROPERTY()
        TArray<FConnection> Connections;

};
```

Kuva 17. URouterVisualComponent.

BP_RouterInfoWidget toimii virtuaalisena lokikirjana ja sisältää tiedon käsittelyyn liittyvien ohjauskomentojen toteutuksen. Se sisältää myös BPW_RouterInfon, joka on käyttöliittymäelementti. BPW_RouterInfo puolestaan hoitaa vikatietojen visualisoinnin käyttäjälle. Se toimii reagoiden käyttäjän katseella tai ohjaimella osoittaessa mahdollistamalla vuorovaikuttamisen loki elementtiin sekä ohjaimella että eleillä (kuva 18). Käyttäjä voi kaapata lokin ja tuoda sen luoksensa helpon lukemisen mahdollistamiseksi. BPW_RouterInfo taas pitää sisällään itse lokikirjan ja sen selaamisen mahdollistavat toiminnot, joita voidaan käyttää, kun loki on tuotu käyttäjän luokse lukemistilaan (kuva 19).

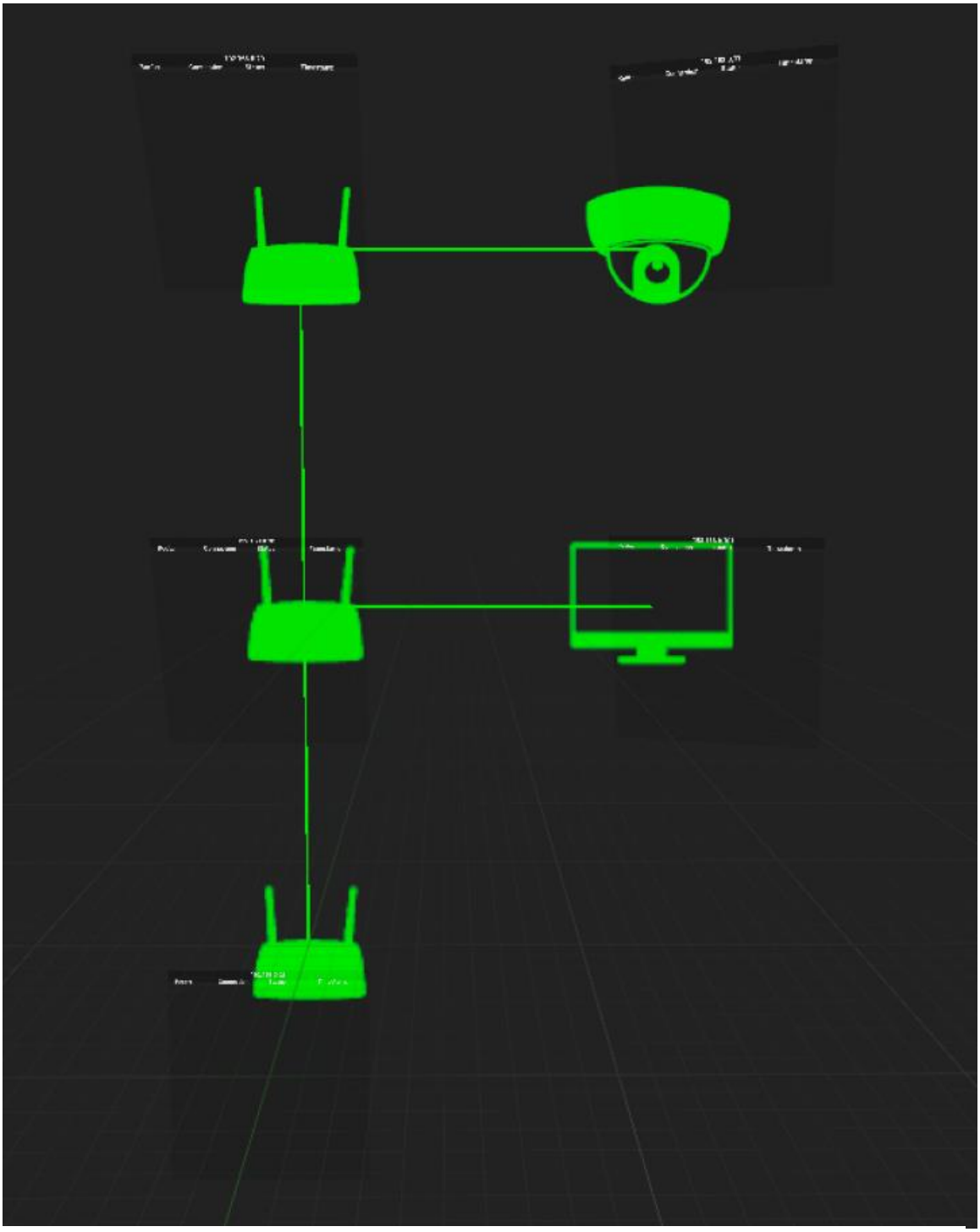


Kuva 18. Peruslogiikka vuorovaikutukseen BP_RouterInfoWidgetin sijaintiin sekä lokitietojen suoraan käsittelyyn.



Kuva 19. BPW_RouterInfo käyttöliittymä elementti käytössä.

BP_Network toimii emoluokkana, joka sisältää kaikki reitittimet (kuva 20). Se myös yhdistää ne toisiinsa ja simuloituun verkkoliikenteeseen, jota BP_Listener pitää yllä paikallisesta tietopankista. Sen pohjimmainen tarkoitus on mahdollistaa koko tietoverkon käsittely yhtenä palasena ja huolehtia sen tietojen alustuksesta. Luokka huolehtii siitä, että siihen asetetut komponentit löytävät toisensa mahdollistaen niiden välisen keskustelun. Kun virtuaalisilla komponenteilla on parempi käsitys toisistaan, pystytään visualisoimaan niiden yhteistoiminta ja viat paremmin samalla datamäärällä kuin mitä oikea vastaava järjestelmä käyttää.

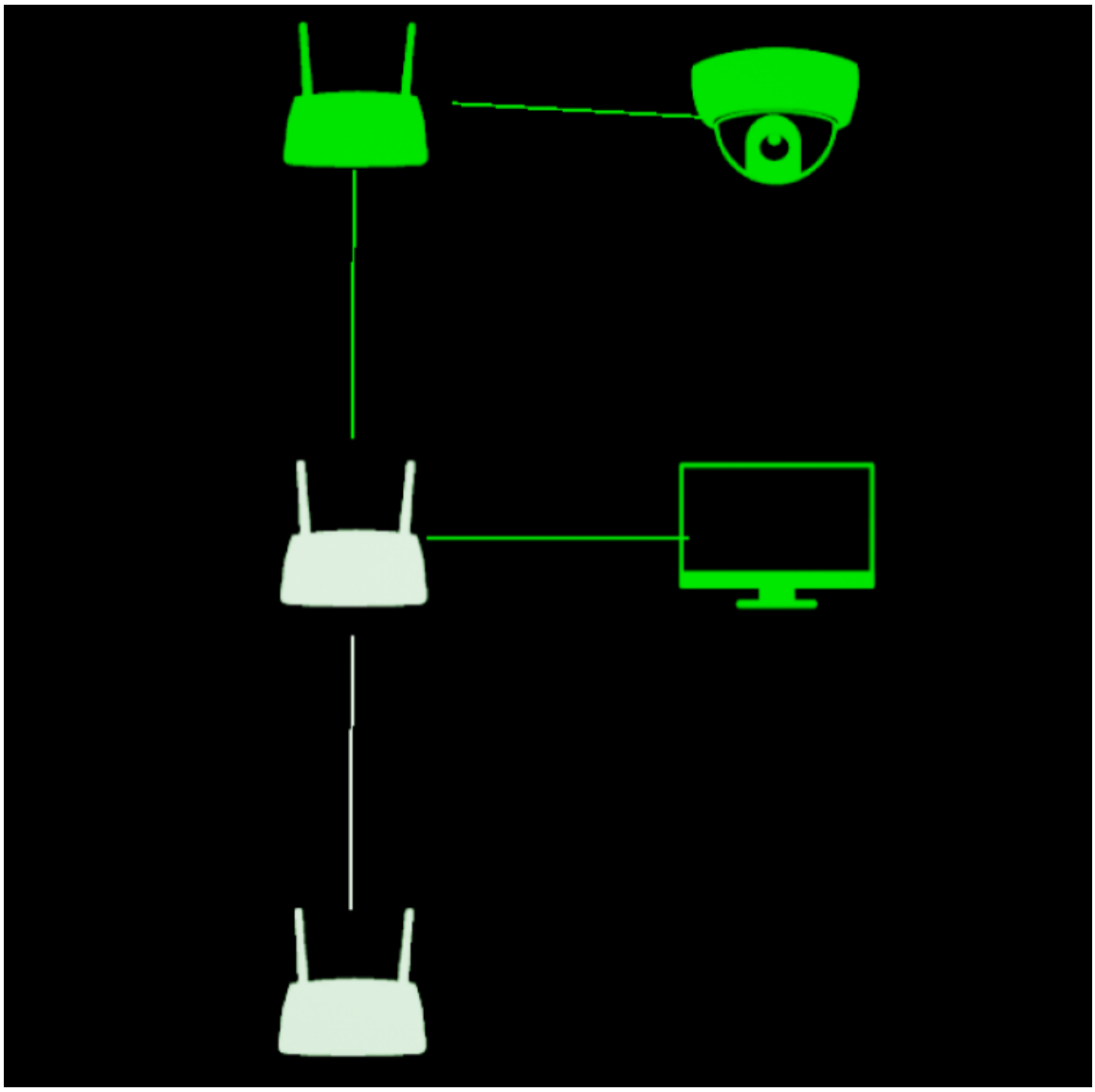


Kuva 20. Tietoverkon graafinen visualisointi lokeineen BP_Network luokassa.

5.2.3 Yhteystiedon visualisointi

Lokitietoihin lisätään virheelliset yhteydet ja tuodaan käyttäjälle näkyviin BPW_RouterInfo käyttöliittymäelementtiin, jonka omistaa BP_RouterInfoWidget. Jokaisella verkkokomponentilla on oma loki. Lisäksi järjestelmässä on yksi yhteinen loki, johon kirjataan kaikki vikatiedot.

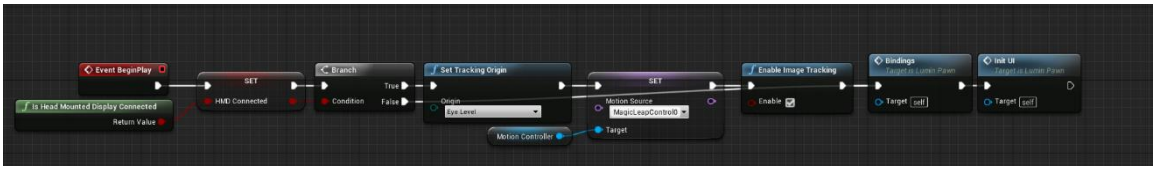
