



Mikael Meller

Sisätilapaikannus ja lisätyn todellisuuden sovelluksen prototyyppi

Metropolia Ammattikorkeakoulu

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinöörityö

5.5.2022

Tiivistelmä

Tekijä:	Mikael Meller
Otsikko:	Sisätilapaikannus ja lisätyn todellisuuden sovelluksen prototyyppi
Sivumäärä:	37 sivua
Aika:	5.5.2022
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto ja viestintäteknikka
Ammatillinen pääaine:	Mediateknikka
Ohjaaja:	Lehtori Antti Laiho

Insinöörityön tarkoituksena oli konenäköön perustuvan sisäpaikannusteknologian testaaminen ja kehitys. Tavoitteena oli tuottaa teknologian kehittäneelle yritykselle käytännön tietoa järjestelmän toiminnasta, testata sille kehitettyä uutta Unity-liitännäistä ja rakentaa Unitylla lisättyä todellisuutta (augmented reality, AR) hyödyntävä sovellus yhteistyökumppanina olleen kauppakeskuksen käyttöön.

Projektissa rakennettiin ensin kauppakeskukseen kahdelle pienelle alueelle rajattu paikannusjärjestelmä, jonka toimivuutta testattiin navigointisovelluksella ja Unity-liitännäistä käytävällä koesovelluksella. Paikannuksen rakentamiseen ja navigointiin liittyvät ongelmakohdat raportoitiin teknologian kehittäjille, ja kun testijärjestelmä oli saatu toimivaksi, sen pohjalle alettiin kehittää AR-sovellusta.

AR-sovellusta lähdettiin kehittämään Unity-liitännäisen esimerkkiprojektin pohjalle. Kameran ohjelmalliseen käsittelyyn liittyvien teknisten haasteiden vuoksi sovellusta ei saatu toimimaan täysin luotettavasti, eikä se kyennyt käyttäjän jatkuvaan paikantamiseen. Teknisistä ongelmista huolimatta projektin tuloksena saatiin rakennettua yksinkertainen, mutta olennaisimpia toimintoja havainnollistava AR-peli. Sovellus esiteltiin kauppakeskuksen edustajille ja muille sidosryhmille projektin päätteeksi, ja se voi toimia tulevaisuudessa monipuolisemman jatkokehittelyn pohjana.

Avainsanat: sisätilapaikannus, lisätty todellisuus, konenäkö

Abstract

Author: Mikael Meller
Title: Indoor positioning and augmented reality application prototype
Number of Pages: 37 pages
Date: 5 May 2022

Degree: Bachelor of Engineering
Degree Programme: Information and Communications Technology

Professional Major: Media Technology
Supervisors: Antti Laiho, Senior Lecturer

The purpose of this thesis work was to test a proprietary indoor positioning technology and to create an augmented reality (AR) application for a shopping mall using a Unity plugin recently developed for the positioning system.

The final year project consisted of building a limited test case scenario at the mall premises. Technological problems were reported back to developers and in the end the scenario was used as the basis for the AR application.

The Unity plugin posed some issues, mainly relating to the camera handling by the plugin. These problems resulted in unreliable behaviour of the application and failure to continuously track the movements of the user. Nevertheless, a simple AR game demonstrating the essential features was developed and presented to the stakeholders. In the future this game can be used to facilitate further development.

Keywords: indoor positioning, augmented reality, machine vision

Sisällys

1	Johdanto	1
2	Sisätilapaikannuksen teknologioita	2
2.1	Satelliittipaikannus	2
2.2	Wi-Fi-tukiasemiin perustuva paikannus	4
2.3	Bluetooth-majakat	6
2.4	Magneettikenttään perustuva paikannus	7
2.5	Liikkeen tunnistamiseen perustuva paikannus	8
2.6	Visuaaliset ratkaisut	9
3	VimAI-sovellus ja sen käyttö	9
3.1	Kartoitussovelluksen käyttö	10
3.2	Navigointisovelluksen käyttö	13
3.3	Edut ja haitat	14
4	VimAI ja lisätty todellisuus	15
4.1	Unity	16
4.2	ARCore	17
4.3	VimPlugin	18
5	Kauppakeskus Tripla	19
5.1	Kartoitus	20
5.2	VimPluginin testaus	24
5.3	AR-Testit	26
5.4	AR-sovellus	28
6	Yhteenveto	31
	Lähteet	32

1 Johdanto

Älypuhelimien yleistyessä yhä harvempi enää turvautuu perinteiseen karttaan maailmalla suunnistaessaan. Puhelimien GPS-paikannussovellukset eivät ainoastaan ohjaa käyttäjää perille kohteeseen, vaan ne tarjoavat navigointiavun lisäksi valtavan määrän ajantasaista tietoa ympäristöstä. Lisättyä todellisuutta (augmented reality, AR) hyödyntäen älylaite voi myös liittää puhelimen kameran esittämään videokuvaan hyödyllistä, opettavaa ja viihdyttävää mediasisältöä. Muinaisrauniot voivat herätä eloon puhelimen näyttöruudulla ja metsässä vilistää mielikuvituksellisia pelihahmoja.

GPS-paikannuksen epätarkkuus ja alttius fyysisten esteiden aiheuttamille häiriöille rajoittaa kuitenkin sen käytön ulkotiloihin. Tarkempaan sisätilapaikannukseen on tarjolla erilaisia teknisiä ratkaisuja, joista yksi on suomalaisen ohjelmistoyritys VimAI:n kehittämä, konenäköön perustuva teknologia. Älypuhelimien kameraa hyödyntämällä VimAI:n mobiilisovellus kykenee paikantamaan käyttäjän vajaan metrin tarkkuudella ja selvittämään, mihin suuntaan tämä katsoo. Jälkimmäinen ominaisuus erottaa VimAI:n useimmista kilpailijoistaan ja mahdollistaa hyvinkin monipuolisten AR-sisältöjen kehittämisen.

Insinööriyön tarkoituksena on testata VimAI:n teknologiaa ja Unity-pelimoottorille kehitettyä liitännäistä helsinkiläisessä kauppakeskus Triplassa. Lisäksi tavoitteena on kehittää AR-teknologiaa hyödyntävä pilottisovellus, joka toimisi pohjana jatkokehitykselle ja mahdolliselle kaupalliselle yhteistyölle. Insinööriyö toteutetaan osana Metropolian Digi-Salama-projektia ja yhteistyössä VimAI:n, PasilaHUB-hankkeen ja kauppakeskus Triplan kanssa.

2 Sisätilapaikannuksen teknologioita

Sisätilapaikannuksen toteuttamiseen on kehitetty useita erilaisia teknologisia ratkaisuja, jotka Kunhoth ym. (1, s. 2–5) ovat jaotelleet seuraavasti:

- radioaaltoihin perustuvat teknologiat (satelliittipaikannus, Wi-Fi, Bluetooth)
- magneettikenttään perustuvat teknologiat
- liikkeen tunnistamiseen perustuvat teknologiat
- visuaaliseen tunnistukseen perustuvat teknologiat.

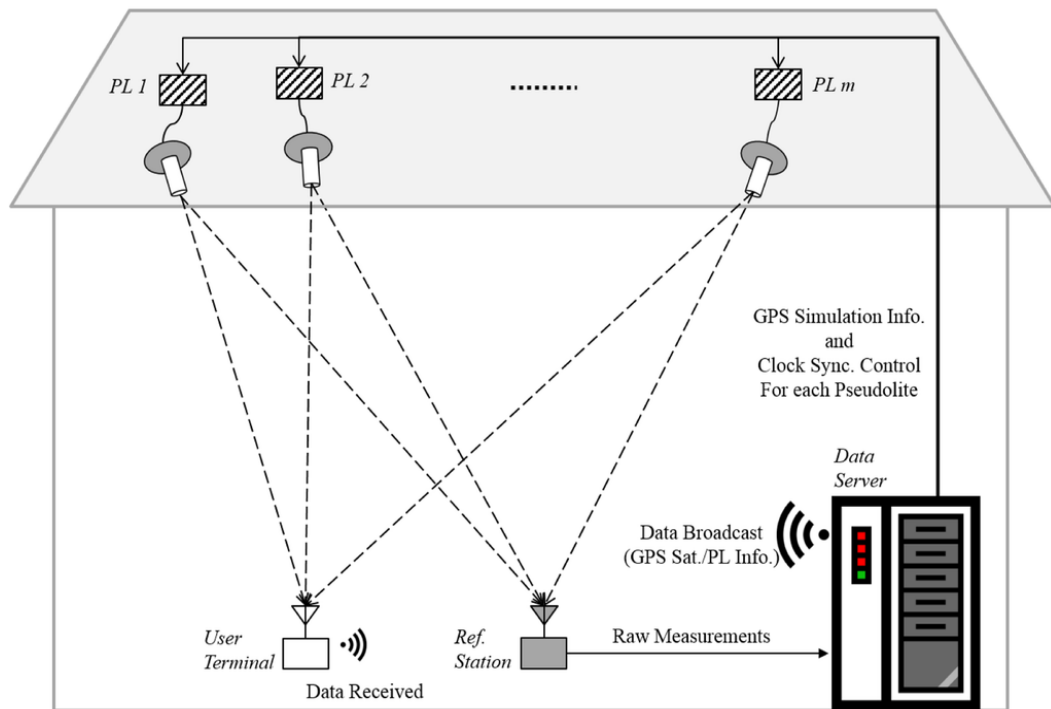
Kaikilla käytössä olevilla teknologioilla on omat vahvuutensa ja heikkoutensa, jotka liittyvät pääasiassa paikannuksen luotettavuuteen, vaadittaviin laiteinvestointeihin ja käyttöönoton helppouteen. Toistaiseksi yksikään lähestymistapa ei ole noussut selvästi muiden yläpuolelle ja niitä kaikkia kehitetään edelleen.

2.1 Satelliittipaikannus

Ulkotiloissa tapahtuvan navigoinnin *de facto* -standardi, Yhdysvaltain armeijan kehittämä satelliittipohjainen GPS (global navigation system) soveltuu huonosti sisätiloissa käytettäväksi. Vaikka järjestelmä kykenee teoreettisesti saavuttamaan useaa eri taajuutta hyödyntävillä erikoisvastaanottimilla jopa kymmenien senttimetrien tarkkuuden, käytännössä älypuhelimilla päästään National Institute of Standards and Technology -viraston (2) mukaan keskimäärin 4,9 metrin tarkkuuteen. Lisäksi fyysiset esteet, kuten rakennusten seinät ja katot, heikentävät GPS-signaalin luotettavuutta sisätiloissa (3).

Sisätilapaikannusta varten on kehitetty vastaavaa periaatetta hyödyntäviä teknologioita, kuten pseudolite ("pseudo-satellite"), jossa taivaalla olevan satelliitin sijaan käytetään rakennuksen kattoon tai ulkoisiin antenneihin

sijoitettuja "valesatelliitteja" (4). Kuvassa 1 voidaan nähdä esimerkki pseudolite-järjestelmästä, jossa kattoon asennettujen lähettimien signaalia käytetään paikantamiseen vertaamalla liikkuvan vastaanottimen signaalia kiinteään referenssivastaanottimeen.



