



# Haasteet täydennetyin todellisuuden mobiilisovelluksen luomisessa

Penjami Rantakangas

OPINNÄYETYÖ  
Kesäkuu 2019

Tietojenkäsittely  
Ohjelmistotuotanto

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittely  
Ohjelmistotuotanto

RANTAKANGAS, PENJAMI:

Haasteet täydennetyt todellisuuden mobiilisovelluksen luomisessa

Opinnäytetyö 33 sivua

Kesäkuu 2019

---

Opinnäytetyössä tutkittiin parhaita työkaluja ja teknologioita täydennetyt todellisuuden mobiilisovelluksen luomiseen. Opinnäytetyössä käsiteltiin täydennetyt todellisuuden historiaa. Työssä tarkasteltiin yksityiskohtaisesti kolmea täydennetyt todellisuuden ohjelmistokehystä: Googlen ARCorea, Applen ARKittiä ja Vuforiaa. Työn toimeksiantaja oli Intopalo Digital.

Käytännön osuudessa analysoitiin projektin työkaluja ja teknologioita, ja käytiin lävitse projektin aikana tulleita haasteita ja ratkaisuja. Mobiilisovelluksen optimoinnissa käytettiin 3d-grafiikan optimointitekniikoita, jotka sopivat mobiililaitteille, kuten staattisten varjojen käyttäminen ja yksinkertaisten materiaalien hyödyntäminen. Tuotteen sisällön skaalaukseen käytettiin kameran sijaintia luomaan optisen illuusion skaalauksesta. Tuotteen versioiden rakentamisessa hyödynnettiin Unityn Cloud Build -palvelua.

Opinnäytetyön tuotos on Intopalo Digitalille laajennetun todellisuuden markkinointidemo, joka on uusille iOS-laitteille kehitetty mobiilisovellus. Demo näyttää Intopalo Digitalin eri osaamisalueet animoidussa dioraamassa, johon voi olla vuorovaikutuksessa kosketusnäytön avulla. Täydennetyt todellisuuden tulevaisuus mobiililaitteissa vaikuttaa lupaavalta. Laitteiston kannalta mobiililaitteet ovat tarpeeksi tehokkaita ja niissä on tarpeeksi hyviä sensoreita luomaan vakuuttavia täydennetyt todellisuuden sovelluksia, mutta työkaluissa ja ohjelmistoissa on vielä puutteita. Opinnäytetyön tuotosta voitaisiin jatkokehittää tukemalla muita mobiilialustoja ja vaihtamalla tunnistusteknologiaa vakaampaan ja nopeampaan vaihtoehtoon.

## **ABSTRACT**

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

**PENJAMI RANTAKANGAS:**  
Challenges in Creating an Augmented Reality Mobile Application

Bachelor's thesis 33 pages  
June 2019

---

This thesis was made to study the current technologies of creating an augmented reality mobile application and to solve challenges in the creation process. The client for this thesis was Intopalo Digital. The theoretical section explores the current and past mobile augmented reality technologies and compares them.

The practical portion of the thesis analyzes the challenges and solutions in creating an augmented reality mobile application. Standard 3d optimization techniques were used in the optimization of the application, such as using static shadows and simple materials. Scaling was done by moving the position of the camera in relation with the augmented reality content. Unity's Cloud Build service was used to build versions of the project.

The product of the thesis is an augmented reality marketing demo for Intopalo Digital made to run on last two generations of mobile iOS devices. The product was created using ARKit from Apple and the Unity game engine. The demo showcases Intopalo Digital's different expertise in an animated diorama that can be interacted with using the device's touch screen.

The product can be improved by changing the tracking technique to a stable and faster alternative and by supporting more mobile devices. The future of mobile augmented reality is promising. The technology and hardware of the current mobile devices are impressive, but the tools and software are still in the early stages of development and will need improvements in usability and stability.

## SISÄLLYS

1	JOHDANTO .....	6
2	VIRTUAALINEN MATERIAALI TODELLISUUDESSA .....	7
	2.1 Yhdistetty todellisuus .....	7
	2.2 Täydennetty todellisuus .....	7
	2.3 Käyttäjän ja laitteen seuranta .....	8
	2.4 Mobiililaitteet .....	9
3	SUUNNITTELU JA TEKNIikka .....	11
	3.1 Unity-pelimoottori .....	11
	3.2 AR-laajennuksen valinta .....	11
	3.3 ARKit 2.0 .....	14
	3.4 Markkinointi .....	15
4	TOTEUTUKSEN HAASTEET .....	16
	4.1 Totetutuksen esittely .....	16
	4.2 Immersio ja väärät tunnistukset .....	16
	4.3 Okluusio .....	18
	4.4 Cloud build .....	18
	4.5 Suorituskyky .....	20
	4.5.1 Partikkeliefektit .....	22
	4.6 Skaalaus .....	23
	4.6.1 Universaali skaala .....	23
	4.6.2 Kaksi kameraa .....	24
	4.6.3 ARSessionOrigin .....	25
	4.7 Käyttöliittymä .....	26
	4.8 Käyttäjän vuorovaikutus .....	26
5	POHDINTA .....	28
	5.1 Jatkokehitys .....	28
	5.2 Täydennetyn todellisuuden tulevaisuus .....	30
	LÄHTEET .....	32

**LYHENTEET JA TERMIT**

AR	Augmented Reality, täydennetty todellisuus
Partikkeli	Yksinkertainen kappale, joka liikkuu osana partikkeliefektiä
Pistepilvi	Kokoelma pisteiden sijainneista kolmiulotteisessa ympäristössä
Sävytin	Grafiikkakortilla suoritettavaa koodia, joka on usein vastuussa 3d-mallin pinnan materiaalista ja heijastuksesta.
XR	Yhdistetty todellisuus

## 1 JOHDANTO

Tässä opinnäytetyössä tutkitaan täydennetyin todellisuuden mobiilisovelluksen rakentamisen haasteita ja ratkaisuja. Toimeksiantajana opinnäytetyölle on Intopalo Digital Oy, joka on Tampereella ja Saksan Munichissa toimiva ohjelmistoalan konsultointifirma. Intopalo Digital luo ratkaisuja eri ohjelmistoalan osa-alueilla, kuten web, XR ja tietoturva. Opinnäytetyön tuotoksena loin Unity-pelimoottorilla täydennetyin todellisuuden markkinointidemon. Markkinointidemo on kolmiulotteinen animoitu dioraama, jossa esitellään Intopalo Digitalin osaamista.

Opinnäytetyön tarkoituksena on luoda Intopalon myyntitiimille mobiililaitteilla pyörivä täydennetyin todellisuuden markkinointidemo, jota he voivat käyttää seminaareissa ja asiakastapaamisissa esittelemään Intopalon osaamista. Tarkoituksena on myös luoda opinnäytetyön raportti, jossa esittelen projektin aikana esille tulleita haasteita ja ratkaisuja. Vertailen mobiili AR:n eri työkaluja ja tekniikoita. Käyn lävitse merkittävät vaiheet täydennetyin todellisuuden historiasta ja tutkin täydennetyin todellisuuden tulevaisuutta mobiililaitteissa. Selvittelen myös, mitkä asiat pidättelevät täydennetyin todellisuuden laajempaa käyttöönottoa ja kehittelyä.

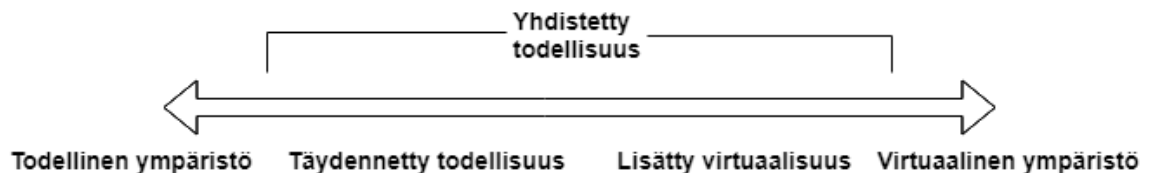
Opinnäytetyön tavoite on tutkia mobiililaitteille tehdyn täydennetyin todellisuuden sovelluksen haasteita suorituskyvyssä, täydennetyin todellisuuden illuusion ylläpitämisessä, työkalujen yhteensopivuudessa ja mobiililaitteiden rajoittuneisuudessa. Opinnäytetyön tavoitteena on myös selvittää täydennetyin todellisuuden mahdollisuuksia markkinoinnin kannalta.

## 2 VIRTUAALINEN MATERIAALI TODELLISUUDESSA

### 2.1 Yhdistetty todellisuus

Yhdistetty todellisuus on termi, jonka Paul Milgram ja Fumio Kishino (1994) esittelivät tutkimuksessaan lisätyn, virtuaalisen ja yhdistetyn todellisuuden luokittelusta. Yhdistetty todellisuus on välimaasto todellisuuden ja virtuaalisen maailman välissä. Yhdistettyä todellisuutta luokitellaan kolmella kategoriolla: Järjestelmän kyky ymmärtää ympäristöä, virtuaalisen informaation laatu, käyttäjän immersion laatu. (Milgram & Kishino 1994.)