Yhteydet visualisoidaan myös muuttamalla verkkokomponenttien ja niiden välisten yhteyksien väritystä vihreästä valkoiseksi ongelmien ilmestyessä ja punaiseksi yhteyden kadotessa (kuva 21). Vikatietojen käsittely on toteutettu delegaatiojärjestelmällä. Sen toimintaperiaatteena on, että kun sovellus saa tiedon päivitetystä vikatiedosta, se lähetetään ohjelman sisäisenä kutsuna. URouterVisualComponentit ottavat sen vastaan ja vertaavat sen ID:tä omaansa. ID:n täsmätessä URouterVisualComponent lukee delegaatin pitämän vikatiedon ja päivittää sen omiin, vastinkappaleensa ja niiden välisen yhteyden tietoihin. Samalla se muuttaa visualisoinnin väritystä uudelleen värikoodin mukaan. Lopuksi se päivittää vielä sitä koskevan lokin kutsumalla sen omistavaa BP_RouterInfoWidgetiä, samalla kun sen vastinkappalekin päivittää oman lokinsa



Kuva 21. Verkkokomponentit muuttavat väritystään yhteyden tilan mukaan.

5.2.4 Magic Leap -laitejärjestelmäkohtainen kehitys

Normaalisti Magic Leap One kysyy ohjelman tarvitsemia lupia vain ensimmäisellä käynnistys kerralla. Käyttäjää helpottamaan voi tehdä yksinkertaisen tarkistuksen, joka kysyy tarvittavat luvat joka kerta, mikäli niitä ei ole jo myönnetty. Magic Leap -Pluginista löytyy jo valmiiksi ASyncissä toimiva luvankysely, joka mahdollistaa luvan kysymisen vastauksen saamisen delegaatilla ilman, että ohjelman suoritus pysähtyy luvan kysynnän ajaksi.



Kuva 22. Sama Pawn-luokka asetetaan käyttäytymään eri tavalla riippuen siitä, onko käytössä päähän puettava laitejärjestelmä.

Demoympäristöä tullaan käyttämään Magic Leap -laitejärjestelmän lisäksi myös eri laitekokoonpanoilla. Eri alustat voidaan huomioida kehitystyössä joko omilla Pawn-luokillaan tai rajapintojen toimiessa kaikilla laitteilla voidaan käyttää samaa Pawn-luokkaa kaikilla alustoilla. Pawn-luokka toimii Unreal Enginen ”pelaajahahmona”, jolla hallinnoidaan kameran liikuttelua. On kuitenkin huomioitavaa, että jokaiselle alustalle on tehtävä oma vuorovaikuttamisen toimintamalli ja monimutkaisessa sovelluksessa se aiheuttaa paljon päällekkäisiä toimintoja. Siksi on suotavaa tehdä alustoille omat Pawn-luokat ohjelman siisteyden ja kehittämisen helpottamiseksi. Alustakohtaisilla