Kuva 1. Pseudoliten toimintaperiaate (2, s. 6107).

Pseudolite ja muut vastaavat ratkaisut kykenevät varsin tarkkaan paikannukseen, ja ne voivat hyödyntää älypuhelimissa valmiina olevaa GPS-teknologiaa, mutta niiden rasitteena ovat lähettimiin liittyvät laitteistovaatimukset ja käyttöönottokulut. GPS ja muut radioaaltoihin perustuvat ratkaisut kärsivät myös erilaisista signaalihäiriöistä, kuten monitie-etenemisestä (multipath propagation).

Weillin (5, s. 59–60) mukaan monitie-etenemistä tapahtuu, kun radiosignaali saapuu vastaanottavaan laitteeseen useista eri suunnista esimerkiksi suurista pinnoista heijastuen tai ilmakehän aiheuttaman diffraktion takia. Eri reittejä kulkevat signaalit saavuttavat vastaanottimen eri aikaan, mikä aiheuttaa vaihevirhettä ja interferenssiä ja heikentää järjestelmän tarkkuutta.

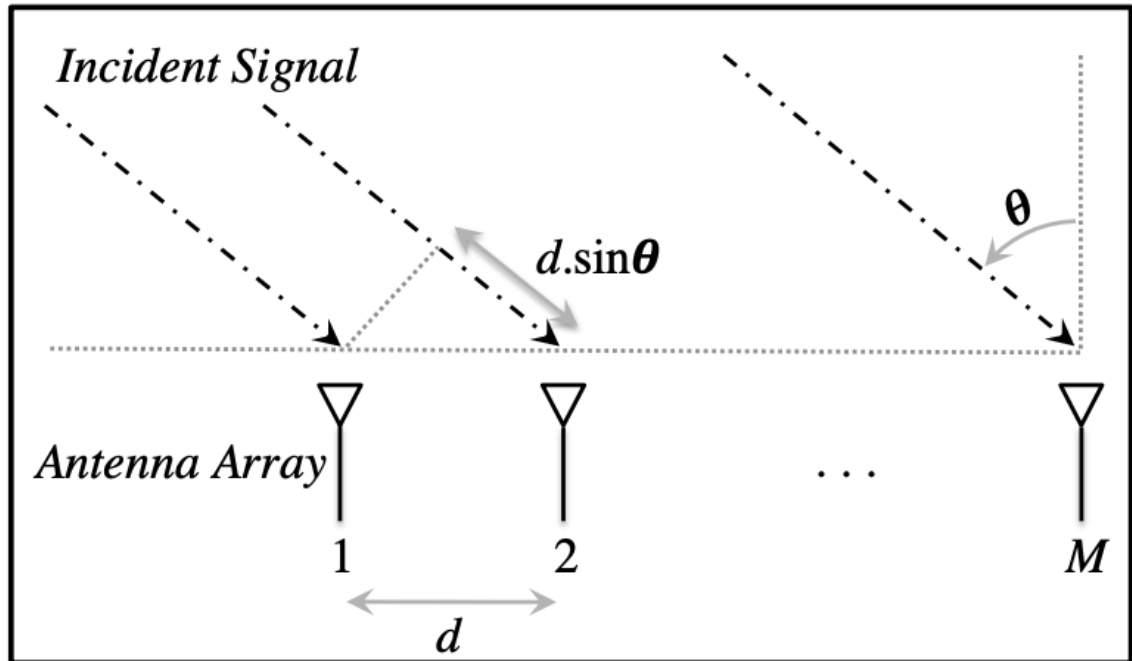
2.2 Wi-Fi-tukiasemiin perustuva paikannus

Langattomien Wi-Fi-tukiasemien valjastaminen paikannukseen on herättänyt paljon kiinnostusta. Teknologian etu on se, että tarvittava infrastruktuuri on jo hyvin pitkälti olemassa tukiasemien yleisyyden ansiosta. Varsinkin kaupunkiympäristössä alkaa olla vaikeaa löytää paikkaa, joka ei olisi yhden tai useamman Wi-Fi-tukiaseman vaikutusalueella.

Kotaru ym. (6, s. 270–271) jakavat Wi-Fi-paikannusteknologiat (Wi-Fi Positioning system, WPS) karkeasti neljään tyyppiin:

- signaalin voimakkuutta analysoivat ratkaisut (RSSI, received signal strength indicator)
- uniikkeihin, vektoreista rakennettuihin ”sormenjälkiin” perustuvat ratkaisut
- signaalin tulokulmaan perustuvat ratkaisut (AoA, angle of arrival)
- signaalin kulkuun kuluvan ajan laskemiseen perustuvat ratkaisut (ToA, time of arrival).

Näiden teknologioiden väliset erot näkyvät mm. luotettavuudessa, tarkkuudessa ja laitteistovaatimuksissa. Viime aikoina suurimpia edistysaskelia on ottanut tulokulmaan perustuva teknologia. Kuvassa 2 havainnollistetaan AoA-periaatetta, jossa signaalin tulokulman avulla lasketaan signaalin kulkema matka eri vastaanottimiin. Kehittyneempien algoritmien avulla tulokulmaan perustuva paikannus on saatu tarkemmaksi ja vähemmän riippuvaiseksi tukiasemien teknisestä laadusta.



Kuva 2. AoA-navigoinnin periaate (6, s. 270).

Wi-Fi-paikannus on nostanut esiin yksityisyyteen ja tietoturvaan liittyviä kysymyksiä. Google (5) on ehdottanut järjestelyä, jossa tukiasemien haltijat voisivat kieltää niiden käytön paikannustarkoitukseen liittämällä aseman SSID-tunnisteeseen tekstin ”_nomap”, ja mm. Mozilla on jo ottanut sen käyttöön omassa paikannuspalvelussaan. Toistaiseksi kuitenkin alan toimijoiden kesken ei ole saavutettu yksimielisyyttä yhteisestä käytännöstä.

2.3 Bluetooth-majakat

Bluetooth-majakoihin perustuva paikannus herätti kiinnostusta 2010-luvun alkupuolella, ja mm. Apple panosti paljon teknologiaan tuomalla markkinoille oman iBeacon-protokollansa vuonna 2013 (9).

Bluetoothin vahvuus on järjestelmän käyttöönoton helppous ja edullisuus. Markkinoilla on useita kustannustehokkaita pakettiratkaisuja, joiden asentaminen ja käyttö ei vaadi syvällistä teknistä asiantuntemusta, ja joiden paikannustarkkuus on suhteellisen hyvä. Bluetooth-paikannukseen perustuvia opastuksia ja AR-sovelluksia on toteutettu mm. museoissa (9; 10). Kuvassa 3 on esimerkkitoetus Bluetooth-paikannuksesta taidemuseossa; kävijöiden älypuhelimet havaitsevat seinille sijoitellut Bluetooth-majakat ja määrittelevät niiden avulla sijaintinsa.



Kuva 3. Bluetooth-paikannus käytössä taidemuseossa (10, s. 5).

Bluetoothiin perustuvien järjestelmien varjopuolena on rajallinen skaalautuvuus. Bluetooth-signaali toimii luotettavasti noin 10 metrin säteellä, joten jo keskikokoisen toimistorakennuksen kattamiseen tarvitaan useita kymmeniä majakoita, mikä nostaa järjestelmän hintaa ja monimutkaistaa sen asennusta ja ylläpitoa.

2.4 Magneettikenttään perustuva paikannus

Maapallon magneettikenttää hyödyntävää paikannusta on tutkittu jo 1970-luvulta lähtien Yhdysvaltain laivaston (10) toimesta. Ulkotiloissa magneettinen paikannus kykenee merkittävään tarkkuuteen, mutta sisätiloissa teräsrakenteet ja -kalusteet häiritsevät sitä. Kuvassa 4 nähdään, miten metalliset rakenteet vaikuttavat magneettikenttään. Tämän ongelman ratkaisua on kehittänyt mm. suomalainen yritys IndoorAtlas, jonka teknologia pyrkii hyödyntämään näitä häiriöitä paikannuksessa. Metallirakenteiden aiheuttamia yksilöllisiä muutoksia analysoimalla sovellus kykenee rakentamaan rakennuksesta virtuaalisen mallin, jonka avulla käyttäjä pystytään paikantamaan jopa 10 senttimetrin tarkkuudella.



Kuva 4. New Yorkin Time Warner Centerin magneettikenttä visualisoituna. Suuren metallisen patsaan voimakas vaikutus magneettikenttään näkyy punaisena värinä. (12.)

Magneettinen paikannus ei vaadi toimiakseen internetyhteyttä, koska se hyödyntää puhelimen omaa kompassia. Varjopuolena teknologia on herkkä ympäristössä tapahtuville muutoksille. Liikkuvat hissit tai siirrettävät metalliset kalusteet saattavat sotkea paikannusta, ja pahimmillaan suuret muutokset esimerkiksi sisustuksessa voivat pakottaa rakentamaan virtuaalisen mallin kokonaan uudestaan. (13.)

2.5 Liikkeen tunnistamiseen perustuva paikannus

Älylaitteiden kiihtyvyyssantureita hyödynnetään laajasti käyttäjän liikkeen mittaamiseen mm. erilaisissa AR-peleissä ja liikuntasovelluksissa. Suurempien tilojen navigointiin liiketunnistus (dead reckoning) soveltuu huonosti, koska pienet virheet paikannuksessa kertautuvat nopeasti (15, s. 1). Liiketunnistus myös analysoi sananmukaisesti vain liikettä eli sijainnin muutosta, joten se vaatii jonkin muun teknologian apua käyttäjän absoluuttisen alkusijainnin selvittämiseen.

Liiketunnistus ei sellaisenaan sovellu sisätilapaikannukseen, mutta useat muihin teknologioihin perustuvat ratkaisut hyödyntävät sitä sekundäärisenä paikannusmuotona. Mikäli esim. Wi-Fi-paikannus kadottaa signaalin katvealueella, liiketunnistin voi pitää käyttäjän kartalla, kunnes tukiaseman signaali taas löytyy.