Milgramin ja Kishinon jatkumon toisessa päässä on fyysinen ympäristö, jossa ei ole mitään tietokoneella generoitua, ja toisessa päässä on virtuaalinen ympäristö, jossa kaikki on tietokoneella generoitu. Jatkumo peittää kaikki mahdolliset oikean ja virtuaalisen ympäristön yhdistelmät ja variaatiot. (KUVIO 1.)



KUVIO 1. Yksinkertaistettu kuvio virtuaalisesta jatkumosta (Milgram & Kishino 1994, muokattu)

Teknologiset saavutukset ovat mahdollistaneet yhä enemmän yhdistetyn todellisuuden kokemuksia. Tällä hetkellä ei ole olemassa laitetta, joka kattaisi koko yhdistetyn todellisuuden jatkumon. Laitteet tukevat tietyn alueen jatkumosta, ja ajan mittaan uusein laitteiden pitäisi laajentaa aluetta. Virtuaali-todellisuuden laitteet, kuten Oculus Rift ja HTC Vive, ovat jatkumon oikeassa reunassa, kun taas Microsoftin hololens kattaa jatkumon puolesta välistä vasempaan reunaan. (Microsoft n.d.)

### 2.2 Täydennetty todellisuus

Täydennetty todellisuus, kansainväliseltä lyhenteeltään AR, on teknologia, joka mahdollistaa ympäristön rikastamista virtuaalisella informaatiolla (Van Krevelen 2007, 2). Täydennetty todellisuus pyrkii yksinkertaistamaan käyttäjän elämää tuomalla virtuaalista informaatiota hänen ympäristöönsä. Fruth (2011, 3) ei pidä täydennettyä todellisuutta vain näköaistiin rajoittuvana. Täydennetty todellisuus pätee myös muihin aisteihin, kuten hajuaistin, kosketuksen tai kuulon augmentointiin. (Furht 2011, 3.)

Ivan Sutherland ja hänen oppilaansa kehittivät ensimmäiset AR prototyypit 1960-luvulla. He käyttivät läpinäkyvää kypärää, joka oli katossa kiinni käyttäjän liikkeitä tallentavalla mekaanisella kädellä. 1990-luvun loppupuolella täydennetystä todellisuudesta tuli oma tutkimusalanensa ja alettiin järjestää useita konferensseja AR:n mahdollisuuksista. Näihin aikoihin hyvin rahoitetut organisaatiot MRLab Japanissa ja Arvika consortium Saksassa saivat myös alkunsa. (Van Krevelen 2007, 2.)

Tietokoneiden web-kamerat mahdollistivat täydennetyin todellisuuden käytön laajemmin kuin koskaan ennen. Tietokoneohjelma käytti kameraa tunnistamaan jonkin kohteen, kuten QR-koodin tai jonkin kuvan. Jotkut kehittyneemmät ohjelmat pystyivät tunnistamaan ja seuraamaan kameralla käyttäjän kasvoja tai käsiä. Seurattavan kohteen päälle piirrettiin jotain digitaalista sisältöä, kuten aurinkolasit kasvoille. (Lojiens 2017, 15.)

### **2.3 Käyttäjän ja laitteen seuranta**

On olemassa kaksi optista tapaa tulkita ympäristöä suhteessa käyttäjään: piirteiden seuranta ja merkkien seuranta. Seuratut merkit voivat olla aktiivisia tai passiivisia. Passiivisten merkkien yhteisenä piirteenä on kuvio, joka erottuu helposti ympäristön tekstuurista. Yksi esimerkki passiivisen merkin seurannasta on QR-koodien lisääminen ympäristöön seurattavina merkkeinä. Passiivisten AR-merkkien ei tarvitse välttämättä olla mustia ja valkoisia ruutuja, kuten QR-koodi, vaan ne voivat olla lähes mitä tahansa kaksiulotteista kuvaa (Lojiens 2017, 19). Aktiiviset merkit lähettävät jotain signaalia, kuten magneettista tai valoa, jota pystyy seuraamaan sensorilla. (Bostanci ym. 2013, 93.)



Piirteiden seurantaan on tarkoitus luoda menetelmä ilman ennalta lisättyjä AR-merkkejä. Yksi tapa tehdä piirteiden seuranta on etsiä kameran videon peräkkäisistä kuvista yhtäläisyyksiä. Korkean kontrastin yhtäläisyyksistä tehdään pisteitä, joiden perusteella voidaan arvioida kamerallisen laitteen liikkeitä. Piirteiden seuranta vaatii yleensä paljon enemmän laskentatehoa, kuin merkkien seuranta, mutta se toimii valmistelemattomassa ympäristössä. (Bostanci ym. 2013, 94.)

Sisätilat ovat usein paras ympäristö täydennetyille todellisuudelle, koska sisätiloissa on helpompi vaikuttaa ympäristön valoon ja mittasuhteisiin. Sisätiloissa käyttäjän liikkeitä on myös helpompi ennalta arvioida. Valmis sisätila mahdollistaa seurantalaitteiden esiasentamisen ja kontrolloidun ympäristön. (Bostanci ym. 2013, 93.)

Ainoastaan yhdentyyppisen sensorin käyttö seurannassa voi johtaa seurannan heikkoon tarkkuuteen. On olemassa myös seurantamenetelmiä, jotka yhdistävät monta sensoria luomaan mahdollisimman suurta tarkkuutta. Näissä fuusioseurantamenetelmissä hyödynnetään optisten sensoreiden lisäksi gyroskooppeja, kiihdytysensoreita, GPS-vastaanotinta ja kosketusantureita. (Bostanci ym. 2013, 95.)

## **2.4 Mobiililaitteet**

Loomis ja hänen kumppaninsa kehittivät ensimmäisen täysin omavaraisen mobiilin AR kokemuksen vuonna 1993. He rakensivat GPS-pohjaisen järjestelmän, joka tarjoaa navigointiapua näköaistinsa menettäneille, käyttäen spatiaalista äänikenttää. Vuonna 1997 Feiner kehitti ensimmäisen mobiilin AR prototyypin, joka rekisteröi kolmiulotteista graafista matkaopasinformaatiota käyttäjän näkemiin rakennuksiin ja esineisiin. (Van Krevelen 2007, 2.)

Puhelimilla on täydennetyin todellisuuden suhteen etunsa ja haittansa. Puhelimien etuja ovat sensorien määrä, saatavuus ja helppokäyttöisyys. Useimmissa puhelimissa on nykyään takakamera, joka on yksi täydennetyin

todellisuuden perusvaatimus. Useimmissa puhelimissa on myös kiihtyvyyssmittari, gyroskooppi ja GPS, joista voi olla hyötyä täydennetyin todellisuuden sovellukselle (Furht 2011, 21). ARKit käyttää mobiililaitteen kiihtyvyyssmittaria ja gyroskooppia paikantamaan laitteen kolmiulotteisessa ympäristössä (Apple n.d.a). Puhelimien heikkouksia ovat kameroiden laatu ja laitteiston vaihtelevuus. (Furht 2011, 21.)

Ennen mobiililaitteissa ei ollut tarpeeksi tehoa prosessoida kuvista reaaliajassa pistepilveä, vaan silloin laskennallisesti vaativat operaatiot tehtiin verkon kautta tietokoneella ja valmiit tiedot lähetettiin takaisin mobiililaitteelle (Furht 2011, 22). Nykyään uusimmissa puhelimissa ja tableteissa on tarpeeksi prosessoritehoa tehdä kaikki laskut lokaalisti laitteella (Apple n.d.a.)

### 3 SUUNNITTELU JA TEKNIikka

#### 3.1 Unity-pelimoottori

Unity-pelimoottori on Unity Technologiesin rakentama monipuolinen pelimoottori, jolla on mahdollista luoda yli 25 eri alustalle 3d tai 2d pelejä tai ohjelmia. Unity sisältää laajan valikoiman työkaluja interaktiivisen täydennetyin todellisuuden kokemuksen luomiselle. Unityssa käytetty ohjelmointikieli on uusin versio Microsoftin kehittämästä C# kielestä. (Unity n.d.a.)

Alun perin Unitya käytettiin lähes yksinomaan pelinkehitykseen, mutta unityn työkaluja hyödynnetään yhä enemmän peliteollisuuden ulkopuolella. Visualisointiin erikoistunut yritys NVYVE käytti Unitya luodakseen kolmiulotteisen visualisaation Ottawan kaupungista ja yksityiskohtaiset mallit asunnoista kaupungin sisälle. Interaktiiviseen oppimiseen keskittynyt yritys CliniSpace on kehittänyt Unitylla virtuaalisia koulutusmahdollisuuksia. Geomedia teki Unitylla laajennetun todellisuuden opetuksellisen aarteenmetsästys ohjelman yhdessä Chicagon tieteen ja teollisuuden museon kanssa. (Adams 2014.)

Yli kaksi kolmasosaa kaikista VR ja AR kokemuksista on tehty Unitylla. Unitya käyttää useat isot pelistudiot, mainostoimistot ja elokuvastudiot. Jopa Google käyttää täydennetyin todellisuuden projekteissaan Unitya. (Unity n.d.a.) Kaiken tämän lisäksi minulla oli aikaisempaa kokemusta Unityn käytöstä, joten Unityn valitseminen tuotoksen kehitystyökaluksi oli helppo valinta.