vuorovaikutustoimintamalleilla voidaan yksilöidä käyttökokemus alustan mukaan ja mahdollistaa päähän puettavalle laitteelle tehdyn sovelluksen käyttö myös näyttöpäätteellä (kuva 22). Tällä tavalla mahdollistetaan myös nopeampi ohjelman testaus laitteella, jolla on kehitysympäristö. Testatessa ohjelman ominaisuutta, joka ei tarvitse laitekohtaista laitteistoa toimiakseen, voidaan toimivuutta testata nopeammin kehitysympäristössä, koska ohjelman ajo ulkoiselle laitteelle kestää kauemmin, vaikka käytössä olisikin reaaliaikainen ajo.

Päähän puetun lisätyn todellisuuden menetelmät pelaajahahmon kanssa ovat samat kuin virtuaalitodellisuudessa. Tracking Originin eli sijainnin seurannan alkupisteen määrittämällä voidaan käyttää joko päähineen nykyistä sijaintia nollapisteenä seurannassa tai vaihtoehtoisesti laitteen oman kalibroinnin nollapistettä. Virtuaalinen kamera on vapaasti liikutettavissa, mutta se pysyy silti aina pelaajahahmon lapsena. Tällä tarkoitetaan sitä, että sen virtuaalisen kameran sijainti on aina relatiivinen käyttäjän toimintoja käsittelevään luokkaan ja pelaajahahmon sijaintia liikuttaessa virtuaalisen kameran sijainti liikkuu samassa suhteessa.

Seurattava ulkoinen ohjain toimii Motion Controller -luokalla, joka mahdollistaa ohjaimen sijainnin huomioimisen virtuaalimaailmassa. Motion Controller on Unreal Enginen oma virtuaalitodellisuuteen tehty luokka, joka mahdollistaa sijainniltaan seurattavan ohjaimen luontevan käytön virtuaalimaailmassa.

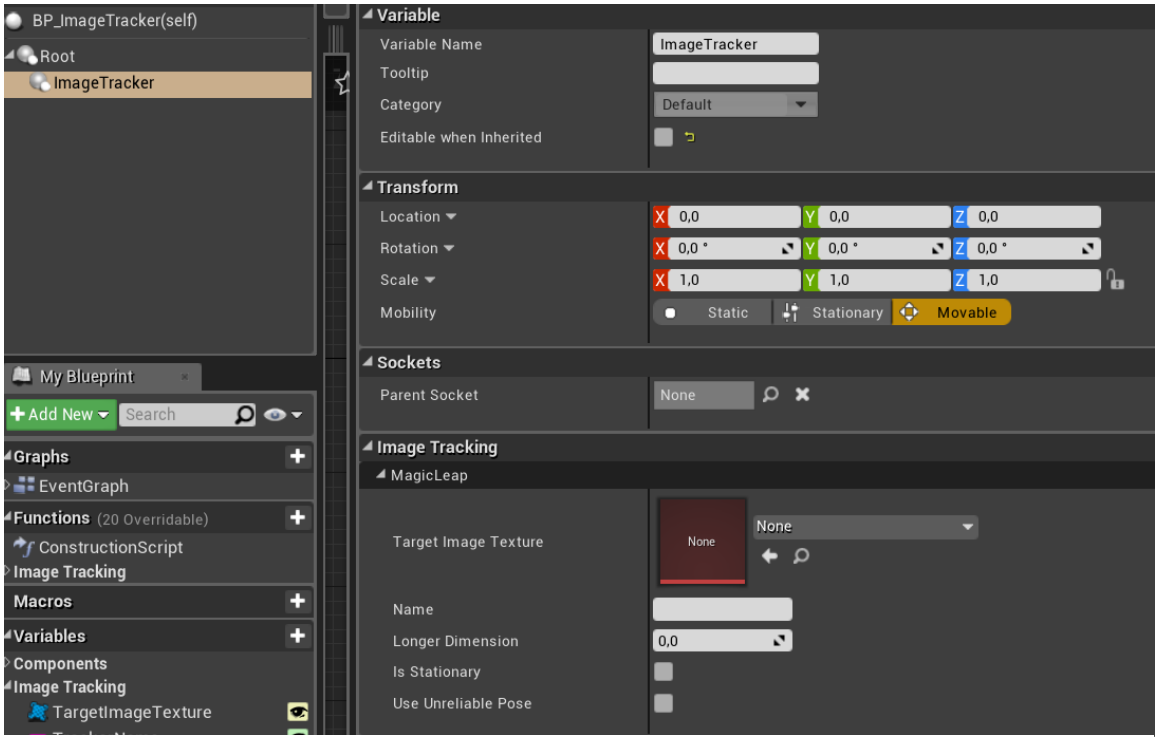
Muuten kehitysmenetelmät ovat varsin yhteneväiset alustasta riippumatta, lukuun ottamatta lisätyn todellisuuden mukanaan tuomia lisäominaisuuksia.

Lisätty todellisuus on käsitelty Unreal Enginessä siten, että maailman keskipiste pysyy paikallaan laitteen asennon ja sijainnin mukaan. Maailman keskipisteen sijaintia voidaan siirtää laitteen paikkaan sitomisen yhteydessä siten, että käyttäjän sijainti ei muutu samalla. Fyysisen maailman käsittely ja hahmottaminen tapahtuu kolmannen osapuolen kirjastossa, tässä tapauksessa MLSDK:lla joka tulee Lumin SDK:n mukana. Unreal Enginen mukana tulee ARCore ja ARKit, jotka tekevät vastaavan työn Android ja IOS laitteille.