2.6 Visuaaliset ratkaisut

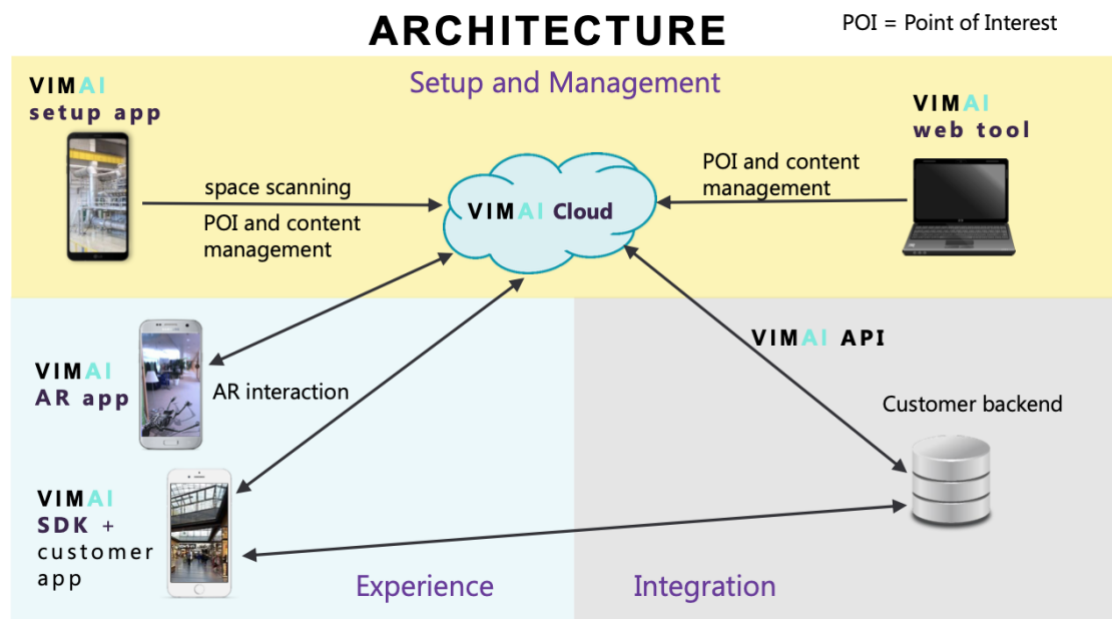
Viime vuosina erilaiset konenäköön perustuvat paikannusteknologiat ovat herättäneet kiinnostusta. Visuaalisessa navigoinnissa älypuhelimien kameralla kerätään ympäristöstä visuaalista dataa, jonka avulla käyttäjä pystytään paikallistamaan (15).

Yksinkertaisimmin visuaalinen paikannus voidaan toteuttaa käyttämällä ympäristöön sijoitettuja kiintopisteitä ("marker"), jotka navigaatio-sovellus tunnistaa ja joiden avulla se arvioi sijainnin. Kehittyneemmät teknologiat eivät tarvitse keinotekoisia kiintopisteitä, vaan ne pystyvät tunnistamaan ympäristön piirteitä valokuvasta ja vertaamaan sitä ympäristöstä aiemmin tehtyyn kolmiulotteiseen malliin. Tällaista teknologiaa käyttää myös VimAI.

3 VimAI-sovellus ja sen käyttö

VimAI on Aalto-yliopistossa vuonna 2017 perustettu yritys, joka on kehittänyt konenäköä hyödyntävän paikannusjärjestelmän (16). Järjestelmä perustuu älypuhelimien kameralla tapahtuvaan datan keräämiseen eli kartoittamiseen. Tämän kuvadatan pohjalta rakennetaan kartoitettavasta tilasta 3D-malli, jonka avulla loppukäyttäjä pystytään paikallistamaan. Kaikki datan käsittely tapahtuu

VimAI:n pilvipalvelimella (kuva 5).

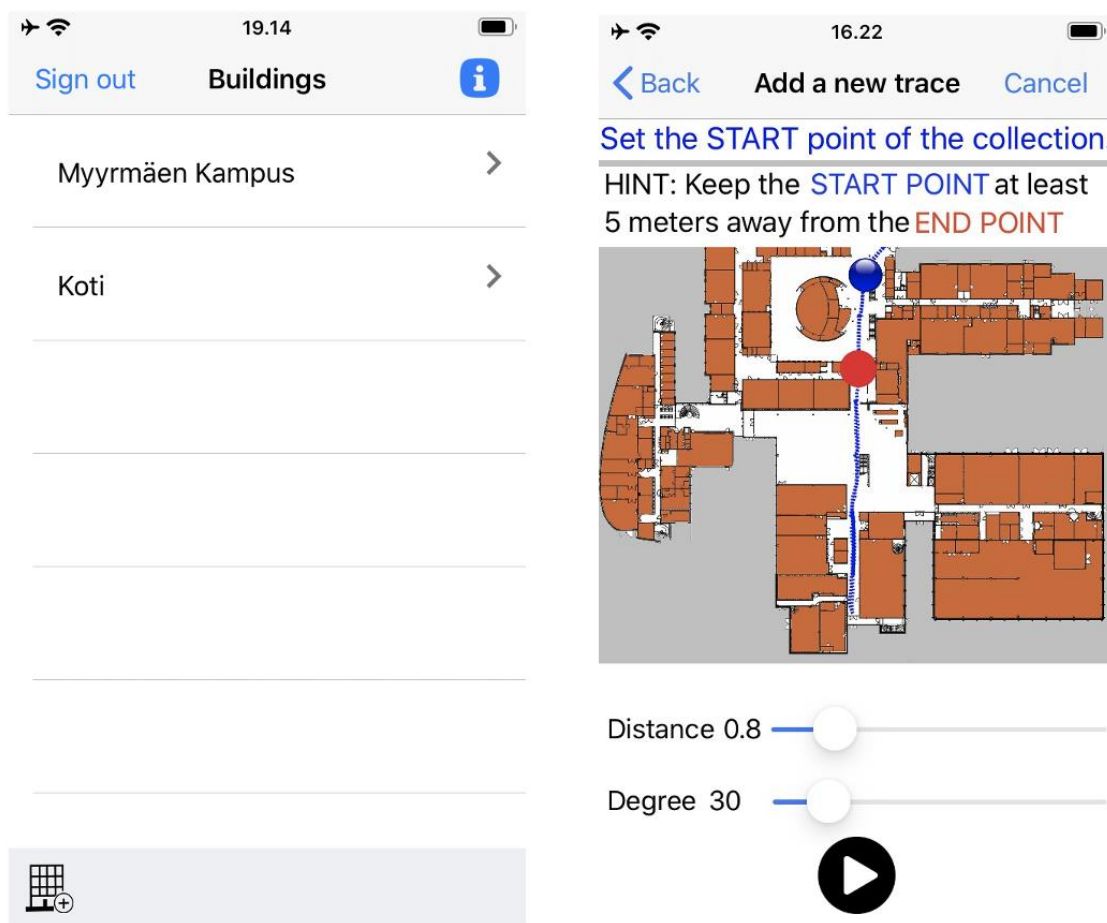


Kuva 5. VimAI-sovelluksen arkkitehtuuri. Keskiössä on VimAI:n pilvipalvelin, jossa kuvadataa säilytetään ja johon sovellukset ottavat yhteyden. (16.)

Kartoitusdataa voidaan käyttää joko VimAI:n omalla navigointisovelluksella tai asiakkaalle yksilöllisesti kehitetyllä sovelluksella, johon voidaan myös liittää asiakkaan itse kehittämää AR-sisältöä tai API (application programming interface) -rajapinnan kautta ladattavaa dynaamista sisältöä.

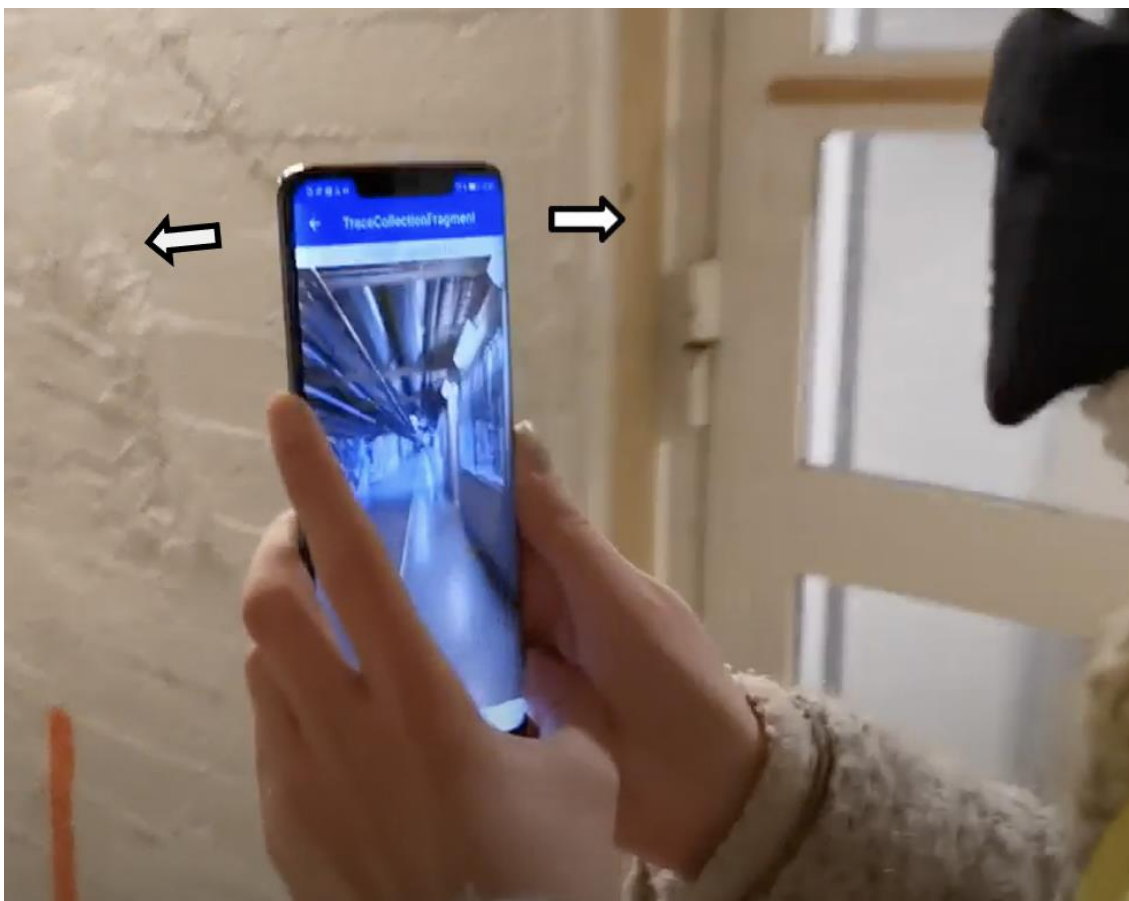
3.1 Kartoitussovelluksen käyttö

Sisätilojen kartoittaminen tapahtuu yksinkertaisen sovelluksen avulla. Aluksi käyttäjä luo sovellukseen uuden rakennuksen ja lataa kartoitettavan alueen pohjapiirustuksen VimAI:n palvelimelle (kuva 6). Tämän jälkeen käyttäjä merkitsee pohjapiirustukseen aloituspisteensä mahdollisimman tarkasti ja aloittaa kartoitusprosessin.



Kuva 6. VimAI:n kartoitussovellus, vasemmalla lista rakennuksista ja oikealla Metropolian Myyrmäen kampuksen kartoitusnäky. Sininen katkoviiva näyttää jo kartoitetun reitin. (17.)

Kartoittaja pitää puhelintaan pystyasennossa katseensa korkeudella ja kävelee kartoitettavan alueen läpi käännettäessä samalla kameraansa vaakasuunnassa rauhallisesti puolelta toiselle, kuten kuvasta 7 voi nähdä. Sovellus ottaa kuvia automaattisesti kameran liikuessa, ja mikäli käyttäjä pysähtyy, myös sovellus lopettaa kuvien ottamisen ja jatkaa sitä, kun käyttäjä taas liikkuu. Ulkoisten häiriöiden, kuten ohikulkijoiden, ilmaantuessa kartoitusta ei siis tarvitse aloittaa alusta uudestaan, vaan käyttäjä voi rauhassa odottaa häiriötilanteen menevän ohi ja jatkaa kartoitusta saumattomasti.