#### 3.2 AR-laajennuksen valinta

Unitylle on olemassa useita eri täydennetyin todellisuuden kehitysalustoja, mutta tällä hetkellä suosituimmat ja ominaisuuksiltaan monipuolisimmat ovat ARKit, ARCore, Vuforia ja AR foundation. Kaikki vaihtoehdot tarjoavat uusinta teknologiaa hyödyntäviä ratkaisuja, mutta niistä löytyy paljon eroja laiteyhteensopivuuden ja seurantamenetelmien laadun kanssa. (Google n.d.a; Apple n.d.a; Unity 2018c).

ARCore on googlen kehittämä täydennetyin todellisuuden ratkaisu tietyille Android ja iOS laitteille. Laitteen liiketunnistus yhdessä kameran kuvan tunnistuksen kanssa luo ymmärrystä ympäristöstä ja laitteen sijainnista suhteessa ympäristöön. ARCore käyttää laitteen kameraa tunnistamaan pisteitä ympäristöstä ja reaaliajassa seuraa niiden liikettä suhteessa toisiinsa. ARCore pystyy myös tunnistamaan tasaisia pintoja, kuten pöydän tai lattian, ja arvioimaan valon määrän pinnan päällä. ARCoren ymmärrys maailmasta mahdollistaa 3d-mallien, tekstin, tai minkä tahansa muun virtuaalisen informaation lisäämisen ympäristöön. (Google n.d.a.)

ARCore vaatii laitteelta Android 7.0 käyttöjärjestelmän, suhteellisen hyvän kameran, kiihtyvyyssanturin, gyroskoopin ja tarpeeksi tehokkaan prosessorin (Google n.d.a). Näiden vaatimusten lisäksi Googlen pitää hyväksyä laite sopivaksi ennen, kun se on virallisesti yhteensopiva (Google n.d.a). Johtuen Android laitteiden monimuotoisuudesta, ARCorea tukevien laitteiden lista on yllättävän pieni. Tuettujen laitteiden listalla on tämän opinnäytetyön julkaisuaikana vain kolme Android tablettia. (Google n.d.c.)

ARKit on Applen kehittämä täydennetyin todellisuuden alusta, joka toimii Applen iOS-mobiililaitteilla. ARKit selvittää käyttäjän laitteen olinpaikan suhteessa ympäristöön liiketunnistuksen, kameran kuvan ja kuvan prosessointitekniikoiden avulla. ARKitillä on mahdollisuus käyttää täydennettyyn todellisuuteen laitteen taka- tai etukameraa, tai molempia samaan aikaan. ARKit on ominaisuuksiltaan kaikista vahvin vaihtoehtoista. (Apple n.d.a.)

Vuforia on täydennetyin ja yhdistetyin todellisuuden kehitysalusta. Vuforia tukee tablettien ja puhelinten lisäksi täydennetyin todellisuuden laseja, kuten Microsoftin HoloLenssiä. Toisin kuin ARKitissä ja ARCoressa, jotka tukevat vain uusimpia laitteita, Vuforia tukee lähes kaikkia Android ja iOS laitteista, kunhan niissä on vain takakamera. Vuforia käyttää ARKitin ja ARCoren teknologioita tilanteissa, joissa laitteisto tukee niitä, muuten Vuforia käyttää omaa täydennetyin todellisuuden teknologiaa, joka ei riipu laitteistosta. ARCore ja ARKit ovat täysin ilmaisia käyttää, kun taas Vuforian käyttöhintaa riippuu käyttäjän yrityksen koosta. (Unity 2018c.)

AR foundation on Unityn lisäosa, joka tarjoaa yhteisen ohjelmistorajapinnan Applen ARKitin ja Googlen ARCoren käyttöön Unityssa. (Miller, Mowrer & Weiers 2018.) AR foundation on vielä varhaisessa vaiheessa kehitystä, ja siitä puuttuu paljon ominaisuuksia, joita löytyy muista vaihtoehdoista. Tällä hetkellä AR foundationissa ei vielä ole kuvan tai esineen tunnistusta, vaan ainoastaan tasojen tunnistus. ARKit on taulukon mukaan ominaisuuksiltaan rikkain vaihtoehto. ARCoressa on yksi ominaisuus, jota muilta alustoilta ei löydy: Cloud anchorit, jotka ovat ankkuroituja pisteitä ympäristössä, joita muut saman sovelluksen käyttäjät voivat nähdä internetin välityksellä. (Google n.d.a; taulukko 1.)

TAULUKKO 1. AR Foundationin ominaisuudet verrattuna ARCoren ja ARKitin ominaisuuksiin Unityssa (Miller, Mowrer & Weiers 2018)

Supported Feature	AR Foundation	Google ARCore SDK for Unity	Unity ARKit Plugin
Plane Detection (Vertical)	✓	✓	✓
Plane Detection (Horizontal)	✓	✓	✓
Feature Point Detection	✓	✓ + Oriented Feature Points	✓
Light Estimation	✓	✓ + Color Correction	✓ + Color Temperature
Hit Testing (Feature point and Plane raycasting)	✓	✓	✓
Image Tracking	in development	✓ (Static Only)	✓
3D Object Tracking	in development		✓
Environment Probes	in development		✓
World Maps	✓		✓
Face Tracking (Pose, Mesh, Blendshapes)	✓		✓ iPhone X + Variants Only
Cloud Anchors	in development	✓	
Editor Remoting	in preview	✓ - Instant Preview	✓ - ARKit Remote
Editor Simulation	in preview		
LWRP support (+ Shader Graph*)	3.3.0 supported	in development	in development
Camera Image API	✓		

Täydennetyin todellisuuden kehitysalustojen testauksen jälkeen päädyin siihen, että ARKit oli paras vaihtoehto projektiin, koska siinä oli vaihtoehtoista paras tunnistusteknologia ja eniten ominaisuuksia. ARKit päivittää kameraa ja tunnistuksia 60 kertaa sekunnissa, kun taas ARCore päivittää kameraa 30 ruudunpäivitystä sekunnissa ja tunnistusta 10 kertaa sekunnissa (Apple n.d.a; Google n.d.d). Tämä johtaa siihen, että ARKitillä tehty täydennetty todellisuus näyttää ja tuntuu paljon sulavammalta. Näiden syiden lisäksi Intopalo Digitalon toimistolla hyvin monella oli Applen puhelin työpuhelimena, joissa projektin tuotos toimisi. Projektin toimivuus mahdollisimman monella laitteella ei ollut ykkösprioriteetti, vaan tärkeintä oli tehdä laadukas tuotos, joka näyttää hyvältä ja toimii hyvin.

### **3.3 ARKit 2.0**

ARKit 2.0 on Applen vuoden 2018 puolella välissä julkaistava päivitys heidän täydennetyin todellisuuden ohjelmistokehityspakettiin. Päivitys lisäsi paljon uusia hyödyllisiä ominaisuuksia ja paransi seurauksen tarkkuutta. ARKit 2.0 mahdollistaa ympäristöstä luodun pistepilven, ja kaikkien siihen tehtyjen muutoksien, tallentamisen seuraavaa käyttökertaa varten. ARKit 2.0 on mahdollista luoda sovelluksia, joissa useampi käyttäjä vaikuttaa samaan täydennettyyn todellisuuteen. ARKit käyttää iOS-laitteiden wifi tai bluetooth vertaisverkkoa kommunikoidaan täydennetyin todellisuuden kartoitetusta pistepilvestä ja siihen tehdyistä muutoksista. (Apple n.d.a.)

ARKitin 2.0 päivityksen mukana tulee myös esineiden tunnistus ja seuranta, ja liikkuvien kuvien seuranta. Esineiden tunnistuksessa ARKit pystyy tunnistamaan aikaisemmin skannattuja esineitä. Skannaus hoidetaan erillisellä ohjelmalla, jossa iOS-laitteen kamera luo esineestä pistepilven, jonka avulla on mahdollista tunnistaa oikea esine. Liikkuvien kuvien seurannassa ARKit pystyy seuraamaan kuvaa, vaikka se liikkuisi samaan aikaan kuin mobiililaitte. (Apple n.d.a.)

### 3.4 Markkinointi

Markkinointi on yksi suosituimmista tavoista hyödyntää täydennettyä todellisuutta kaupallisesti. Huonekalumyymäläketjun Ikean mobiilisovellus tarjoaa mahdollisuuden kokeilla mobiililaitteen kautta eri huonekaluja omassa kodissa, ennen kuin harkitsee ostoa. Joissain vaatekaupoissa ja silmälasiliikkeissä on niin sanottuja älypeilejä, jotka koostuvat kamerasta ja suuresta näytöstä. Älypeilin avulla voi nähdä itsensä eri vaatteissa tai silmälaisessa laittamatta niitä päällensä. (Lojiens 2017, 61.)