5.2.5 Paikkaan sitominen

Magic Leap One -laitteessa ei ole sisään rakennettua GPS-vastaanotinta vaan sen sijainti on sama kuin WiFi-verkon IP-osoitteen, johon se on yhdistetty. Magic Leap Onen -järjestelmän kohdalla onkin parempi turvautua kuvatunnistuspohjaiseen sijainnin tarkastamiseen paikkaan sidotussa lisätyssä todellisuudessa.

Kuvatunnistus voidaan tehdä itse tai kolmannen osapuolen kirjastolla suoraan kameroiden raakakuvasta, mutta MLSDK pitää sisällään jo tehokkaan kuvatunnistuksen, jolla voidaan laskea myös kuvan tarkka etäisyys ja asento (kuva 23).



Kuva 23. UImageTracker komponentti, joka kuvan löytäessään siirtyy sijainniltaan vastaavaan kohtaan virtuaalimaailmassa.

MLSDK:n pelimoottorirajapinnan kuvatunnistus on kuitenkin suunniteltu ainoastaan yhden kuvan seurantaa varten. Tämä tarkkuus on riittävä kehitetyn demon päätarpeisiin, koska virtuaalisen maailman kalibrointi voidaan toteuttaa raitiovaunuun siirryttäessä. Demoa varten kuvantunnistus järjestelmää kuitenkin monipuolistettiin mahdollistamaan usean kuvan ja samalla yksinkertaisen objektien etsinnän maailmasta. Tämä mahdollistaa digitaalisen informaation tuonnin käyttäjälle fyysisen maailman raitiovaunun komponentteihin sidonnaisena. Samalla se myös mahdollistaa virtuaalisen maailman uudelleen asettelun laitehäiriön tai epäonnistuneen alkuperäisen paikantamisen jälkeen ilman käyttäjän toimenpiteitä.

Objektien etsintä toteutettiin tekemällä kuvasarjoja sisältävä Data Table tietokanta. Kuvasarjoihin voidaan lisätä seurattavan objektin kuvia eri kuvakulmista, antaen lisäksi jokaiselle kuvalle oman relatiivisen sijainnin ja asennon. Objektitunnistusta voidaan parantaa edelleen poistamalla ku-

vista taustat. Tämän kuvapankin käsittelyn hoitaa UMultiImageTracker-luokka, joka pitää sisällään dynaamisesti luotavan määrän UImageTracker-luokkia (kuva 24). Se mahdollistaa useamman kuvan tarkistuksen, jossa kuvasarja vastaa samaa kohdetta fyysisessä maailmassa eri kuvakulmista. Hyvin optimoituna tämä lähestymistapa mahdollistaa objektipohjaisen sijainnin tunnistuksen relatiivisesti vähällä laskennan lisäämisellä.