Kuva 7. VimAI-kartoitussovelluksen käyttö. Puhelinta liikutellaan rauhallisesti sivuttaissuunnassa noin 30 asteen kulmaan asti.

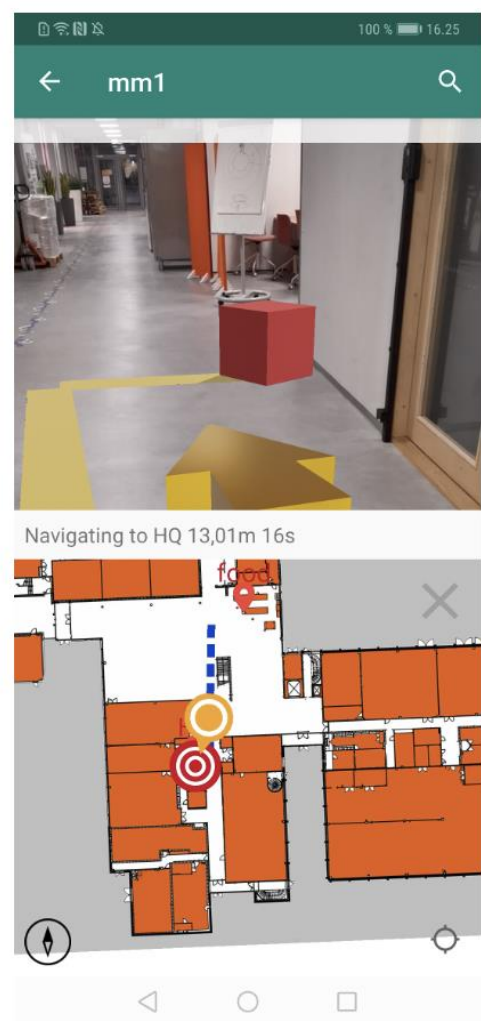
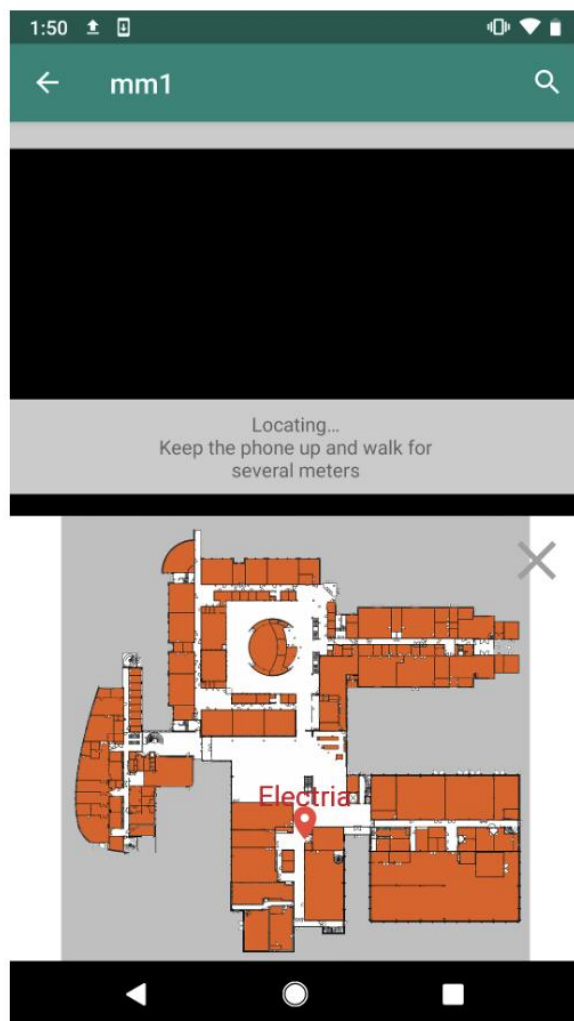
Kun käyttäjä saavuttaa kartoitettavan reitin lopun, hän pysäyttää kartoituksen ja merkitsee lopetuspisteen pohjapiirustukseen. Isojen tilojen kartoitus suositellaan tehtäväksi pienemmissä osissa. Kartoitus tehdään reitin molempiin suuntiin, ja sama reitti on mahdollista käydä useampaan kertaan läpi, jolloin kerätty data lisääntyy kumulatiivisesti.

Kun haluttu määrä dataa on kerätty, se voidaan ladata VimAI:n pilvipalvelimelle. Tämän jälkeen voidaan aloittaa opettaminen (training), jossa VimAI:n algoritmi rakentaa datasta mallin. Tämä vaihe kestää kerätyn datan määrästä riippuen noin viidestä minuutista reiluun tuntiin. Reitin opettamisen jälkeenkin sen voi tarvittaessa vielä kartoittaa uudestaan, eikä aikaisemmin kerätty data pyyhkiydy pois automaattisesti.

Valmiiseen karttaan voi sovelluksen avulla lisätä rajattoman määrän kiintopisteitä (POI, point of interest), joihin loppukäyttäjä voi navigoida ja jotka näkyvät hänen puhelimensa kameranäkymässä kolmiulotteisina AR-hahmoina.

3.2 Navigointisovelluksen käyttö

Loppukäyttäjä käyttää VimAI:ta erityisellä navigointisovelluksella, jonka ulkonäköä ja ominaisuuksia voidaan räätälöidä kohteen mukaan (18). Kuvassa 8 sovellus on käytössä Metropolian Myyrmäen kampuksella. Navigointisovellus tunnistaa ensin GPS:n avulla käyttäjän karkean sijainnin ja osaa sen perusteella avata oikean rakennuksen mallin. Malli voidaan valita myös manuaalisesti.



Kuva 8. VimAI:n navigointisovellus käytössä Metropolian Myyrmäen kampuksella. Oikealla POI-piste näkyy kameran kuvassa punaisena kuutiona. (18.)

Paikantaminen tapahtuu pitämällä kameraa pystyasennossa ja kävelemällä. Sovellus ottaa valokuvia automaattisesti ja muutaman metrin kävelyn jälkeen tunnistaa ympäristön ja sijoittaa käyttäjän siihen. Tämän jälkeen sovellus pystyy seuraamaan käyttäjän liikettä 3D-mallissa. Sovellus kykenee myös hyödyntämään väliaikaisesti puhelimen kiihtyvyyssensoreita, mikäli se kadottaa sijaintinsa.

Käyttäjä voi suunnistaa eri kohteisiin joko valitsemalla jonkin etukäteen määritellyn POI-pisteen luettelosta tai vapaasti kartasta haluamaansa kohtaa painamalla. Pelkän navigoinnin lisäksi VimAI:n paikannusta voi hyödyntää monipuolisten AR-sovellusten tekoon käyttäen suosittua Unity-kehitysalustaa.

3.3 Edut ja ongelmat

VimAI ei vaadi palvelun käyttöönottajalta merkittäviä laiteinvestointeja: kartoittamiseen ei tarvita muuta kuin Googlen ARCore-alustaa tukeva älypuhelin (19). VimAI pystyy paikantamaan käyttäjän noin yhden metrin tarkkuudella, ja monista muista markkinoilla olevista tekniikoista poiketen se kykenee määrittelemään myös käyttäjän katseen suunnan, mikä on erityisen tärkeää AR-sisältöä ajatellen.

Varjopuolena on teknologian riippuvuus käyttäjien toiminnasta. Kartoitusta vaatii jonkin verran huolellisuutta toimiakseen luotettavasti, ja hankalat kohteet, kuten avarat ja epäsäännöllisen muotoiset tilat, saattavat vaatia useita päällekkäisiä kuvausajoja ja testausta. Tätä ongelmaa voi tulevaisuudessa yrittää ratkaista esimerkiksi robotiikan avulla. Ihmisen sijaan yksinkertainen robotti voisi huolehtia kartoituksesta täsmällisesti ja luotettavasti kulkemalla rakennuksen käytävillä.

Myös loppukäyttäjän toiminnalla on merkitystä navigoinnissa. Käyttäjä ei saa liikkua liian nopeasti tai sovellus ei ehdi kerätä tarpeeksi kuvadataa ympäristöstä. Käyttäjän täytyy myös pitää puhelintaan pystyasennossa katselukorkeudella, mikä varsinkin julkisissa tiloissa saattaa tuntua kiusalliselta ja herättää huomiota.

VimAI, kuten muutkin visuaaliseen tunnistukseen perustuvat ratkaisut, on altis ympäristön muutoksille. Jos esimerkiksi kauppakeskuksessa liikkeiden näyteikkunat muuttuvat, järjestelmä ei välttämättä enää tunnista ympäristöä luotettavasti. Kiihtyvyyssantureiden käyttö sekundäärisenä paikannusmuotona auttaa ongelman ratkaisussa jonkin verran, mutta kovin suuret muutokset käyttöympäristössä voivat vaatia uuden kartoituksen. Myös suuret ihmismäärät häiritsevät navigointia, mutta käytännön testeissä esimerkiksi kauppakeskuksen normaalit asiakasvirrat eivät tuottaneet sovellukselle mainittavia ongelmia.

4 VimAI-sovellus ja lisätty todellisuus

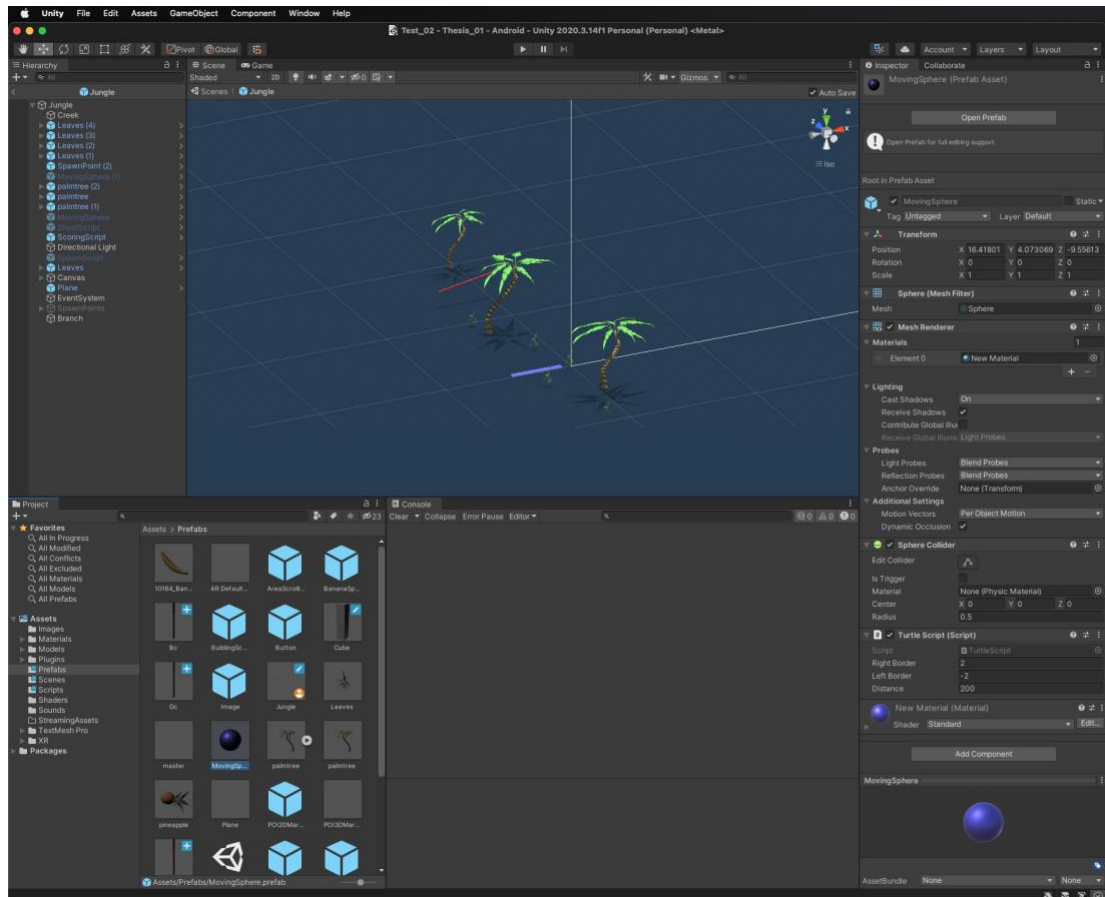
Kuten luvussa 3.3 mainittiin, VimAI soveltuu erityisen hyvin AR-sovelluksiin, koska sillä pystytään määrittelemään käyttäjän rintamasuunta (AR Pose). Tämä helpottaa monipuolisten AR-ympäristöjen rakentamista, koska kehittäjä voi luottaa siihen, että luotu AR-maailma on aina paitsi oikeassa paikassa, myös oikein päin. VimAI:lla voidaan kehittää AR-sisältöä pieniin ja rajoitettuihin tiloihin, kuten käytäviin, koska AR-objektit saadaan huolellisella suunnittelulla pysymään aina tilan fyysisten rajojen sisällä.