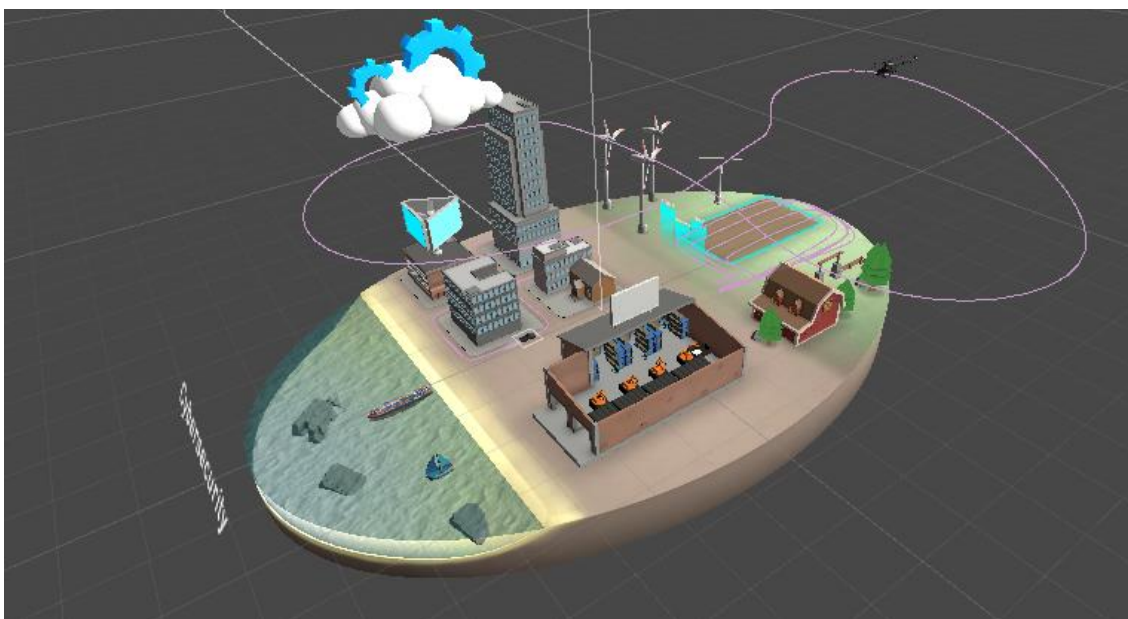
On vielä liian aikaista sanoa, mitä kaikkea täydennetyllä todellisuudella on mahdollista saavuttaa markkinoinnissa, mutta on huomattu tiettyjä asioita menestyneiden markkinointien kautta. Täydennetty todellisuus on vieläkin uutta monille, joka saattaa joidenkin silmissä lisätä täydennettyä todellisuutta hyödyntävän tuotteen arvoa. Täydennetty todellisuus voi myös lisätä tietynlaisen jännittävän ja hauskan elementin tuotteeseen, joka saattaa saada käyttäjän innostumaan tuotteesta entisestään. (Lojiens 2017, 66.)

## 4 TOTEUTUKSEN HAASTEET

### 4.1 Totetutuksen esittely

Toteutuksessa pyrin luomaan täydennetyn todellisuuden sovelluksen, jossa sisältönä on dioraama, jossa esitellään Intopalo Digitalin osaamista eri osa-alueilla. Kun sovellus tunnistaa skannatun esineen, joka meidän tapauksessa oli leluauto, sen päälle ilmestyy neljä virtuaalista skenaariota. Jokainen näistä skenaarioista on animoitu tarina Intopalo Digitalin osa-alueesta. (KUVA 1.)

Useimmat haasteet projektissa koostuivat uusien teknologioiden opettelemisesta ja odottamattomista tilanteista. Täydennetty todellisuus ja teknologiat, joita käytin tässä toteutuksessa, olivat sen verran uusia, ettei niistä välillä löytynyt tarvittavaa informaatiota. Vaan opettelu ja luominen pohjautui paljolti eri vaihtoehtojen testaukseen.



KUVA 1. Toteutus Unity-pelimootorin sisällä

### 4.2 Immersio ja väärät tunnistukset

Optimaalinen täydennetyn todellisuuden kokemus pitää jatkuvasti yllä immersiota ympäristöön sijoitetusta virtuaalisesta informaatiosta. Jotta immersio ei katoaisi, pitää ottaa huomioon sovelluksen lisäksi myös ympäristö, jossa sovellusta



käytetään. Täydennettyyn todellisuuteen vaikuttavia ympäristöön liittyviä tekijöitä on muunmuassa: valon määrä, piirteiden määrä ympäristön tekstuurissa ja ympäristön dynaamisuuden määrä. (Apple n.d.c.)

Visuaalisissa seurausmenetelmissä valon muutos voi vaikuttaa vahvasti seuraamisen laatuun. Varsinkin ulkotiloissa, joissa valon määrään ja suuntaan ei voi vaikuttaa, kameraan perustuva seuraaminen voi olla haastavaa. (Bostanci ym. 2013, 94.) Puhelimeissa ja tableteissa on suhteellisen pienet kuvakennot ja huono valonottokyky. Tämän takia ympäristön täytyy olla entistä paremmin valaistua.

Väriä tunnistuksia voi tapahtua, jos tunnistettavan esineen skannaus tai referenssikuva on laadultaan huono, ei sisällä tarpeeksi piirrepiirteitä tai ympäristön valon määrä on pieni tai epäsuotuisa. Väärällä tunnistuksella tarkoitan mobiililaitteen optisen tunnistuksen epäonnistumista, jossa seurattavaa esinettä tai kuvaa ei tunnisteta ollenkaan tai se tunnistetaan väärin. (Apple n.d.c.)

Väärät tunnistukset voivat johtaa tilanteisiin, joissa täydennetyyn todellisuuden sisältö pyörähtää kallelleen tai alkaa siirtymään hiljalleen johonkin suuntaan. Sisällön pyörähtäminen ympäri johtuu usein tunnistetun objektin kierron ja sijainnin väärästä tunnistuksesta. Tämäntyylisiä väriä tunnistuksia tapahtuu enemmän esinetunnistus tavassa, koska esineen tunnistukseen käytetyt pistepilvet voivat olla hyvinkin puutteellisia. Tunnistuksen taso riippuu paljolti esineen skannauksen laadusta ja tunnistettavien piirteiden määrästä. (Apple n.d.a.)

Sisällön hiljalleen siirtyminen johonkin suuntaan johtuu tunnistetun objektin visuaalisen seurauksen loppumisesta. Kun mobiililaitte menettää visuaalisen kontaktin tunnistettavaan esineeseen, se joutuu turvautumaan muihin sensoreihin liikkumisen paikantamiseksi. Mobiililaitteen muihin sensoreihin kuuluu gyroskooppi ja kiihtyvyyssanturi. Näiden sensorien tarkkuus ei ole läheskään samaa luokkaa kuin kameran optinen seuraus. (Apple n.d.a.)

Testauksessa huomasi, että ARKitillä toimivilla iOS-laitteilla väriä tunnistuksia tapahtuu harvemmin, kuin kilpailevilla täydennetyyn todellisuuden alustoilla. Tämä

voi johtua siitä ARKitin ohjelmiston laatutaso on parempi. ARCorella toimivilla Android-laitteilla immersion ylläpitäminen on paljon haastavampaa, koska ARCore ei ole vielä niin kehittynyt.

### **4.3 Okluusio**

Okluusiolla täydennetyssä todellisuudessa tarkoitetaan virtuaalisten objektien piilottamista oikeiden esineiden taakse (Furht 2011, 69). Tämä toteutetaan usein luomalla ympäristöstä kolmiulotteinen malli, jolla simuloidaan fyysistä ympäristöä virtuaalimaailmassa (Furht 2011, 69). Reaaliajassa ympäristön luominen on hankalaa, eikä useimmiten tuota parasta jälkeä. Ympäristön kolmiulotteinen malli saattaa olla epätarkka, joka johtaa immersion katoamiseen.

Alkuperäisenä tarkoituksena täydennetyt todellisuuden markkinointidemossa oli saada fyysiset esineet maailmassa vaikuttamaan digitaalisiin esineisiin. Nykyajan teknologialla mobiililaitteissa ei vielä ole mahdollisuutta tehdä reaaliaikaista ja tarpeeksi tarkkaa syvyyskarttaa dynaamisesta ympäristöstä, jolla voisi sijoittaa virtuaalisia objekteja oikeiden esineitten taakse (Neil 2018). Lopulta päädyttiin siihen, että virtuaalinen sisältö peittää kokonaan pöydän ja kaikki fyysiset esineet sen päällä.

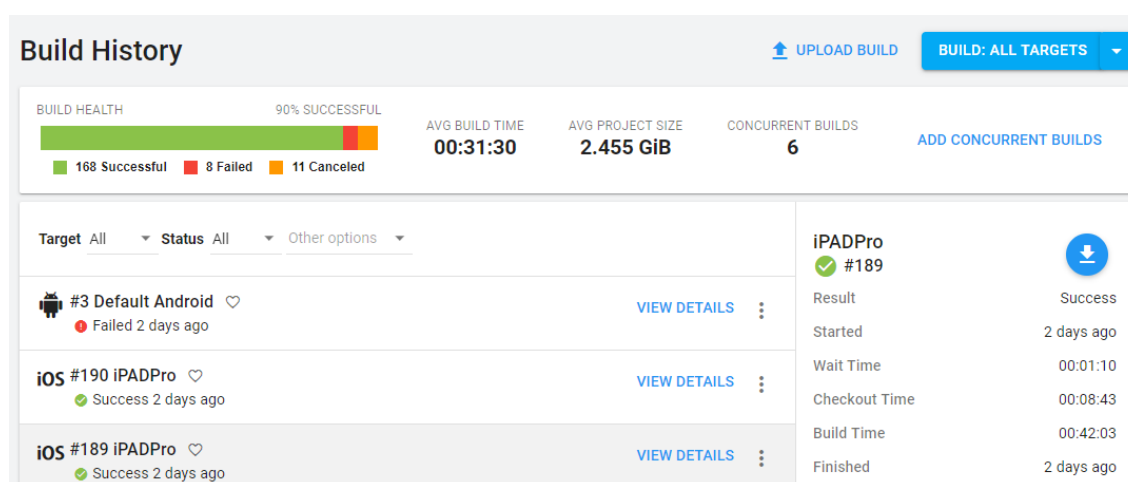
Reealiaikainen fyysisten objektien ja virtuaalisten objektien välinen okluusio on mahdollista, mutta useimmiten se on epätarkkaa tai se vaatii paljon tehoja ja erikoistyökaluja. Yksi mahdollinen vaihtoehto on värisyvyyskamera, joissa jokaista pikseliä kohden on värin lisäksi myös syvyys suhteessa kameraan. Normaalisti värisyvyyskameran kuvat sisältävät liikaa kohinaa ja ovat liian epätarkkoja, mutta viime vuosina on kehitetty tapoja hyödyntää kohinallisia kuvia ja saada päteviä tuloksia. (Walton & Steed 2017.)