```
UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
class INDUSTAR_API UMultiImageTracker : public USceneComponent
{
    GENERATED_BODY()

public:
    UMultiImageTracker();

    UFUNCTION(BlueprintCallable, Category = "ImageTracking|MagicLeap")
    void InitTracking(TArray<UTexture2D*> TargetImageTextures,FString Name, float LongerDimension, bool bIsStationary, bool bUseUnreliablePose);

    UFUNCTION(BlueprintCallable, Category = "ImageTracking|MagicLeap")
    UImageTrackerComponent* GetImageTracker(int Index);

    virtual void TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction) override;

    void SetAllTicks(bool enabled);

    UFUNCTION(BlueprintCallable, Category = "ImageTracking|MagicLeap")
        void SetEnabled(bool enabled);

protected:

    UFUNCTION(BlueprintInternalUseOnly, Category = "ImageTracking|MagicLeap")
    void TargetChange(AImageTrackerParent* Index,bool bFound);

public:

    DECLARE_DYNAMIC_MULTICAST_DELEGATE_OneParam(FTargetFound, const int&, Index);

    UPROPERTY(BlueprintAssignable)
        FTargetFound OnTargetFound;

private:

    int Index;
    TArray<AImageTrackerParent*> TargetTrackers;
};
```

Kuva 24. UMultiImageTracker säiliöluokka, joka käsittelee kuvasarjaan liitettyjä UImageTracker luokkia ja niiden laskentaa.

UMultiImageTracker luokan perustehtävänä on tietää, mistä kuvasarjasta se on luotu. Lisäksi se huolehtii UImageTracker-luokkien laskennasta siten, että samaan aikaan laskettavien UImageTracker-luokkien määrä voidaan minimoida. Kuvan löytyessä muiden kuin kuvan löytäneen luokan toiminta pysäytetään kokonaan (kuva 25). Sen tehtävä on myös tietää, mitä kuvaa kuvasarjasta kuvan löytänyt UImageTracker edustaa.

Tämän lisäksi projektiin lisättiin BP_CalibrationRoom-luokka, johon on lisätty funktionalisuus manuaaliseen virtuaalimaailman liikutteluun, käyttäjän tarpeiden mukaan.

```
void UMultiImageTracker::TargetChange(AImageTrackerParent* Container,bool bFound)
{
    Index = TargetTrackers.Find(Container);
    if (bFound)
    {
        OnTargetFound.Broadcast(Index);
        SetAllTicks(false);
        TargetTrackers[Index]->TargetTracker->SetComponentTickEnabled(true);