Tieto käyttäjän rintamasuunnasta tarjoaa myös joustavuutta sovelluksen arkkitehtuurin suunnitteluun ja toteutukseen, koska kaikille peliobjekteille ei tarvitse määritellä absoluuttista sijaintia VimAI:lla, vaan niitä voidaan sijoitella suhteessa toisiinsa. Näin AR-maailman rakentaminen voidaan tehdä sitä varten suunnitelluilla tehokkailla työkaluilla, kuten Unitylla.

4.1 Unity-pelimoottori

Unity on vuonna 2005 julkaistu pelimoottori, joka on graafisen käyttöliittymänsä ja laajan yhteensopivuutensa ansiosta erityisesti aloittelevien ja indie-pelikehittäjien suosiossa (20). Unity julkaistiin alun perin Mac OS X -käyttöjärjestelmälle, mutta se on ajan mittaan kehittynyt tukemaan kaikkia merkittävimpiä alustoja. Unity oli ensimmäisiä iPhonea tukevia kehitysalustoja ja sen ansiosta pitkään mobiilipelikehityksen *de facto* -standardi. Unity on edelleen laajalti käytetty mobiilikehittäjien keskuudessa, ja monet menestyksekkäät pelit, kuten Pokemon GO! ja Call Of Duty Mobile, on tehty Unitylla.

Kuvassa 9 näkyy Unityn graafinen käyttöliittymä, jonka keskellä on pelimaailman kolmiulotteinen näkymä. Sen alapuolella näkyy projektikansio, johon käyttäjä voi tuoda uutta sisältöä (assets) kuten C#-ohjelmointikielellä laadittuja skriptejä tai audiovisuaalista sisältöä. Vuosien varrella Unityn ympärille on kehittynyt aktiivinen yhteisö ja ekosysteemi, jossa käyttäjät voivat jakaa kehittämiään sisältöjä mm. Unityn omassa Asset Store -palvelussa.



Kuva 9. Unityn graafinen käyttöliittymä (21).

Viime vuosina Unityn käyttö on laajentunut myös pelimaailman ulkopuolelle, sen VR- ja AR-ominaisuuksia on hyödynnetty mm. autoteollisuudessa muotoilijoiden työkaluna (22). Vuonna 2018 Unity julkaisi AR Foundation -nimisen, ARCorea tukevan kehitysympäristön, joka on yksinkertaistanut huomattavasti AR-sisällön kehittämistä ja kasvattanut Unityn suosiota AR-kehitysalustana.

4.2 ARCore-kehitysalusta

ARCore on Googlen vuonna 2017 julkaisema AR-kehitysalusta. Alun perin ARCore kehitettiin Googlen omalle Android-käyttöjärjestelmälle, mutta myöhemmin siihen lisättiin tuki myös Applen iOS:lle. ARCore on nopeasti noussut laajalti käytetyksi työkaluksi, ja myös VimAI hyödyntää sitä.

ARCore tarjoaa kehittäjille dataa laitteen suhteesta ulkomaailmaan kolmella tavalla:

- liikkeen seuranta (Motion tracking): älylaitteen liikkeen ja suunnan tunnistaminen kiihtyvyyssensorien ja kameran avulla
- ympäristön havainnointi (Environmental understanding): kyky tunnistaa reaali maailmasta tasaisia pintoja ja sijoittaa AR-objekteja suhteessa niihin
- valaisun arviointi (Light estimation): kyky havaita muutoksia ympäristön valaistuksessa ja toisintaa niitä AR-maailmassa.

Kaikki markkinoilla olevat älypuhelimet eivät vielä tue ARCorea, mutta määrä kasvaa koko ajan (19).

4.3 VimPlugin-liitännäinen

Kesällä 2021 VimAI julkaisi VimPlugin-liitännäisen, jonka avulla Unitylla voidaan käyttää VimAI-sovelluksen tarjoamaa paikannustietoa. Samassa yhteydessä julkaistiin myös ARTEST3-esimerkkiprojekti, jossa liitännäisen käyttöä esitellään yksinkertaisen mobiilisovelluksen avulla.

VimPlugin käyttää VimAI:n omaa NativeSDK-ohjelmistokehityspakettia. NativeSDK on ohjelmoitu C++-kielellä, ja se käyttää OpenCV- ja Eigen3-kirjastoja käyttäjän sijainnin määrittelyyn. Liitännäisen toiminta Unityssa on samankaltaista kuin VimAI-sovelluksessa: ensin se vastaanottaa REST-rajapinnan kautta VimAI:n palvelimelta listan rakennuksista ja käyttäjä valitsee oikean rakennuksen tai sovellus tekee sen automaattisesti. Tämän jälkeen VimPlugin on valmis vastaanottamaan kuvia puhelimen kameralta. Liitännäinen ei käytä kameraa itse, vaan ainoastaan välittää kuvadatan VimAI:n pilvipalvelimelle. Paikallistettuaan käyttäjän palvelin palauttaa tiedot sijainnista ja rintamasuunnasta (AR Pose).

Unityssa sijaintitietoihin pääsee käsiksi C#-skripteillä VimPluginin NativeTracker-luokan julkisten funktioiden kautta. Näistä funktioista projektin kannalta olennaisimmat ovat GetCurrentNativeLocation ja NativePositionToARPose (ks. esimerkkikoodi 1).

```
GetCurrentNativeLocation(long photoTakenMillis, Pose pose_photo_taken,
long currentTimeMillis, Pose pose_now, double[] intrinsics, string
path_to_image = "")
```

```
NativePositionToARPose(double[] nativeUserPosition, Vector3
currentUserARPose, Vector3 nativePOIPosition)
```

Esimerkkikoodi 1. GetCurrentNativeLocation- ja NativePositionToARPose-funktiot.

GetCurrentNativeLocation palauttaa käyttäjän sijaintikoordinaatit, jotka voidaan konvertoida Unityn Vector2-, Vector3- ja eulerAngles-muotoihin.

NativePositionToARPose-funktiota käytetään POI-pisteiden visualisointiin. Sen parametrit ovat käyttäjän sijainti ja rintamasuunta sekä POI-pisteen sijainti.

5 Kauppakeskus Triplan kartoitus ja AR-sovellus

Insinööriyöprojektin tarkoituksena oli testata VimAI:n sovellusta ja Unity-liitännäistä kauppakeskus Triplan tiloissa Pasilassa ja kehittää Unitylla yksinkertainen AR-sovellus, joka voisi toimia pohjana tuleville kehityshankkeille. Tripla on mukana PasilaHUB-hankkeessa, joka kehittää uusia digitaalisia palveluita Pasilan alueen asukkaille ja yrityksille.

Projektin aluksi järjestettiin kesäkuussa 2021 tapaaminen paikan päällä Triplan edustajan kanssa, joka esitteli kauppakeskuksen tiloja ja kertoi ajatuksiaan mahdollisesta sovelluksesta. Triplan tulevaisuuden toiveena oli sovellus, joka auttaisi asiakkaita suunnistamaan kauppakeskuksen tiloissa ja jota kauppakeskuksen yrittäjät voisivat hyödyntää tiedottamisessa. Ravintolat voisivat esimerkiksi esitellä ruokalistojaan ja kaupat kertoa erikoistarjousistaan virtuaalisilla mainostauluilla. Asiakaskokemusta voisi myös kehittää

pelillistämisen kautta: käyttäjät voisivat esimerkiksi kerätä Triplan käytäviltä bonuspisteitä, joilla he saisivat etuja eri liikkeistä.

Näin laajan sovelluksen kehittäminen ei projektin puitteissa ollut mahdollista, mutta tavoitteeksi asetettiin vaadittavien perustoiminnallisuuden testaaminen ja toimivan prototyypin esittely. Tarkoituksena oli luoda luotettavasti toimiva navigointi rajatulle alueelle ja AR-sovellus, joka esittelisi VimAI:n teknologian mahdollisuuksia ja toimisi pohjana jatkokehittelylle. Projektin pituudeksi määriteltiin kaksi kuukautta, ja tulokset suunniteltiin esiteltäväksi elokuussa 2021.

5.1 Kauppakeskuksen kartoitus

Kauppakeskus antoi projektiryhmälle suhteellisen vapaat kädet testauspaikan valintaan. Pääkriteereinä olivat tilan muoto, koko ja asiakasliikenteen määrä. Hyvin avarat ja epäsäännölliset tilat olisivat olleet haastavia VimAI:n tarkkuuden kannalta, eivätkä välttämättä olisi esitelleet parhaalla tavalla teknologian toimivuutta rajatussa tilassa. Suuret ihmismäärät taas olisivat vaikeuttaneet kartoitus- ja testausprosessia, koska liikkuvat kohteet häiritsevät kartoitusta ja kuvaustoiminta olisi saattanut herättää asiakkaiden keskuudessa kielteistä huomiota.

Kovin laajaa kartoitusta ei projektin puitteissa ollut mahdollista tehdä, koska suuren alueen kartoituksessa datan kerääminen, algoritmin kouluttaminen ja testaaminen vievät runsaasti aikaa. Laaja kartoitus ei myöskään ollut mielekästä projektin tavoitteen kannalta: tarkoituksena ei ollut saada aikaan valmista tuotetta, vaan toimiva ja kehityskelpoinen prototyyppi.