### **4.4 Cloud build**

Projekti tehtiin iOS-laitteille ja käytin projektin luomiseen Windows-tietokonetta. Normaalisti ainoa tapa rakentaa sovelluksia iOS-laitteille on Mac-tietokoneella.

Kaikki iOS-sovellukset täytyy koota Xcode-ohjelmointiympäristössä, joka on käytettävissä vain Mac-tietokoneilla. Yhtenä vaihtoehtona oli pitää uusien versioiden buildausta varten yksi Mac-tietokone toimistossa, mutta lopulta päätin hyödyntää Unityn uutta Cloud Build -palvelua. (Apple n.d.b.)

Unityn Cloud Build -palvelussa projektit lähetetään verkon kautta Unityn palveluille, jossa niistä tehdään pakattuja versioita suoraan laitteille. Cloud Buildin avulla on mahdollista rakentaa projekti useammalle alustalle samanaikaisesti. Tämä vapauttaa tietokoneeni uusien versioiden rakentamisen työstä. Cloud Buildissa on ajallisesti suuri ongelma, koska yhden version rakentaminen Unityn palvelimilla voi kestää 20 minuutista yli tuntiin (KUVA 2). Android-version rakentaminen suoraan koneelta kesti maksimissaan kolme minuuttia ja keskimäärin noin minuutin. Ominaisuuksien nopeaa testausta oli lähes mahdotonta tehdä, koska jokainen testausiteraatio saattoi kestää tunnin. Tämä teki myös virheiden ja bugien etsinnästä ja korjaamisesta hyvin hankalaa. Onneksi useimmat ongelmatilanteet oli mahdollista selvittää Unity-pelimoottorin sisällä. Alla olevassa kuvassa näkyy keskimääräinen rakennusaika projektille. (Unity 2018b.)



KUVA 2. Unityn Cloud Build -palvelun käyttöliittymä

Unityn ARKit-laajennus mahdollistaa nopeampaa ominaisuuksien iteroimista pyörittämällä sovellusta suoraan Unity-editorin kautta iOS-mobiililaitteelle. Unityn ARKit-laajennuksen mukana tulee sovellus iOS-mobiililaitteelle, joka lähettää täydennetyin todellisuuden datan Unity-editoriin. Tämä ominaisuus toimii vain Mac-tietokoneilla. (Unity. n.d.c.)

## 4.5 Suorituskyky

Tarkoituksena oli tukea kaksi sukupolvea vanhoja iOS-laitteita. Sovellusta täytyi silti optimoida, vaikka tehoa Applen mobiililaitteissa on yllättävän paljon. Immersoivaa kokemusta varten tarvitaan tasainen 60 ruudunpäivitystä sekunnissa, jotta virtuaalikuva sopisi yhteen kameran videokuvan kanssa. Projektia tehdessäni minulla oli käytössä kaksi Ipad pro -mallia, ensimmäinen oli vuoden 2017 malli ja toinen oli vuoden 2018 malli.

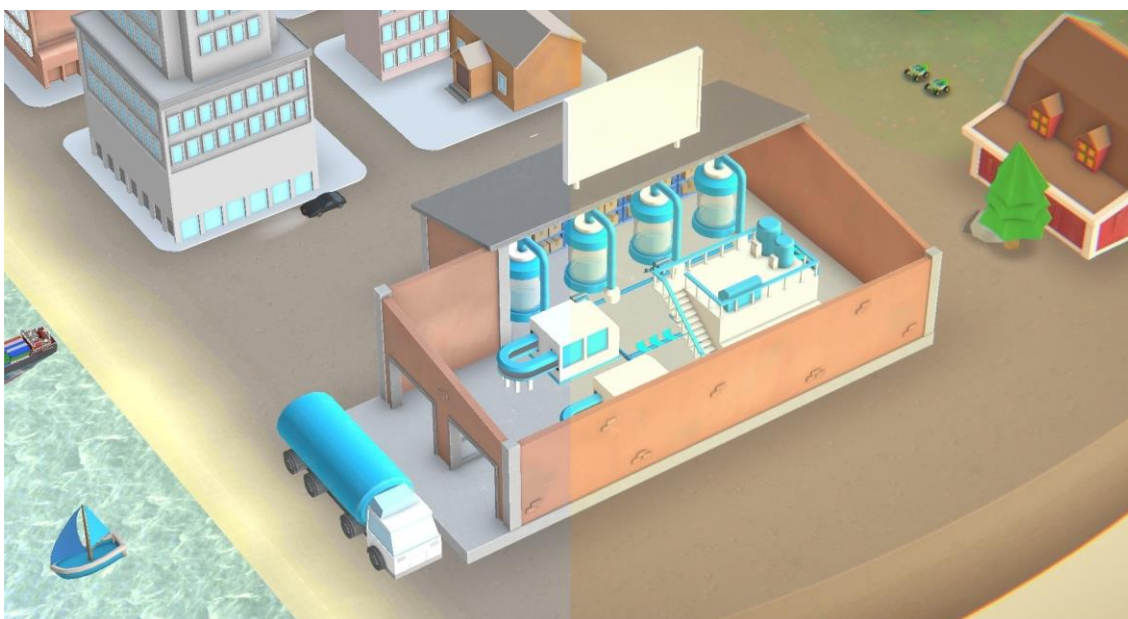
Prossessoritehoa tarvitaan myös täydennetyssä todellisuuden seurannan ylläpitämiseen. Applen ARKit luo ympäristöstä pistepilven prosessoimalla takakamerasta saatua kaksiulotteista kuvaa yhdessä laitteen liiketunnistuksen kanssa. Pistepilven avulla laite pystyy päättämään sijaintinsa suhteessa maailmaan ja virtuaaliseen sisältöön. Pistepilven ylläpito ja ympäristön seuranta vie huomattavan määrän laskentatehoa ja akkua, joka on pois sisällön tuottamiselta. (Apple n.d.a.)

Yksi yleinen tapa optimoida sovellusta, joka käyttää 3d-malleja, on rajoittamalla polygonien määrää näytöllä olevissa 3d-malleissa. Suurin osa projektia varten tehdyistä 3d-malleista oli polygonimäärältään alle tuhannen luokkaa. Unityssa kaikki alle 300 verteksin 3d-mallit piirretään samassa joukossa, joka säästää tehoa. Samoin kaikki staattiseksi ilmoitetut 3d-mallit piirretään samassa joukossa (Unity. 2018e). Käyttämällä näitä optimointimenetelmiä pystyttiin näyttämään satoja 3d-malleja ruudulla saman aikaisesti ja samalla ylläpitämään tarpeeksi nopea ruudunpäivitys. (Unity. 2018a.)

Realistisen näköinen ympäristö olisi ollut lähes mahdotonta toteuttaa siinä skaalassa mobiililaitteille, joten realismin sijaan päädyttiin käyttämään tyylielämmän näköistä grafiikkaa. Tekstuureina käytettiin resoluutioltaan pieniä tekstuureita, joissa oli yksinkertaisia pastelliväriyhdistelmiä. Sävyttimenä käytettiin lähes kaikkiin projektin 3d-malleihin Unityn omia mobiilisävyttimiä, jotka on optimoitu toimimaan mobiililaitteilla. Erilaisten sävyttimien määrä projektissa

vaikuttaa näytönohjaimen piirtonopeuteen. Säästettiin tehoja käyttämällä enimmäkseen Unityn omia sävyttimiä. (Unity 2018a.)

Post-processing, eli pikseleiden jälkikäsittely, lisää valmiiksi renderöityihin pikseleihin ylimääräisiä efektejä, kuten reunanpehmennystä tai terävyyssalueen muutoksia. Efekteillä pystyy myös simuloimaan fyysisen kameran ja filmin ominaisuuksia. Jotkut efektit vaativat enemmän laskentatehoja näytönohjaimelta, kuin toiset. Efektit kuten, reunanpehmennys, ympäristön okklusio ja liike-epäterävöinti, ovat laskentatehoiltaan hyvin vaativia, jonka takia niitä ei normaalisti mobiilisovelluksissa käytetä. Efektit kuten, väriporrastus ja vinjetti, voivat lisätä kuvan filmistä kvaliteettia hyvin pienellä suorituskyvyn menetyksellä. Kuvan jälkikäsittely parantaa sovelluksen kolmiulotteista ulkonäköä, mutta vaatii paljon tehoja toimiakseen (KUVA 3). Projektissa tehojen niukkuuden takia käytettiin yksinkertaisia efektejä. (Unity 2018f.)