        return;
    }
    OnTargetFound.Broadcast(-1);
    SetAllTicks(true);
}
```

Kuva 26. Yksinkertainen automaattinen laskennan lopetus kuvan löytyessä ja sen uudelleen käynnistys löydetyn kuvan sijainnin kadotessa.

Magic Leap -laitteessa on myös mahdollisuus tarkkaan fyysisen maailman virtualisointiin tehtäessä merkkaamatonta lisättyä todellisuutta, joskaan tässä projektissa sille ei ole tarvetta, koska se sisältää vain koordinaattisidonnaisia virtuaalisia objekteja (kuva 26).

Kuva 25. Maailma virtuaalisesti visualisoituna Magic Leap -laitejärjestelmällä.

5.2.6 Sovelluksen ohjaus

Virtuaalisen maailman vuorovaikuttaminen on hoidettu sekä ohjaimella että eleillä.

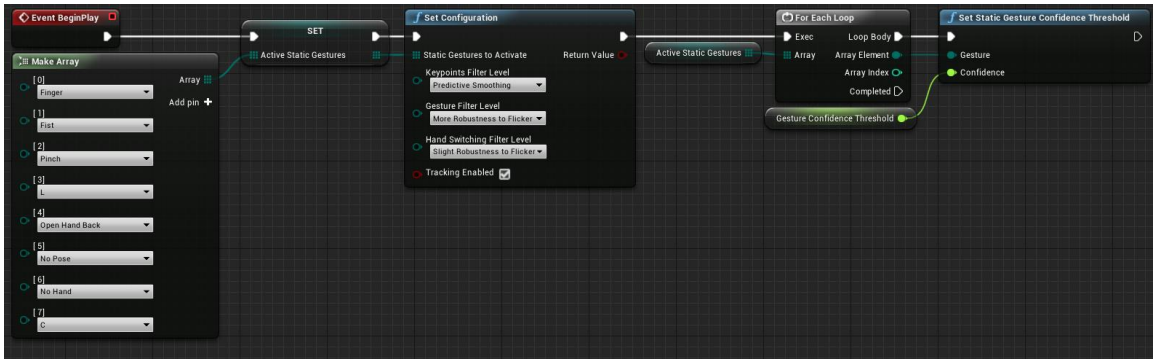
Näistä ohjainratkaisu on monelle vielä luonnollisempi ja helppokäyttöisempi, koska se mahdollistaa fyysisen laitteen käytön, jolla voidaan olla vuorovaikutuksessa virtuaalisen maailman kanssa. Magic Leap -laitejärjestelmä mahdollistaa myös tarkan käsien seurannan, joten ympäristöön on mahdollista toteuttaa myös monipuolinen eleiden seuranta (kuva 27). Eleseuranta mahdollistaa täysin ohjainlaiteriippumattoman vuorovaikutuksen virtuaalimaailmaan.

Kuva 27. Käden kiintopisteet tunnistettuna MLSDK-kirjastolla.

Ohjainratkaisussa ohjaimen sijainti on fyysisessä ja virtuaalisessa maailmassa samassa paikassa. Ohjain toimii osoituskynänä, jonka avulla voidaan avata ja sulkea osoitettu lokielementti. Kosketuslevyn avulla voidaan selata lokin tietoja.

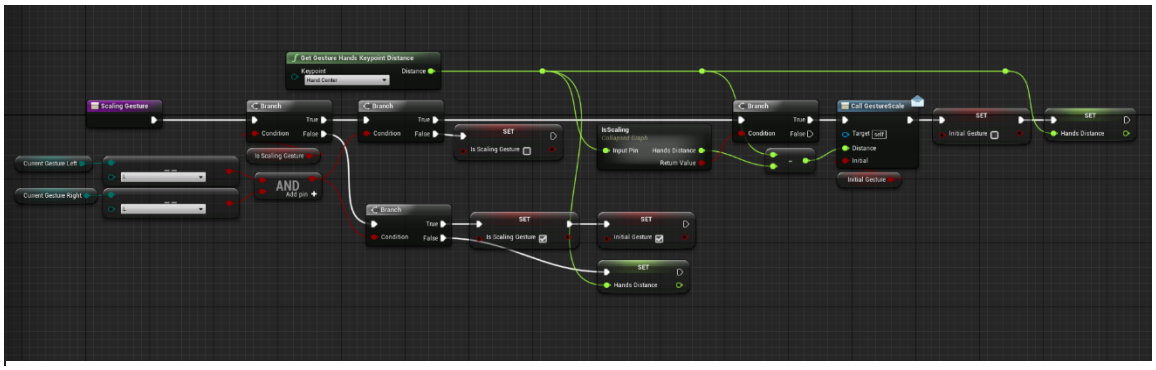
Elepohjaisessa ratkaisussa lokielementti avataan ja suljetaan kosketusnäytön zoomausta vastaavalla eleellä, jossa ovat käytössä molemmat kädet. Molempien käsien käytöllä varmistetaan, että liike on tahdonalainen. Lokielementti valitaan katseella ja tietoa voidaan selata osoittamalla.

Pelimoottorirajapinnan mukana tulee eleiden ja ohjaimen painikkeiden tunnistuksen perustoiminta. Kehittäjän vastuulle jää niiden sulava yhdistäminen toiminnallisuuteen. Käytännössä tämä tarkoittaa eleiden tarkistuksen toleranssin asetusta ja niiden yhdistämistä sovelluksen funktionalisuuteen (kuva 28). Tämä ei kuitenkaan pelkästään riitä sujuvan käyttäjäkokemuksen luomiseen, koska eletunnistus ilmoittaa aina, kun se tunnistaa eleen kuvasta. Tämä tarkoittaa käytännössä sitä, että eleille on tehtävä oma seuranta ja jossain tapauksissa myös useamman eleen ja liikkeen seuranta. Eleseurannan tehtävänä on varmistaa, että liikkeet ovat tahdonalaisia, eivätkä käyttäjän vahinkoja tai normaaleja käden asennon muuttamisia.



Kuva 28. Eleiden tarkistuksen alustaminen ja tunnistus toleranssin asettaminen.

Toiminnot, kuten vieritys ja skaalaus tarvitsevat laskuihin mukaan käden liikkeen. Liikkeen seurannalla voidaan määrittää esimerkiksi, kuinka paljon skaalausta tehdään (kuva 29). Luonnollinen toimintaperiaate eleillä on, että ne toimisivat hyvin samalla tavalla kuin kosketusnäytön eleet. Jos tarvitaan ele, joka toimii kosketusnäytöllä mutta ei vastaavasti kolmiulotteisessa maailmassa, tulisi eleen olla samankaltainen kuin vastaavassa tilanteessa fyysisessä maailmassa tehtävä käsien luontainen liikuttelu. Luonnollinen tai jo opitun liikkeen yhdistäminen eleisiin tekee sovelluksesta sekä mukavamman että helpomman käyttää.



Kuva 29. Skaalauseleen tarkistus.

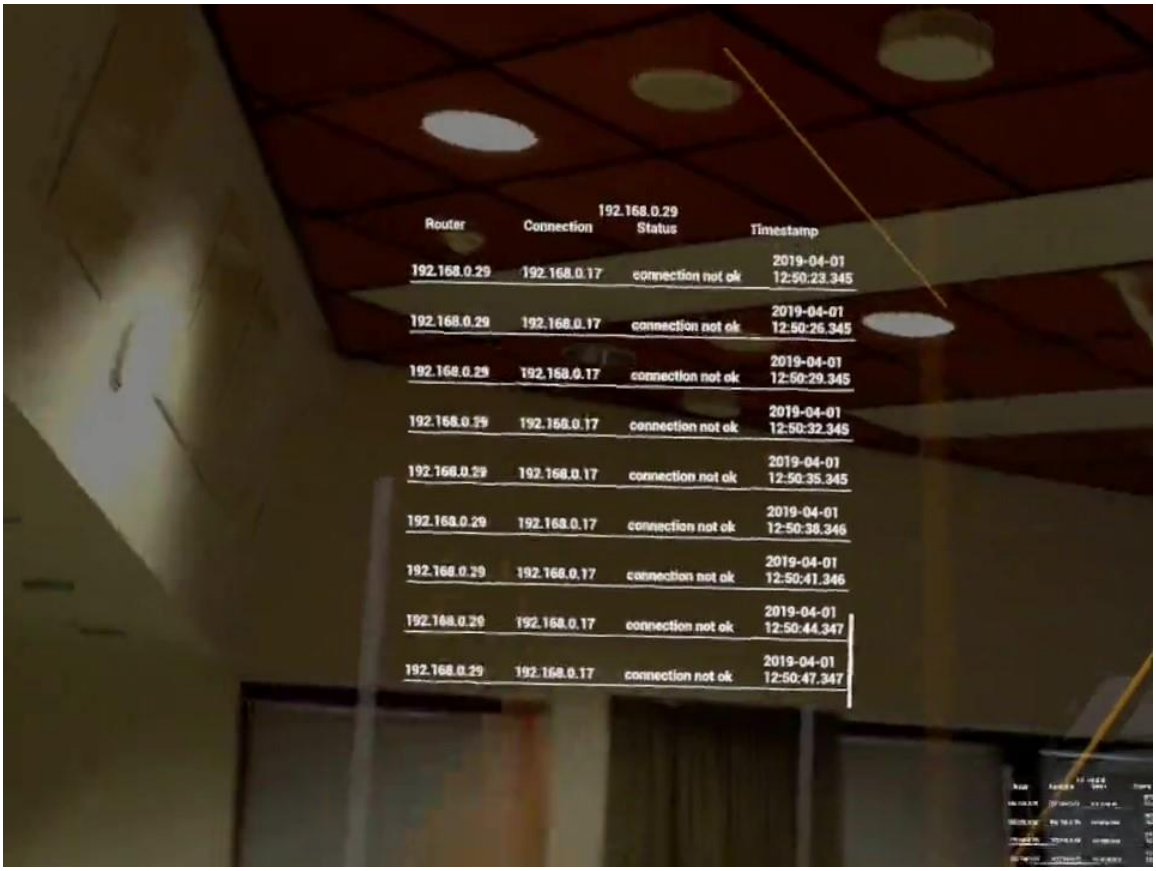
5.3 Demon nykytila

Nykytilassaan demolla pystytään karkealla tasolla havainnollistamaan lisätyn todellisuuden mahdollisuuksia huolto- ja vikatiedon visualisoinnissa. Se esittelee, miten fyysiseen maailmaan pystytään lisäämään virtuaalista tietoa sulauttaen se käyttäjälle samaan kokemukseen. Demo sisältää luonnollisen tavan tarjota käyttäjälle käyttöön informaatiota, ilman haittaa käyttäjän toimintakyvylle esimerkiksi niin, että se estäisi toisen käden käytön.

Demo sisältää paikkaan sidotun tietoliikennejärjestelmän visualisoinnin, lokin kirjoittamisen ja lukemisen sekä raitiovaunun visualisoinnin testiympäristössä (kuva 30). Järjestelmä sisältää mahdollisuuden objekti- ja kuvapohjaiseen paikannukseen sekä käyttäjän vuorovaikutuksen joko ohjaimen tai eleiden kautta. Lisäksi virtuaalinen maailma on mahdollista kohdentaa manuaalisesti. Järjestelmän verkkoliikenne on simuloitu paikallisesti.