Lopulta päädyttiin kahteen lokaatioon: toisen kerroksen ravintola-alueen läheisyydessä sijaitsevaan L-muotoiseen käytävään ja P4-kerroksen pysäköintihallien yhteydessä olevaan ”Down Under” -elämyskeskukseen (kuva 10). Valitut kaksi aluetta ovat toisiinsa yhteydessä hissillä, ja suunniteltava sovellus voisi toimia myös opasteena toisesta kerroksesta elämyskeskuksen

palveluihin. Lisäksi elämyskeskus, jossa toimii mm. suosittu lasten sisäleikkipuisto, koettin otolliseksi ympäristöksi pelillisen AR-sovelluksen kokeilulle.

Testauksen kannalta oli suotuisaa, että valitut alueet olivat ympäristöinä sängen erilaisia. Toisen kerroksen käytävässä on paljon liikkeitä ja vaihtelevia yksityiskohtia, kuten näyteikkunoita ja kylttejä, kun taas P4-kerroksen käytävä on yksinkertainen ja paljas tila, jossa on hyvin vähän visuaalisia kiintopisteitä. Myös tilojen valaistus ja mittasuhteet olivat erilaiset. Näiden kahden tilan avulla saatiin tietoa VimAI:n algoritmin toimimisesta erilaisissa ympäristöissä.

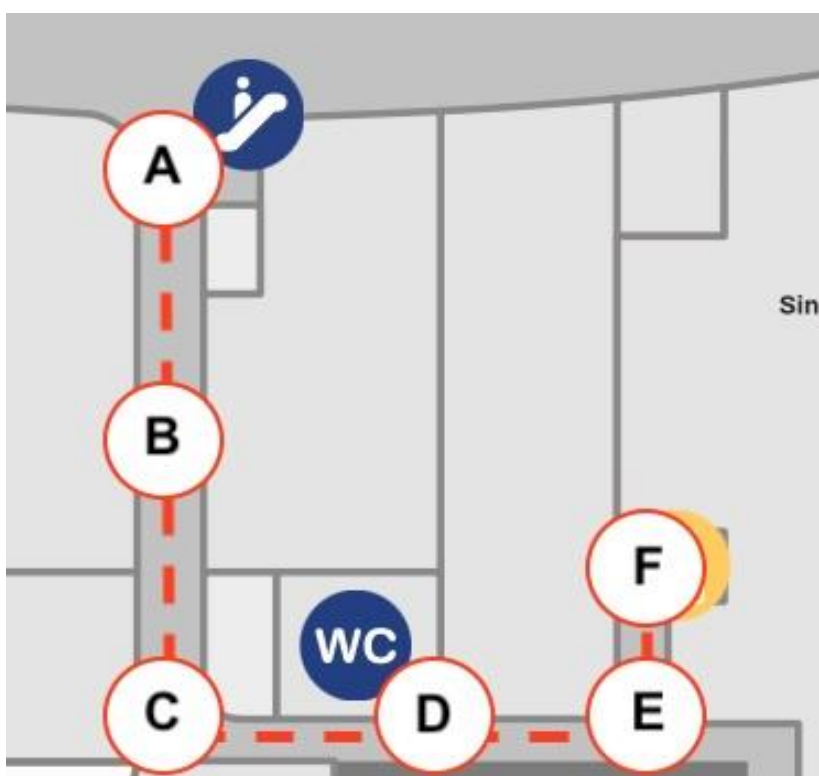


Kuva 10. Kartoitettavat alueet, 2. kerros ylempänä. Reitit merkitty punaisilla katkoviivoilla.

Projektiryhmän työaika Triplassa rajoittui aikaiseen aamuun, koska kello yhdeksän aikoihin asiakasmäärä lähti kasvuun ja viimeistään kello kymmeneltä väenpaljous teki tehokkaan työskentelyn käytännössä mahdottomaksi. Tulosten

analysointiin ja sovelluskehitykseen projektiryhmällä oli käytössään työskentelytila Haaga-Helian kampuksella kauppakeskuksen välittömässä läheisyydessä.

Kartoittaminen aloitettiin 2. kerroksesta. Reitti jaettiin viiteen 5–10 metrin pituiseen osaan, joista jokaista lähdettiin kartoittamaan erikseen (kuva 11). Tavoitteena oli saada malli toimimaan niin, että navigointisovellus löytäisi sijainnin 2–3 sekunnissa riippumatta siitä, mistä kohtaa reittiä käyttäjä aloittaa navigoinnin.



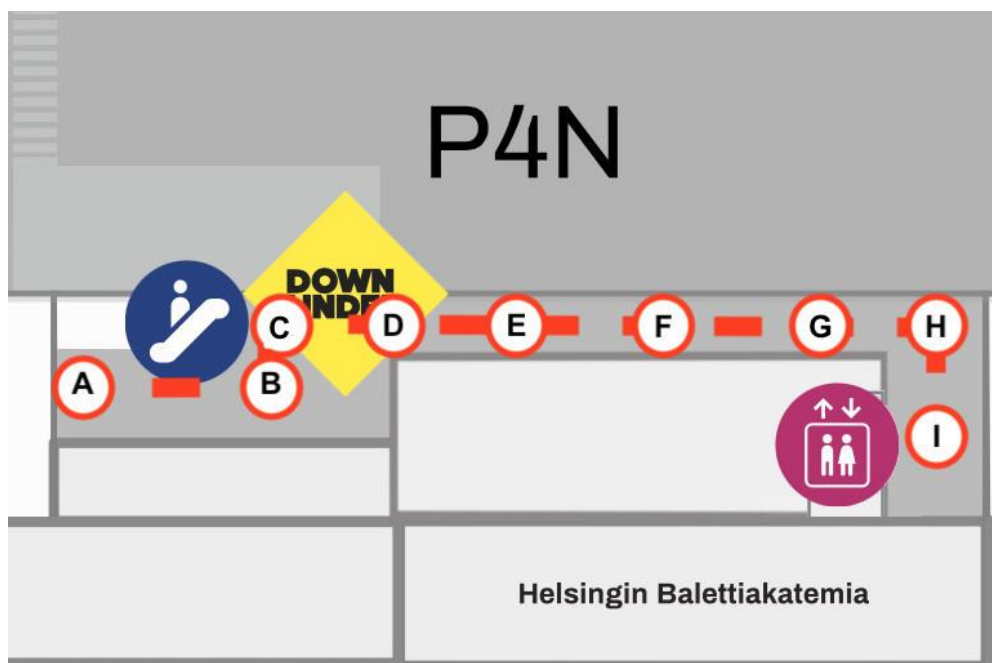
Kuva 11. 2. kerroksen reitti. Aakkosilla merkityt ympyrät näyttävät kartoitettujen pätkien alku- ja loppupisteet.

Kartoittamiseen käytettiin aluksi Huawei Mate Pro 20 -älypuhelin (Android), mutta sillä saadut tulokset olivat vaihtelevia. Esimerkiksi reitin väliä B–C ei saatu Huaweiilla toimimaan niin, että navigaatio-sovellus olisi pystynyt paikantamaan käyttäjää riittävän nopeasti ja johdonmukaisesti. Aiemmissa

projekteissa kartoitussovelluksen iOS-version oli havaittu toimivan Androidia luotettavammin, joten kartoitusta yritettiin uudelleen Applen iPhone 12:lla.

Puhelimen vaihtamisen jälkeen 2. kerroksen kartoitus sujui ongelmitta. VimAI:lla ei projektin aikaan ollut tarjolla toimivaa navigointisovellusta iOS-alustalle, joten navigoinnin toimivuuden testaukseen käytettiin Huaweita. Reitti kartoitettiin kumpaankin suuntaan kahdesti läpi, minkä jälkeen navigointisovellus kykeni paikantamaan käyttäjän reitin eri osissa. Kokonaisen mallin opettamiseen VimAI:n pilvipalvelimella kului noin 20 minuuttia.

P4-kerroksessa kartoitusta vaikeutti heikko internetyhteys. Triplan oma Wi-Fi ei ollut kerroksessa käytettävissä, ja puhelimen 4G-signaalin voimakkuus oli korkeintaan välttävä. Navigointisovellus toimi hitaasti ja epävarmasti, koska yhteys VimAI:n palvelimiin oli huono. Lisäksi ongelmia tuotti ympäristön homogeenisuus, mikä oli odotettavissa aiempien kokemusten perusteella. Näiden seikkojen takia kartoitus tehtiin selvästi lyhyemmissä osissa kuin 2. kerroksessa: yhden kartoitusvälin pituus oli noin 3–5 metriä (ks. kuva 12).



Kuva 12. P4-kerroksen reitti.

P4-kerroksen navigointi saatiin suurimmalta osin toimimaan tyydyttävästi, tosin reitin välillä C—D sovellus käyttäytyi oudosti. Sovellus saattoi kyseisessä kohdassa tulkita käyttäjän kävelevän päinvastaiseen suuntaan tai sijoittaa hänet täysin reitin ulkopuolelle. Kyseinen reitin osa kartoitettiin useaan kertaan uudestaan, mutta ongelma ei poistunut. Tästä ja muista navigoinnin ongelmatilanteista raportoitiin päivittäin VimAI:lle.

5.2 VimPlugin-liitännäisen testaus

Kun testialueiden navigaatio oli saatu toimimaan, oli vuorossa Unity-liitännäisen testaus. Testauksessa käytettiin jälleen Huaweiin puhelinta, johon asennettiin ARTEST3-esimerkkiprojekti.

ARTEST3 on yksinkertainen sovellus, joka emuloi VimAI:n navigointisovelluksen toimintaa Unity-ympäristössä. Navigaatio-sovelluksen tapaan se antaa käyttäjän valita rakennuksen ja lataa VimAI:n palvelimelta tämän kartan. Navigointisovelluksesta poiketen ARTEST3 ei kuitenkaan kuvaa ympäristöä automaattisesti, vaan kameran laukaisemiseen vaaditaan puhelimen liikuttamista sivuttaissuunnassa (kuva 13). Kun käyttäjä on onnistuttu paikallistamaan, ARTEST3 avaa puhelimeen AR-näkymän (scene), jossa mahdolliset POI-pisteet näkyvät tekstimuotoisina peliobjekteina.



Kuva 13. VimAI:n ohje puhelimen liikuttamiseen.

Unity-testaus aloitettiin 2. kerroksen reitillä, mutta lukuisista yrityksistä huolimatta ARTEST3-sovellusta ei saatu toimimaan. Rakennuksen valitseminen ja kartan lataaminen onnistuivat, mutta sovellus ei pystynyt paikallistamaan käyttäjää kartalle. Kameran kuva myös päivittyi paikannusvaiheessa hyvin hitaasti, mikä teki videokuvasta katkonaista. Sovellusta kokeiltiin reitin eri kohdissa, ja puhelinta liikuteltiin eri tekniikoilla ja nopeuksilla.

Ongelmaa lähdettiin selvittämään työtilana toimineessa Haaga-Helian tilassa kartoittamalla siellä pieni testialue. Siellä sovellus saatiin lopulta toimimaan, mutta paikantaminen onnistui vain satunnaisesti. Sovellus oli hyvin herkkä liikuttelunopeuden suhteen, ja kuvissa esiintyi usein liike-epäterävyyttä.