KUVA 3. Vasen puoli on ilman kuvan jälkikäsittelyä, oikealla käytetään suurinta osaa jälkikäsittelyefekteistä

Useat dynaamiset valot ja varjot vievät paljon tehoja. Päädyttiin käyttämään projektissa staattisia ympyrän muotoisia varjoja kaikkiin objekteihin, koska mobiililaitteilla on rajalliset laskentatehot (KUVA 4). Me käytettiin projektissa vain yhtä suuntavaloa, ja ei yhtään dynaamisia varjoja. Tällä tavoin me saatiin suorituskykyä paremmaksi. (Unity 2018a.) Toinen syy miksi me päätettiin olla käyttämättä varjoja, oli immersio. Vaikka tehot olisivat riittäneet dynaamisten

varjojen käyttöön, ARKit ei sisällä mahdollisuutta arvioida ympäristön valon suuntaa. ARKit osaa vain arvioida ympäristön valon intensiteetin, joten dynaamiset varjot olisivat voineet rikkoa immersiota. (Apple n.d.b.)



KUVA 4. Projektin graafinen ilme. Varjojen ääriiviat ovat oranssit

#### 4.5.1 Partikkeliefektit

Unityn partikkeliefektien laskut suoritetaan prosessorissa, joka mahdollistaa partikkeleiden helpompaa manipulointia, mutta samalla se ei ole yhtä suorituskykyinen, kuin näytönohjaimella lasketut partikkelit. Prosessorilla lasketuissa partikkeliefekteissa voi myös käyttää fyysikkamoottoria tunnistamaan partikkeleiden ja muiden fysiikkaobjektien törmäyksiä. Suorituskyvyssä täytyy ottaa huomioon partikkelien määrä ja käytetty sävytin. (Dickinson 2017, 216.)

Partikkeleiden määrällä on suurin yksittäinen vaikutus partikkeliefektin suorituskykyyn. Yli 10000 partikkelin samanaikainen renderöinti mobiililaitteella alkaa jo hidastaa merkittävästi mobiililaitteen ruudunpäivitystä. Yleensä ottaen muuttamalla kokoa ja vaihtamalla partikkeliefektien parametrejä on mahdollista saada alle sadan partikkelin partikkeliefektin näyttämään lähes samalta, kuin tuhansien partikkelien partikkeliefektin. (Dickinson 2017, 216.)

Sävyttimen valinta on toinen vaikuttava tekijä partikkeliefektin suorituskykyyn. Yksinkertaiset sävyttimet, joissa on vain yksinkertainen tekstuuri parametrinä, on nopea näytönohjaimen renderöidä. Saman sävyttimen käyttö monessa eri objektissa mahdollistaa objektien renderöimisen joukossa. Monimutkaiset sävyttimet vaativat näytönohjaimelta useampaa renderöintikertaa. (Dickinson 2017, 216.)

Joissain tilanteissa on hyvä myös harkita muita vaihtoehtoja partikkeliefektien tilalle. Tein projektia varten vesiputouksen partikkeliefektinä, mutta en saanut sitä toimimaan hyvin ja samalla näyttämään hyvältä. Joten käytin partikkeliefektin sijaan tekstuuria, joka kiertää ajan mukaan vesiputouksen 3d-mallin ympäri. Sovelluksen suorituskyky kasvoi huomattavasti, ja efektin ulkonäkö ei kärsinyt merkittävästi. (Dickinson 2017, 216.)

## **4.6 Skaalaus**

Unityssa yksi yksikkö 3d-koordinaatistossa vastaa yhtä metriä. ARKit ja ARCore käyttävät myös samaa koordinaatisto järjestelmää. Unity-pelimoottorissa yhden yksikön kokoinen kuutio on täydennetyssä todellisuudessa yhden kuutiometrin kokoinen. (Johnson 2017.)

Virtuaalisen dioraaman koko on noin 100 x 150 unityn yksikkö, joka on aivan liian suuri täydennetylle todellisuudelle. Dioraamaa täytyy skaalata pienemmäksi, että se mahtuu alle neliömetrin kokoisen pöydän päälle. Skaalaus vaikutti hyvin yksinkertaiselta ongelmalta, mutta se osoittautui paljon odotettua haastavammaksi, sillä objektien skaalaaminen liian pieneksi johtaa ongelmiin muiden Unityn komponenttien kanssa. (Johnson 2017.)

### **4.6.1 Universaali skaala**

Luonnollinen vaihtoehto olisi skaalata kaikkea pienemmäksi. Jos pitäisi saada kahden neliökilometrin kokoisen alueen yhden neliömetrin kokoisen pöydän päälle sovitettua, niin universaalien skaalan pitäisi olla 2000 kertaa pienempi eli

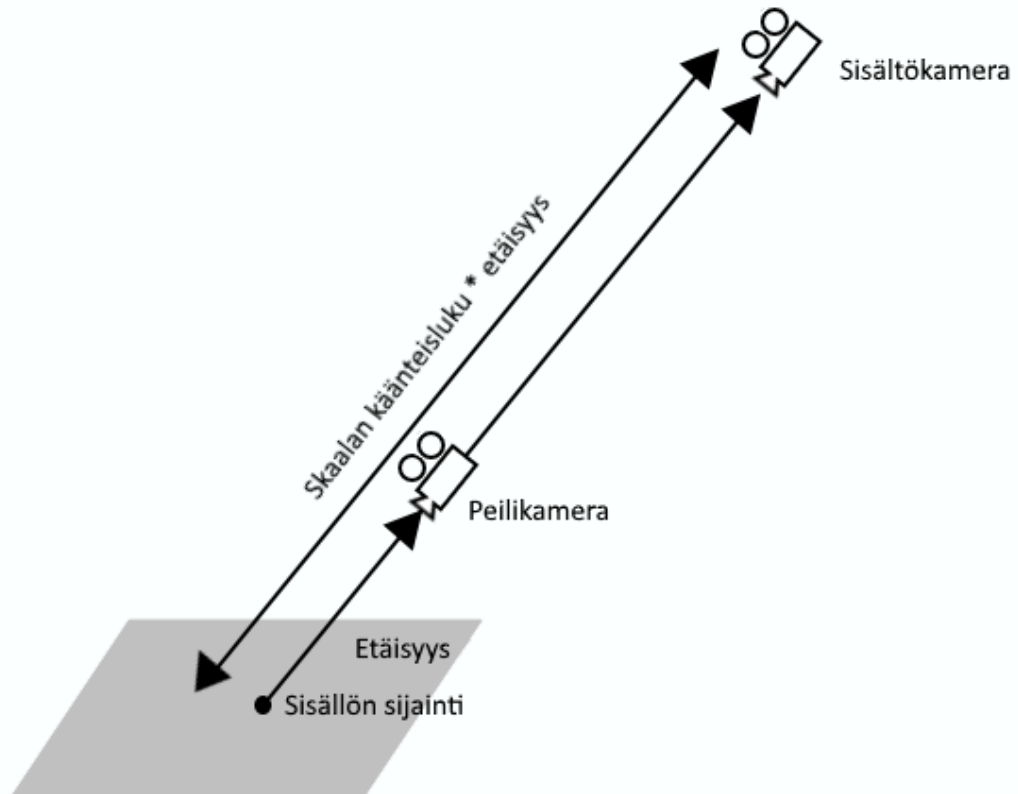
0,0005 kertainen. Skaalaus ei ole mikään ongelma, jos projektissa on vain 3d-malleja, koska niitä pystyy skaalaamaan yhtenevästi. Ongelmia tulee vastaan, kun projektissa on komponentteja, joihin skaalaus ei vaikuta ollenkaan tai joihin skaalaus vaikuttaa odottamattomalla tavalla. Fysiikkamoottori on kalibroitu toimimaan normaalikokoisissa objekteissa. Hyvin pienissä tai suurissa objekteissa fysiikat toimivat realismin vastaisesti. Myöskään partikkeliefektit, Unityn Terrain maailmanluonti työkalu ja Unityn navmesh polunetsintäjärjestelmä eivät tue skaalausta. Tuotoksessa käytin paljon partikkeliefektejä ja Unityn navmeshiä, joten universaali skaalaus ei ollut ideaali vaihtoehto. (Johnson 2017.)

#### **4.6.2 Kaksi kameraa**

Kahden kameran ratkaisussa ensimmäisellä kameralla seurataan oikean maailman objekteja. Ensimmäistä kameraa voidaan kutsua nimellä peilikamera. Peilikamera liikkuu ja kääntyy samalla tavalla, kuin mobiililaitteen fyysinen kamera. Peilikamera myös renderöi kaikki ARKitin generoimat objektit, kuten tilapäiset seuraus merkit ja tasot. (Johnson 2017.)