Kuva 30. Transtech demo testiympäristössä. Kuvassa kolme reititintä ja tietokone lokeineen.

Lisätyn todellisuuden vuorovaikutus on pyritty tekemään mahdollisimman luontevaksi yhdistämällä katseen, eleiden ja ohjaimen toiminnot yhdessä toimiviksi samaan tapaan kuin vastaava vuorovaikuttaminen tehtäisiin fyysisessä maailmassa vastaaviin kohteisiin.



Kuva 31. Kuvan 30 etualalla olevan reitittimen lokitiedot testiympäristössä.

5.4 Projektin tulevaisuus

Projektin tulevaisuudessa sen komponenttijärjestelmää viedään edelleen mallipohjaisempaan suuntaan, joka mahdollistaa saman logiikan käytön muissakin järjestelmissä tietoverkon lisäksi.

Käyttöliittymään tullaan lisäämään enemmän graafisia elementtejä. Vikatietolokit yhdistetään ja kategorisoidaan. Virtuaalisen maailman visualisointitapaa muutetaan siten, että oletuksena käyttäjä näkee vain vialliset komponentit tai sen hetken tarkastelussa olevan komponentin. Viallisten komponenttien löytämiseen tullaan myös toteuttamaan paikannusta helpottava moniulotteinen kompassi.

Demosta tehdään enemmän komponenttipohjainen ja ulkoasultaan valmiin kategorisoidun sovelluksen näköinen, nykyisen raa’an visualisoinnin sijaan.

6 Pohdinta

Lähivuosina lisätty todellisuus tuo samansuuruisen muutoksen ihmisten elämään kuin älypuhelimiin siirtyminen aikanaan. Tämän hetken teknologisten rajoitusten ja kustannusten takia järjestelmä ei kuitenkaan vielä ole soveltuva kuin suljettuihin ympäristöihin. Lisätyn todellisuuden laitteet ovat vielä kehitysvaiheessa. Ihmiselle helposti omaksuttavia informaation esittämistapoja testataan parhaillaan ja lisätyn todellisuuden käyttökohteet keskittyvät puetuilla laitteilla lähinnä teollisuuteen ja kädessä pidettävillä laitteilla viihdeteollisuuden puolella. Teknologian ja standardien kehittyessä se on kuitenkin suhteellisen helposti siirrettävissä jokapäiväiseen käyttöön.

Ratkaiseva käännekohta lisätyn todellisuuden teknologian yleistymisessä tulee olemaan se hetki, kun päähän puettava lisätty todellisuus saadaan sovitettua aurinkolaseja vastaavaan laitteeseen kokonsa ja painonsa puolesta. Laitteiden tulisi myös toimia puhelimen näyttöpäätteenä tai laskenta olisi syytä siirtää kokonaisuudessaan verkkopalveluihin. Siirtyminen laitteen ulkopuoliseen laskentaan poistaisi tarpeen ylimääräiselle laskentayksikölle ja mahdollistaisi lasien ulkomuodon ja koon muuntumisen katukuvaan sopiviksi. Lisäksi esimerkiksi puhelimen näyttöpäätteenä lisätyn todellisuuden lasit mahdollistavat eleohjauksella sen, että puhelinta ei missään vaiheessa tarvitse ottaa taskusta. Puhelinta siis ohjattaisiin AR-lasien kautta eleillä ja sen käyttöjärjestelmän kuva voitaisiin piirtää käyttäjän asettamaan kohtaan näkökentässä joko fyysiseen paikkaan tai käyttäjään sidottuna.

Toinen ratkaiseva käännekohta on lisätyn todellisuuden ratkaisujen tarjoaminen saman alustan tai verkkosovelluksen kautta. Tämä mahdollistaisi sen, että sijainnin tai haun mukaan voidaan toimittaa pilvipalvelusta yksilöityä sisältöä käyttäjälle. Esimerkiksi kivijalkakauppaan mentäessä käyttäjälle voidaan esittää sijainnin perusteella kaupan tuotteet sekä niiden sijainti hyllyissä. Tuotteet voisi jopa maksaa ilman tarvetta erillisille laitteille jo puettuna olevien lasien lisäksi.

Näin pitkälle kehittynyt lisätty todellisuus ei ole vielä saatavilla missään muodossa ja se tuo mukanaan omat ongelmat, kuten mainoksien ja sisällön hallinta, puhumattakaan tietoturvaan vaadittavista päivityksistä.

Nykytilassa lisätyn todellisuuden laitteet, kuten Magic Leap One ovat lähinnä kehitysalustoja, joilla pystytään tekemään sovelluksia valmiiksi, testaamaan käytänteitä ja teknologiaa ennen kuin lisätyn todellisuuden laitteet kehittyvät ja yleistyvät. Suljettuihin työympäristöihin voidaan jo käyttää lisättyä todellisuutta, mutta sekin kärsii vielä laitteiden kankeudesta ja monimutkaisuudesta.

7 Yhteenveto

Opinnäytetyössä tutustuttiin lisätyn todellisuuden käyttöön työympäristössä Magic Leap -laitejärjestelmää hyväksi käyttäen. Kyseinen laitejärjestelmä valittiin ympäristöksi, koska se mahdollisti työn aloitushetkellä kaikkein monipuolisimman lisätyn todellisuuden.