Ensimmäisenä toimenpiteenä ARTEST3-sovelluksesta poistettiin esimerkikoodissa 2 oleva vaatimus puhelimen liikuttamisesta. Tämän jälkeen puhelin paikansi käyttäjän johdonmukaisemmin ja nopeammin. Kameran päivitys ei kuitenkaan nopeutunut, ja tämän syyksi paikallistettiin se, ettei ARTEST3 käyttänyt suoraan Unityn videostreamia, vaan otti yksittäisiä valokuvia, jotka se lähetti VimAI:n palvelimelle tunnistamista varten.

```

if (m_isLocalizationDetecting)
{
    if (Input.acceleration.x > 0.1f) //tilt right
    {
        StartCoroutine(RecordFrame());
    }
}

```

Esimerkkikoodi 2. Puhelimen liikettä odottava rutiini.

Esimerkkisovelluksessa videokuvan ongelmat oli kierretty siten, että paikantamisen jälkeen sovellus lopetti kuvadatan keräämisen ja lähettämisen ja käytti paikantamiseen vain puhelimen liiketunnistinta. Tämä ei ole ihanteellinen ratkaisu ja hukkaa ison osan VimAI:n teknologisesta potentiaalista, koska liiketunnistin on käyttökelpoinen vain suhteellisen pienellä alueella. Laajemmassa ympäristössä sen virheet kertautuvat nopeasti ja tarkkuus kärsii. Tämän projektin puitteissa tarkkuus oli kuitenkin riittävä ja testaamista päätettiin jatkaa paikan päällä Triplassa.

5.3 AR-testit

Valtaosa testauksesta tapahtui 2. kerroksen käytävässä kellarikerroksen yhteysongelmien takia. Ensimmäisissä testeissä huomattiin käyttäjän sijoittuvan karttanäkymässä noin 5 metriä alaspäin y-akselilla ja liikkuvan päinvastaiseen suuntaan kuin todellisuudessa. Koodia tutkiessa selvisi, että y-akseli oli käännetty alkuperäisen testirakennuksen pienen koon takia (ks. esimerkkikoodi 3).

```

UserPointerImage.transform.localPosition = new Vector3(newPoiPos.x,
-newPoiPos.y , 0); //For building MSTEST1 the y coordinate is
flipped. Attributed to small enclosure

```

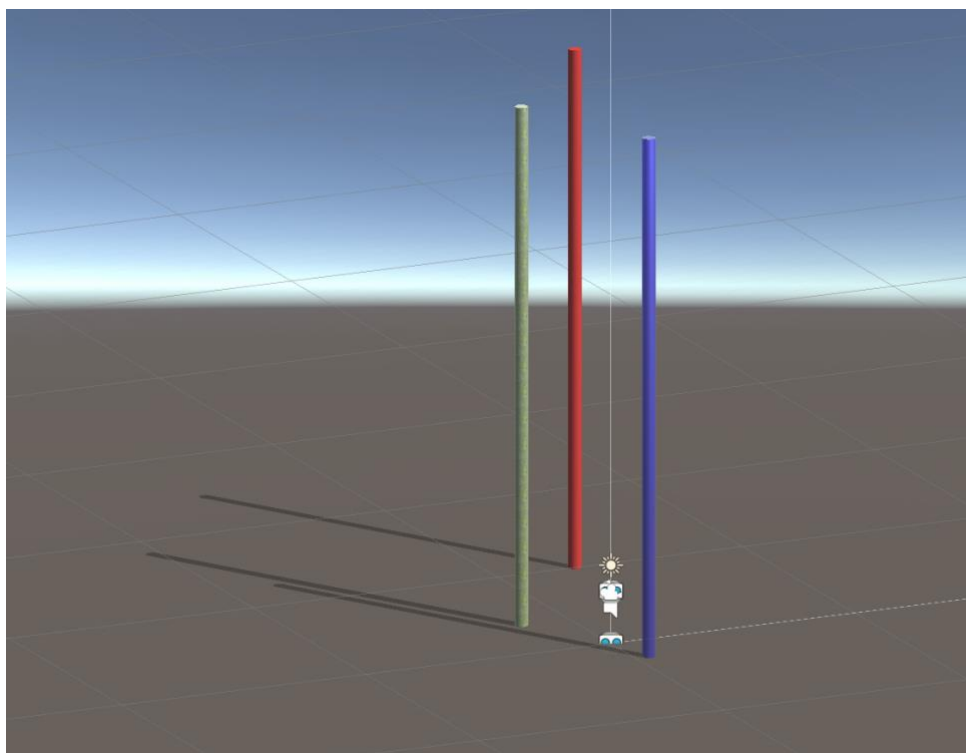
Esimerkkikoodi 3. Y-koordinaatin kääntäminen.

Y-akseli käännettiin positiiviseksi, ja lisäksi luotiin julkinen kokonaislukumuuttuja PointerOffset, jolla käyttäjän sijainti y-akselilla voitiin korjata säätämällä arvoa Unityn editorissa. Karttanäkymä sinänsä ei lopullisen sovelluksen kannalta ollut olennainen, koska sitä ei näytetä käyttäjälle, mutta kehitystyötä varten oli

tärkeää saada se vastaamaan todellisuutta. Yrityksen ja erehdyksen kautta sijainti kartalla saatiin kohdalleen asettamalla PointerOffset-muuttujalle arvo -3.

Seuraavaksi kokeiltiin POI-pisteiden sijoittumista AR-näkymässä.

Esimerkkiprojektissa POI-pisteet esitettiin Unityssa vain tekstinä, joiden näkyvyys varsinkin 2. kerroksen ympäristössä oli huono. POI-pisteiden käsittelyyn luotiin yksinkertainen skripti, jolla pisteet liitettiin Unityn peliobjekteihin. Testejä varten Unityyn luotiin useita erivärisiä tolppamaisia objekteja (kuva 14), joiden ajateltiin erottuvan hyvin ympäristöstä. Nämä tolpat yhdistettiin eri POI-pisteisiin, jotta saatiin selvitettyä, miten pisteet sijoittuvat AR-näkymään



Kuva 14. Testiobjektit Unityn editorissa.

Ensimmäisissä testeissä tolppien havaittiin sijoittuvan oikeille etäisyyksille, mutta väärään suuntaan. Virhe oli noin 180 astetta, joten päätettiin kokeilla esimerkikoodin 4 mukaista z-akselin kääntämistä.

```
Vector3 poiRawPos = new Vector3((float)p.position.x,  
(float)p.position.y, (float)p.position.-z);
```

Esimerkkikoodi 4. Z-akselin kääntäminen.

Seuraavissa kokeiluissa tolpat kuitenkin edelleen ilmestyivät AR-näkymässä väärin paikkoihin, joten z-akseli palautettiin positiiviseksi. Melko pian selvisi, että tolppien sijoittuminen riippui siitä, missä kohtaa reittiä käyttäjän paikallistaminen tapahtui. Joskus tolpat saattoivat osua lähelle oikeita kohtia, joskus taas reilusti sivuun. Tästä voitiin päätellä, ettei sovellus pystynyt määrittelemään käyttäjän rintamasuuntaa luotettavasti. Kartoitusta testattiin varmuuden vuoksi uudestaan myös VimAI-navigaatio-sovelluksella, ja se löysi oikean rintamasuunnan johdonmukaisesti, joten ongelman oli oltava AR-sovelluksessa.

Tässä vaiheessa otettiin yhteyttä VimAI:hin. Keskustelujen perusteella tämänkin ongelman juurisyiksi pääteltiin kameran käsittely. Ensiapuna kokeiltiin kameran intrinsics-parametrien säätämistä, mutta nämä yritykset sekoittivat sijoittumista entisestään. Koska projektiryhmän osaaminen ei riittänyt ongelman selvittämiseen ja valitsemalla huolellisesti navigoinnin aloituspaikka oli kuitenkin mahdollista simuloida toimintaa tyydyttävästi, päätettiin keskittyä AR-sovelluksen pelillisen sisällön kehittämiseen.

5.4 AR-sovellus

Testialueiksi valitut käytävät tarjosivat otollisen ympäristön esteradan tyyppiselle pelille. Triplan edustajan kanssa käytyjen keskustelujen perusteella AR-sovelluksen kohderyhmäksi valikoituivat lapset ja nuoret, joten peli haluttiin pitää leikkisänä ja väkivallattomana. Lopulta päädyttiin viidakkoteemaan, jossa käyttäjä väistelee esteitä ja poimii puista putoavia hedelmiä pisteitä kerätäkseen.

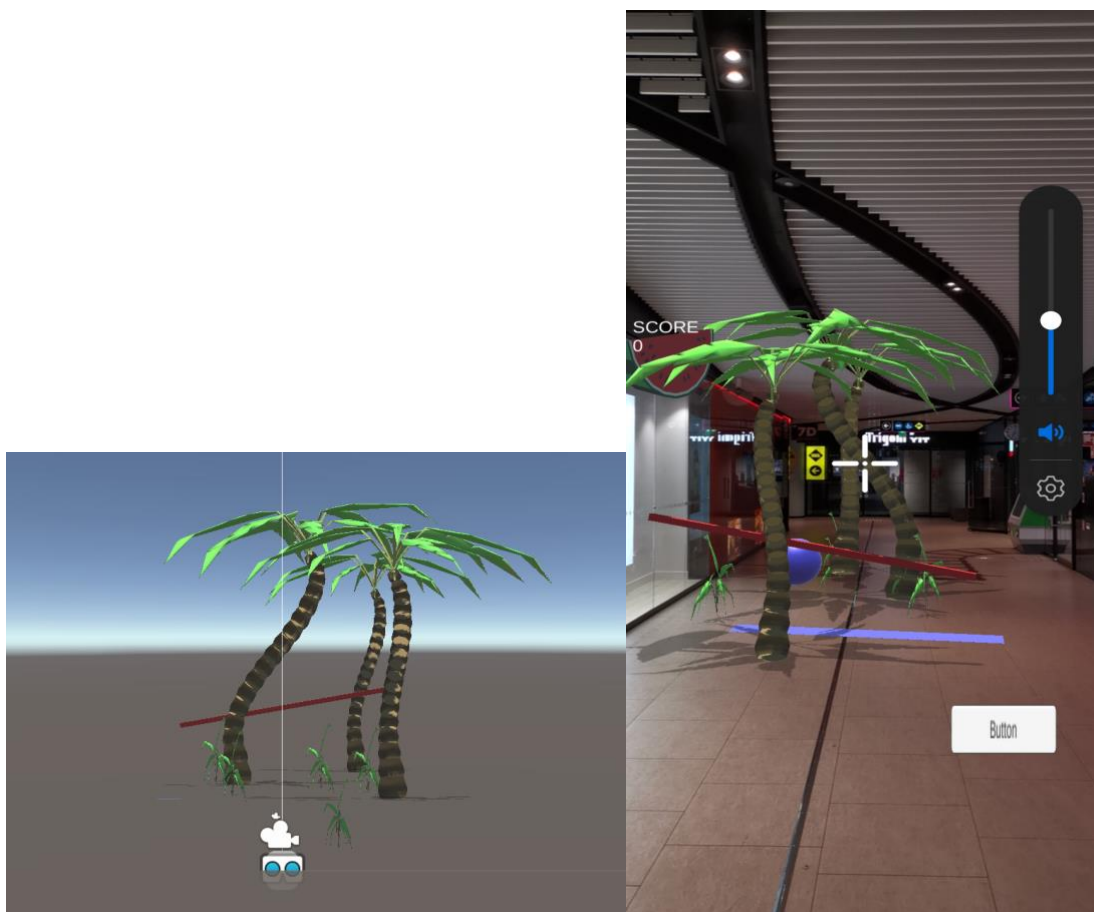
Peliympäristön rakentaminen aloitettiin luomalla Blender-ohjelmalla muutamia yksinkertaisia kolmiulotteisia malleja, kuten palmuja ja hedelmiä. Nämä mallit

tuotiin Unityyn, ja niistä luotiin ns. prefab-objekteja, joita pystyttiin liittämään kartalla oleviin POI-pisteisiin. Aluksi kokeiltiin yksittäisten 3D-kappaleiden sijoittelua pelkästään POI-pisteiden avulla, jolloin jokaista pistettä vastasi yksi kappale. Tämä havaittiin kuitenkin pian epäkäytännölliseksi, koska POI-pisteitä ei voi asettaa Unityn sisällä, vaan se täytyy tehdä kartoitussovelluksella. Näin ollen niiden sijoittaminen oikein Unityn AR-näkymään vaatii paljon työtä yrityksen ja erehdyksen kautta. POI-pisteiden siirtäminen ja poistaminen on myös kömpelöä.