Toinen kamera on sisältökamera, se renderöi projektin sisällön, eli kaikki 3d-mallit, partikkelit ja muut objektit, jotka ovat osa näytettävää kokonaisuutta. Ensin tarvitaan sisällön ankkurin, joka kertoo oikean maailman koordinaatistossa sisällön sijainnin. Tämän ankkurin sijainnista vähennetään peilikameran sijainti, että saadaan niiden etäisyys toisistaan. Etäisyys-vektori kerrotaan halutun skaalan käänteisluvulla. Meidän tapauksessamme dioraaman skaala halutaan noin 0.005 kertaiseksi, joten etäisyys kerrotaan 200. Sisältökameran sijainti siirretään etäisyys-vektorin sijaintiin, orientaatio pysyy silti samana, mutta sisällön skaala näyttää pienemmältä kuin se todellisuudessa on. Tämän ratkaisun hyvä puoli on se, että sisällölle ei tarvitse tehdä mitään, vaan kaikki tapahtuu sisältökameran siirtelyn kautta. (Kuva 5.)





KUVA 5. Kahden kameran ratkaisun esittely (Johnson 2017, muokattu)

#### 4.6.3 ARSessionOrigin

Kolmas ratkaisu on Unityn kehittämä `ARSessionOrigin` luokka, jonka tarkoituksena on muuntaa täydennetyt todellisuuden koordinaatit Unityn koordinaatteihin. `ARSessionOrigin` on alun perin luotu AR Foundation rajapintaa varten, mutta se toimii myös muokattuna ARKitin ja ARCoren Unity toteutuksissa. `ARSessionOrigin` sisältää funktioita, joiden avulla voi siirtää, kääntää ja skaalata `ARSessionOrigin`in sijaintia, joka muuttaa kameran sijaintia näyttämään sisällön oikeassa paikassa (KUVA 6). Projektin tuotoksen viimeisimmässä versiossa käytin muokattua versiota Unityn `ARSessionOrigin` luokasta hoitamaan koordinaatistomuunnokset ja skaalauksen. (Unity n.d.b.)

```

public void MakeContentAppearAt(Transform content, Vector3 position)
{
    if (content == null)
        throw new ArgumentNullException("content");

    _contentOffsetTransform.localPosition =
        transform.localPosition - position - _offsetPosition;

    transform.position = content.position;

    _latestTargetPosition = position;
    _latestContentTransform = content;
}

```

KUVA 6. Muokatun ARSessionOrigin luokan sisällön siirto ja kääntö funktio

#### 4.7 Käyttöliittymä

Projektin alkuvaiheista asti tuoteomistaja toivoi käyttöliittymästä mahdollisimman yksinkertaisen ja selkeän. Mobiililaitteen ruudulla ei pitäisi olla mitään muuta, kuin kameran tuottama video ja sen päällä täydennetyyn todellisuuden lisäämä sisältö. Täytyi silti olla mahdollisuus asettaa sovellus alkupisteeseen käynnistämättä sovellusta uudelleen. Tähän tarkoitukseen hyödynnettiin piiloitettua debuggaus menu, jonka sai päälle piirtämällä mobiililaitteen näytölle sormella ympyrän. Tällä debuggaus menulla onnistuttiin välttämään sovelluksen uudelleenkäynnistämisen, ja se myös osoittautui projektin kehitysvaiheessa hyvin hyödylliseksi työkaluksi.

#### 4.8 Käyttäjän vuorovaikutus

Projektin viimeisillä hetkillä päätettiin lisätä sovellukseen tapoja käyttäjän vaikuttaa täydennettyyn todellisuuteen. Ensimmäisenä tuli mieleen lisätä kosketuspohjainen tapa vaikuttaa ympäristöön. Useimmissa mobiililaitteissa on suuret kosketusnäytöt, joiden avulla on mahdollista vaikuttaa ruudulla näkyvään sisältöön. Interaktiivinen dioraama on mielestäni paljon mielenkiintoisempi, kuin vain animoitujen 3d-mallien katsominen.



KUVA 7. Uusin versio tuotoksesta iPad Pro -tabletilla

Unityssa on mahdollista lähettää näytön kosketusta kohdasta fysiikka raycastejä kameran suuntaisesti. Raycast on säde, joka tiedottaa ensimmäisen osumansa fysiikkaobjektin tiedot ja sijainnin osumiskohdasta. Tämä on mahdollista täydennetyssä todellisuudessa, koska tässä vaikutetaan vain Unityn sisällä oleviin virtuaalisiin objekteihin. (Unity 2018d.) Raycastejä hyödynnettiin luomaan yksinkertaista interaktiivisuutta sovellukseen, kuten laivastokenaariossa laivan liikuttamiseen ja kyberhyökkäysskenaariossa virtuaalisten tietokonevirusten ja haittaohjelmien tuhoamiseen (KUVA 7).

## 5 POHDINTA

Opinnäytetyössä kuvaillaan täydennetyin todellisuuden historiaa mobiililaitteissa ja sitä, miten täydennetyin todellisuuden teknologia on kehittynyt ajan saatossa. Opinnäytetyössä käydään pikaisesti läpi kaikki yhdistetyn todellisuuden välimuodot Paul Milgramin ja Fumio Kishinon (1994) alkuperäisen kuvauksen mukaan. Opinnäytetyössä tutkitaan täydennetyin todellisuuden markkinointidemon luomista mobiililaitteelle. Selvitin parhaat työkalut ja teknologiat sovellusta varten, jotka täyttivät vaatimukseni olemalla teknologisesti kärjessä ja mahdollisimman laadukkaita. Parhaat vaihtoehdot tässä tapauksessa olivat Unity ja ARKit.

Opinnäytetyössä selvitin parhaat tavat ratkaista haasteita, joita tuli vastaan sovelluksen kehityksessä. Suurin osa haasteista liittyi sovelluksen optimointiin, mutta kehityksessä ilmeni myös useita odottamattomia ongelmia. Jouduin usein tekemään Kompromisseja, jotta saisin parhaimman mahdollisen sovelluksen tehtyä. Näitä kompromisseja olisi voinut jälkikäteen katsoen välttää muilla ratkaisuilla.

Valintani tehdä sovellus ainoastaan iOS-mobiililaitteille johti valintaan hyödyntää Unityn Cloud Build -palvelua, joka johti ongelmiin projektin rakentamisen kannalta. Cloud Build -palvelun projektin rakennus mobiililaitteelle kesti enintään tunnin ja keskimäärin puoli tuntia. Testausiterointi mobiililaitteella venyi niin pitkäksi, että ominaisuuksien testaus, virheiden etsintä ja korjaus kärsi.

### 5.1 Jatkokehitys

Olisi viisasta jatkokehitystä varten hankkia työkoneen rinnalle Mac-tietokone, jolla voisi hoitaa projektin rakentamisen sovellukseksi. Mac-tietokoneella olisi nopeampaa rakentaa projekti iOS-mobiililaitteelle, kuin käyttämällä Unityn Cloud Build -palvelua. Mac-tietokoneella olisi myös mahdollista testata projektia lennosta mobiililaitteella käyttämällä Unityn ARKit remote -sovellusta.

Esineen tunnistus ARKitissä on vielä varhaisessa vaiheessa kehitystä ja siitä koitui tunnistuksen kanssa enemmän ongelmia, kuin alun perin oletin. Esineen skannaus tunnistusta varten vaati tietynlaisen esineen, joka sisältää paljon selkeitä piirre-eroja. Sopivan esineen löytäminen tunnistusta varten, josta ei tule vääriä tunnistuksia, oli hyvin vaikeaa. Applen tarjoamat työkalut esineen skannausta varten eivät mielestäni olleet samalla laatutasolla, kuin ARKitin muut ominaisuudet. Jatkokehityksenä vaihtaisin esineen tunnistuksen kuvan tunnistukseen, koska kuvan tunnistus on luotettavampi ja säilyttää immersion paremmin.

Toinen vaihtoehto esineen tunnistuksen tilalle on tasojen tunnistus. Tasojen tunnistus toimii kaikissa tässä opinnäytetyössä mainituissa täydennetyin todellisuuden ohjelmistokehyksissä, joten se ei olisi rajoittava tekijä ohjelmiston kehityksen kannalta, kuten esineen tunnistus oli. Tasojen tunnistuksessa käyttäjä joutuu asettelemaan sisällön paikoilleen jonkin tunnistetun tason päälle täydennetyssä todellisuudessa. Tämän takia sovellukseen täytyy lisätä jokin käyttöliittymä sisällön siirtoa varten. Sisällön asettelu tekee myös sovelluksen alkuvaiheista käyttäjän kannalta hankalampia, kuin esineen tai kuvan tunnistuksessa, jossa tarvitsee vain osoittaa mobiililaitteen kamera tunnistettavaa kohdetta kohti ja kaikki asettuu suunnitellusti. Tason tunnistuksessa sisällön tahaton pyörähtäminen on epätodennäköisempää, kuin esineen tunnistuksessa, koska tunnistettava pinta on todennäköisesti kooltaan ja sisällöltään paljon laajempi, kuin yksittäisen esineen. Tason tunnistus parantaisi mahdollisesti myös tunnistuksen laatua.