Demon kehitys jatkuu edelleen ja se toimii kartoittajana lisätyn todellisuuden tuontiin huoltotöihin. Jo tässä vaiheessa on selvää, että käyttöön tulevaa ohjelmaa varten teknologian pitää kehittyä, jotta sovelluksesta pystytään tekemään käytettävyydeltään riittävän sulava ja kaikki toiminnalliset vaatimukset täyttävä. Kuitenkin jo nykyteknologialla on mahdollista tuoda digitaalinen informaation käyttäjälle hyödyllisessä muodossa ja tämän demon tapauksessa nopeuttaa huoltohenkilön työntekoa huomattavasti vähentämällä tarvetta hankkia informaatiota ulkoisesta lähteestä.

Lisätyn todellisuuden rajapintoja on tällä hetkellä olemassa useita ja Magic Leap:n Lumin SDK:n tapauksessa se on jo niin kattava, että lisätyn todellisuuden tekeminen on suhteellisen helppoa. Pelimoottoriin yhdistettynä Lumin SDK tarjoaa aloittelevallekin kehittäjälle mahdollisuuden tehdä monipuolisia lisätyn todellisuuden sovelluksia. Kokeneenkin kehittäjän kannattaa harkita pelimoottori-rajapintaa pelkän C-API:n käytön sijasta, mikäli pelimoottorin tuoma lisärasite suorituskyvyssä ei haittaa, voidaan sillä lyhentää kehitysaikaa ja saada monia työkaluja kehittämiseen ilman, että joudutaan tinkimään mahdollisuudesta puhtaaseen ohjelmointiin ja kolmannen osapuolen kirjastojen käyttöön.

Lisätyn todellisuuden laitekohtaiset rajoitukset ovat tämän hetken suurin rasite sulavan lisätyn todellisuuden tekemisessä. Toteutus pitää sisällään monta osiota kuten laskennan, käyttöliittymän ja sensorien käytön toteutuksen. Nämä ovat ongelmia, jotka voidaan ratkaista hyvällä suunnittelulla ohjelmakehityksessä, mutta laitekohtaiset rajoitteet ja sen sensoreiden laskennan tarkkuudet paranevat ainoastaan teknologian päivityksen mukana.

Tulevaisuus näyttää, milloin teknologia mahdollistaa niin tässä opinnäytetyössä esitellyn huolto- ja vikatietojen visualisoinnin demon kuin muidenkin sulautettujen lisätyn todellisuuden ratkaisujen toteutuksen niin saumattomasti, että integrointi on järkevää toteuttaa isommassa mittakaavassa.

Lähteet

1. Clever Simulation Entertainment. Saatavissa: https://cse.fi/fi/etusivu/.viitattu 23.10.2019.

2. InduSTAR. Saatavissa: https://www.raahenedu.fi/industar/. viitattu 23.10.2019.

3. Transtech Oy. Saatavissa: https://www.transtech.fi/etusivu. viitattu 23.10.2019.

4. Alan B. Craig. Understanding Augmented Reality. Elsevier Science & Technology; 2013

5. Hudway Glass. Saatavissa: https://hudway.co/glass. viitattu 23.10.2019.

6. Google ARCore. Saatavissa: https://arvr.google.com/arcore/. viitattu 23.10.2019.

7. Magic Leap Demos at SIGGRAPH. Saatavissa: https://www.magicleap.com/news/news/your-magic-leap-guide-to-siggraph-2019. viitattu 23.10.2019.

8. Greg Kipper and Joseph Rampolla. Augmented Reality: An Emerging Technologies Guide to AR. Elsevier Science & Technology Books; 2012

9. Virtual Reality: Introduction. Saatavissa: http://www.dsource.in/course/virtual-reality-introduction/evolution-vr/sword-damocles-head-mounted-display. viitattu 23.10.2019.

10. Inventing Interactive. Saatavissa: http://www.inventinginteractive.com/2010/03/22/myron-krueger/. viitattu 23.10.2019.

11. Microsoft HoloLens Development Edition announced. Saatavissa: https://blogs.windows.com/devices/2016/02/29/introducing-first-ever-experiences-for-the-microsoft-hololens-development-edition/. viitattu 7.11.2019.

12. Pokemon Go. Saatavissa: https://www.pokemongo.com/en-gb/. viitattu 24.10.2019.

13. Using Extended Reality to Improve Safety on Construction Sites. Saatavissa: https://www.bechtel.com/blog/innovation/november-2018/using-extended-reality-improve-safety-construction/. viitattu 23.10.2019.

14. Trimble. Saatavissa: https://mixedreality.trimble.com/. viitattu 23.10.2019.

15. IKEA Place. Saatavissa: https://www.ikea.com/au/en/apps/IKEAPlace.html. viitattu 8.11.2019.

16. Minecraft Earth. Saatavissa: https://www.minecraft.net/en-us/earth. viitattu 24.10.2019.

17. Hyper Reality. Saatavissa: http://hyper-reality.co/. viitattu 24.10.2019.

18. Magic Leap. Saatavissa: https://www.magicleap.com/. viitattu 24.10.2019.

19. Engineering.com Magi Leap One. Saatavissa: https://www.engineering.com/ARVR/ArticleID/17325/Magic-Leaps-Less-Than-Spectacular-Demo-Lessens-Anticipation-for-Release.aspx. viitattu 24.10.2019.