Tehokkaampi ratkaisu on liittää POI-pisteeseen laajempi kokonaisuus. Unityn prefab-objektit voivat sisältää melkein mitä tahansa: tekstiä, 3D-objekteja, skriptejä tai vaikka kokonaisen pelin. Periaatteessa olisi mahdollista sijoittaa koko peli yhteen POI-pisteeseen, mutta tällöin menetettäisiin iso osa VimAI:n eduista, koska käyttäjän alkupaikallistamisen jälkeen vastuu pelaajan liikkeen seuraamisesta olisi Unitylla. Pelin suunnittelu vaikeutuisi, koska pelin sisäisellä ympäristöllä on vain yksi kiintopiste ulkoiseen maailmaan ja kaikki objektit pitäisi sijoitella suhteessa siihen. Riski virheisiin, kuten objektien sijoittumiseen seinien ulkopuolelle, kasvaa, mitä laajemmasta pelistä on kyse.

Käytännössä järkevin strategia on POI-pisteiden liittäminen erilaisten yksittäisten tehtävien ja pelinäkemien alkupisteisiin, jolloin pelaaja siirtyy pelin vaiheesta toiseen aina POI:lle saapuessaan. Kun POI-pisteeseen liitetyn pelinäkymän kattama alue pidetään tarpeeksi pienenä, voidaan luottaa siihen, että peliobjektit pysyvät määriteltyjen fyysisten rajojen sisällä.

Projektisovelluksessa päädyttiin rakentamaan noin 10 metriä pitkä rata, jossa käyttäjä joutui ohittamaan erilaisia esteitä kulkemalla niiden yli, ohi tai ali (kuva 15).



Kuva 15. Lopullinen sovellus Unityn editorissa (vas.) ja käytössä Triplan 2. kerroksen käytävässä.

Palmuista putoavat hedelmät ja niiden poimiminen toteutettiin yksinkertaisella skriptillä, joka oli sovellettu versio Parth Anandin (23) AR-pelistä, jossa käyttäjä ampuu kohoavia ilmapalloja. Unityn laajan tuen ja dokumentaation ansiosta pelimekaniikka oli suhteellisen helppo saada toimivaksi.

Valmista pelinäkömää testattiin molemmilla kartoitetuilla alueilla. Vaikka POI-pisteiden sijoittuminen AR-näkymään oli edelleen epävarmaa, peliä pystyi kuitenkin pelaamaan ja illuusio lisätystä todellisuudesta välittyi kohtalaisesti hiomattomasta ulkonäöstä huolimatta. Myös peliobjektien sijoittelu tilaan sopivasti onnistui hyvin.

Monet pelikokemuksen kannalta olennaiset osa-alueet, kuten pisteytys, käyttöliittymä ja tarina jäivät puutteellisiksi, mutta kokonaisuutena pelissä oli

mukana useita peruselementtejä, joita tällä projektilla haluttiin kokeilla ja esitellä. Jatkokehittelyä varten Unity ja ARCore tarjoavat runsaasti työkaluja ja teknologioita, joilla käyttäjäkokemusta voidaan parantaa.

6 Yhteenveto

Sisätilapaikannus hakee teknologiana vielä suuntaansa, eikä yhtä ylivoimaista teknistä ratkaisua ole toistaiseksi ilmaantunut. Siltä puuttuu vielä ns. killer app, sovellus tai palvelu, joka herättäisi kuluttajien ja median mielenkiinnon. Yksi vaihtoehto alustaksi tälle sovellukselle on VimAI, jonka etuina on kevyt käyttöönotto ja kyky käyttäjän rintamasuunnan tunnistamiseen.

VimAI:n integrointi Unityn kanssa on erittäin lupaava kehityspolku ja tärkeä askel kohti houkuttelevaa kaupallista sovellusta, mutta tässä insinööritoimintaprojektissa esiin tulleet ongelmat on ensin ratkaistava. Kuvan hidas päivitys tekee laajempien projektien kehittämisestä käytännössä mahdotonta, eikä se myöskään tarjoa sellaista käyttäjäkokemusta, jota kaupalliselta sovellukselta vaaditaan. Myös dokumentaatio ja muu tuki kehittäjille vaatii lisätyötä.

Kevyt ja laajasti käytettyihin alustoihin perustuva tekniikka kuitenkin mahdollistaa kehitystyön pienillä resursseilla ja leveällä rintamalla. Tämän projektin yhtenä tavoitteena oli luoda toimiva pohja jatkokehitykselle. Syksyllä 2021 työ jatkui uusien opiskelijoiden toimesta, joten tässä suhteessa projekti saavutti tavoitteensa. Myös kauppakeskus Triplan ja VimAI:n edustajat olivat tyytyväisiä tuloksiin ja näkivät yhteistyössä paljon potentiaalia.

Lähteet

- 1 Kunhoth, Jaykanath; Karkar, Abdelghani; Al-Maadeed, Somaya; Al-Ali, Abdulla. 2020. Indoor positioning and wayfinding systems: a survey. *Human-centric Computing and Information Sciences*. Volume 10. Article number 18. May 2020.
- 2 How Do You Measure Your Location Using GPS? 2022. Verkkoaineisto. National Institute of Standards and Technology. <<https://www.nist.gov/how-do-you-measure-it/how-do-you-measure-your-location-using-gps>>. 2.3.2022. Luettu 27.3.2022.
- 3 Why can't GPS be used for indoor positioning? Verkkoaineisto. Redpoint. <<https://www.redpointpositioning.com/blog-gps/>>. Luettu 29.9.2021.
- 4 Kim, Chongwon; So, Hyounghmin; Lee, Taikjin & Kee, Changdon. 2014. A Pseudolite-Based Positioning System for Legacy GNSS Receivers. *Sensors* 2014;14(4): 6104–6123.
- 5 Weill, Lawrence R. 1997. Conquering Multipath: The GPS Accuracy Battle. *GPS World*, April 1997, s. 59–65.
- 6 Kotaru, Manikanta; Joshi, Kiran; Bharadia, Dinesh & Katti, Sachin. 2015. SpotFi: Decimeter Level Localization Using WiFi. *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. August 2015, s. 269–281.
- 7 Greater choice for wireless access point owners. 2011. Verkkoaineisto. Google. <<https://googleblog.blogspot.com/2011/11/greater-choice-for-wireless-access.html>>. 14.11.2011. Luettu 3.4.2022.
- 8 Akpınar, Ege. 2021. Bluetooth beacons: Everything you need to know. Verkkoaineisto. Pointr. <<https://www.pointr.tech/blog/beacons-everything-you-need-to-know>>. 14.7.2021. Luettu 30.9.2021.
- 9 Lahtinen, Mirka. 2020. PoC: Sisätilapaikannusjärjestelmän kokeilu kaupunginmuseossa. *Insinööriyö*. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.
- 10 Giuliano, R.; Cardarilli, G.C.; Cesarini, C.; Di Nunzio, L.; Fallucchi, F.; Fazzolari, R.; Mazzenga, F.; Re, M.; Vizzarri, A. 2020. Indoor Localization System Based on Bluetooth Low Energy for Museum Applications. *Electronics* 2020, 9, s. 1055.

- 11 US3789351A. 1970. Guidance system. US Navy, Washington DC, USA. (Feldman, David W., Slone, James C.) 78341, 4.9.1970. Julk. 29.1.1974.
- 12 Truong, Alice. 2013. IndoorAtlas Hopes to Unlock the “Holy Grail of Advertising” With Magnetic-Field Mapping. Verkkoaineisto. Fast Company. <<https://www.fastcompany.com/3018063/indooratlas-hopes-to-unlock-the-holy-grail-of-advertising-with-magnetic-f>>. 25.9.2013. Luettu 29.3.2022.
- 13 Li, Binghao; Gallagher, Thomas; Dempster, Andrew & Rizos, Chris. 2012. How feasible is the use of magnetic field alone for indoor positioning? International Conference on Indoor Positioning and Indoor Navigation. November 2012.
- 14 Li, Shuang; Yu, Baoguo; Jin, Yi; Huang, Lu; Zhang, Heng & Liang, Xiaohu. 2021. Image-Based Indoor Localization Using Smartphone Camera. Wireless Communications and Mobile Computing. Volume 2021.
- 15 Nguyen-Huu, Khanh; Lee, Kyungho; Lee Seon-Won. 2017. An Indoor Positioning System Using Pedestrian Dead Reckoning with WiFi and Map-matching Aided. 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN). s.1–8.
- 16 VimAI. Verkkoaineisto. <<https://vim.ai/>>. Luettu 3.10.2021.
- 17 VimAI Collection Client. 2020. VimAI Oy.
- 18 VimAI Navigation Client VimAI Oy.
- 19 ARCore supported devices. Verkkoaineisto. Google. <<https://developers.google.com/ar/devices>>. Luettu 3.10. 2021.
- 20 Axon, Samuel. 2016. Unity at 10: For better—or worse—game development has never been easier. Verkkoaineisto. Ars technica. <<https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/>>. 27.9.2016. Luettu 30.9.2021.
- 21 Unity. 2022. Unity Technologies.
- 22 Edelstein, Stephen. 2018. How gaming company Unity is driving automakers toward virtual reality. Verkkoaineisto. Digitaltrends. <<https://www.digitaltrends.com/cars/unity-automotive-virtual-reality-and-hmi/>>. 17.5.2018. Luettu 30.9.2021.

- 23 Anand, Parth. 2020. ARFoundation Unity Tutorial - Augmented Reality Shooting app. Verkkoaineisto. YouTube. <https://youtu.be/jDj_UF8ke48>. Katsottu 2.7.2021.