Sovelluksen interaktiivisten mahdollisuuksien selvittäminen aloitin vasta myöhäisessä vaiheessa projektia. Olen varma, että projektille olisi voinut tuoda lisäarvoa kehittämällä interaktiivisuutta pidemmälle. Interaktiivisuudella on vielä paljon annettavaa täydennetylle todellisuudelle tulevaisuudessa. Lähitulevaisuudessa, kun on mahdollista luoda reaaliaikaisesti kopio fyysisestä maailmasta, mahdollisuus vuorovaikuttaa virtuaalisesti fyysisen ympäristön kanssa helpottuu huomattavasti. Uusimmat yhdistetyn todellisuuden laitteet lukevat ympäristöä jo hyvin tarkasti, mutta mobiililaitteissa ei olla ylletty samalle tarkkuustasolle.

Eri vaihtoehtoja kuvan jälkikäsittelyyn kokeiltiin projektissa moneen otteeseen, mutta kokeiluista huolimatta laitteiden tehot eivät riittäneet näyttävimpiin efekteihin. Unityn kauppapaikalla on vaihtoehtoisia kuvan jälkikäsittely lisäosia, joita mainostetaan kevyemmiksi ja muilla tavoilla paremmiksi, mutta ne eivät luoneet yhtä laadukkaita kuvia, kuin Unityn oma vaihtoehto. Lopulta projektissa hyödynnettiin vain yksinkertaisia efektejä, kuten värinmuutosta ja vinjettiefektiä.

## **5.2 Täydennetyin todellisuuden tulevaisuus**

Täydennetty todellisuus on vielä kehityskulun varhaisessa vaiheessa, sillä maailmaa mullistavia teknologian soveltamisia ei ole vielä kehitetty, mutta tulevaisuudessa teknologisten ja ohjelmiston kehitysten myötä tilanne saattaa olla aivan toinen. Tällä hetkellä uusimmat mobiililaitteet ovat tarpeeksi tehokkaita ja sisältävät kaikki tarvittavat sensorit pyörittämään hyvinkin vaativia täydennetyin todellisuuden sovelluksia. Mobiililaitteiden täydennetyin todellisuuden sovellusten kehitys riippuu paljolti ohjelmistojen kehittymisestä ja sopivien käyttömahdollisuuksien löytymisestä.

Markkinointi on yksi alue, josta on löytynyt runsaasti käyttömahdollisuuksia täydennetyille todellisuudelle. Täydennetyin todellisuuden uutuutta on helppo hyödyntää tuotteiden markkinoinnissa. Täydennetty todellisuus on hyvä markkinointikeino Intopalo Digitalille, koska Intopalo Digitalilta löytyy osaamista yhdistetyin todellisuuden luomiselle, ja se esittää hyvin Intopalo Digitalin asemaa teknologisesti tietoisesta yrityksestä. Markkinointi on hyvin vahvana osana täydennetyin todellisuuden hyödyntämistä tulevaisuudessa.

Opinnäytetyön perusteella voidaan todeta, että mobiililaitteet ovat ideaalivaihtoehto täydennetyin todellisuuden sovelluksille. Täydennetyin todellisuuden sovellusten luominen mobiililaitteille on tällä hetkellä hyvin mahdollistettua, kiitos vahvojen työkalujen ja kehitysalustojen vaihtoehtojen määrän. Tällä hetkellä suurimmat esteet täydennetyin todellisuuden kehityksen kannalta ovat teknologian saatavuus ja kehittäjien vähäinen määrä. Applen ja Googlen kilpailu mobiililaitteiden täydennetyin todellisuuden kehityksestä on hyödyllinen koko täydennetyin todellisuuden kehityksen kannalta. Apple ja

Google ovat oikealla polulla tuomassa täydennettyä todellisuutta satojen miljoonien ihmisten saataville heidän alustojen kautta. Jos jätetään mobiililaitteet pois, niin uusimmat ja kehittyneimmät yhdistetyn todellisuuden laitteistot ovat hyvin kalliita ja osaa niistä ei edes saa ostettua yksityiskäyttöön. Mobiililaitteet ovat tapa tuoda täydennetty todellisuus massoille.

## LÄHTEET

Adams, D. 2014. No Limits – Unity in Cross Industry Development. Luettu 13.3.2019 <https://blogs.unity3d.com/2014/06/05/no-limits-unity-in-cross-industry-development/>

Apple. n.d.a. ARKit dokumentaatio. Luettu 12.3.2019 <https://developer.apple.com/documentation/arkit>

Apple. n.d.b. Xcode etusivu. Luettu 30.3.2019 <https://developer.apple.com/xcode/>

Apple. n.d.c. Understanding World Tracking in ARKit. Luettu 30.3.2019 [https://developer.apple.com/documentation/arkit/understanding\\_world\\_tracking\\_in\\_arkit](https://developer.apple.com/documentation/arkit/understanding_world_tracking_in_arkit)

Bostanaci, E. Kanwal, N. Ehsan, S. Clark, A. 2013. User Tracking Methods for Augmented Reality. International Journal of Computer Theory and Engineering, Vol. 5, No. 1, February 2013.

Dickinson, C. 2017. Unity 2017 Game Optimization: Optimize all aspects of Unity performance. Birmingham: Packt.

Furht, B (ed.) 2011. Handbook of Augmented Reality. New York: Springer.

Google. n.d.a. ARCore overview. Dokumentaatio ARCoresta. Luettu 11.3.2019. <https://developers.google.com/ar/discover/>

Google. n.d.b. Distribution dashboard. Dokumentaatio Android versioista ja niiden käytöstä. Luettu 11.3.2019. <https://developer.android.com/about/dashboards/index.html>

Google. n.d.c. ARCore supported devices. Lista mobiililaitteista, jotka tukee ARCorea. Luettu 19.3.2019. <https://developers.google.com/ar/discover/supported-devices>

Google. n.d.d. Maininta kehittäjältä ARCoren seurannan päivitysnopeudesta. Luettu 13.4.2019 <https://github.com/google-ar/arcore-android-sdk/issues/604>

Johnson, R. 2017. Dealing with scale in ar. Luettu 5.3.2019. <https://blogs.unity3d.com/2017/11/16/dealing-with-scale-in-ar/>

Lojiens, W.S L. 2017. Augmented reality for food marketers and consumers. Wageningen: Wageningen Academic Publishers.

Matthew Neil. 2018. Why is occlusion in augmented reality so hard? Luettu 13.2.2019. <https://hackernoon.com/why-is-occlusion-in-augmented-reality-so-hard-7bc8041607f9>



Microsoft. n.d. What is mixed reality? Dokumentaatio. Luettu 7.3.2019.  
<https://docs.microsoft.com/en-us/windows/mixed-reality/mixed-reality>

Milgram, P. Kishino, F. A 1994. TAXONOMY OF MIXED REALITY VISUAL DISPLAYS. IEICE Transactions on Information Systems, Vol E77-D, No.12 December 1994.

Miller, D. Mowrer, T. Weiers, B. 2018 Unity's Handheld AR Ecosystem: AR Foundation, ARCore and ARKit. Luettu 6.3.2019.  
<https://blogs.unity3d.com/2018/12/18/unitys-handheld-ar-ecosystem-ar-foundation-arcore-and-arkit/>

Mitchell, R. 2017. Understanding the Differences Between ARM and x86 Processing Cores. Luettu 11.3.2019.  
<https://www.allaboutcircuits.com/news/understanding-the-differences-between-arm-and-x86-cores/>

Unity. 2018a. Practical guide to optimization for mobiles. Dokumentaatio Unity 2018.3 versiolle. Luettu 11.3.2019.  
<https://docs.unity3d.com/Manual/MobileOptimizationPracticalGuide.html>

Unity. 2018b. Unity Cloud Build. Dokumentaatio Unity 2018.3 versiolle. Luettu 12.3.2019. <https://docs.unity3d.com/Manual/UnityCloudBuild.html>

Unity. 2018c. Vuforia sdk overview. Dokumentaatio Unity 2018.3 versiolle. Luettu 18.3.2019. <https://docs.unity3d.com/Manual/vuforia-sdk-overview.html>

Unity. 2018d. Raycasting. Dokumentaatio Unity 2018.3 versiolle. Luettu 27.3.2019. <https://docs.unity3d.com/Manual/CameraRays.html>

Unity. 2018e. Draw call batching. Dokumentaatio Unity 2018.3 versiolle. Luettu 27.3.2019. <https://docs.unity3d.com/Manual/DrawCallBatching.html>

Unity. 2018f. Post-processing. Dokumentaatio Unity 2018.3 versiolle. Luettu 27.3.2019. <https://docs.unity3d.com/Manual/PostProcessingOverview.html>

Unity. n.d.a. Tietoja Unity-pelimoottorista. Luettu 13.3.2019.  
<https://unity3d.com/unity>

Unity. n.d.b. Dokumentaatio ARSessionOrigin luokasta. Luettu 27.3.2019.  
<https://docs.unity3d.com/Packages/com.unity.xr.foundation@1.0/api/UnityEngine.XR.ARFoundation.ARSessionOrigin.html>

Unity. n.d.c. ARKit laajennus Unity-pelimoottorille. Luettu 5.4.2019.  
<https://bitbucket.org/Unity-Technologies/unity-arkit-plugin>

Van Krevelen, R. 2007. Augmented Reality: Technologies, Applications, and Limitations. Amsterdam: VU University Amsterdam

Walton, D. Steed, A. 2017. Accurate real-time occlusion for mixed reality. VRST '17 Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology Article No. 11. New York: ACM