

AR-ALUSTAT UNITY-PELIMOOTTORILLE



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeenlinnan korkeakoulukeskus, tietojenkäsittelyn koulutus

Syksy, 2019

Santtu Lamminpää

Tietojenkäsittely
HAMK Visamäki

Tekijä	Santtu Lamminpää	Vuosi 2019
Työn nimi	AR-alustat Unity-pelimoottorille	
Työn ohjaaja/t	Tommi Lahti	

TIIVISTELMÄ

Opinnäytetyön tarkoituksena oli lisätä HAMK Smart -tutkimusyksikön tietämystä lisätyn todellisuuden työkaluista, joita voidaan käyttää Unity-pelimoottorilla. Opinnäytetyön tavoitteena oli testata kappaleen tunnistusta valituilla AR-työkalulla.

Opinnäytetyö oli toiminnallinen työ, jossa suunniteltiin taulukot valittujen AR-alustojen ominaisuuksien vertailuun. Taulukoista saatujen tietojen perusteella valittiin kaksi työkalua, joilla testattiin kappaleen tunnistusta hissin yläkiinnikkeille. Teoriaosuudessa lukija saa käsityksen siitä mitä ovat pelimoottorit. Erityisesti tarkastelun kohteena unity pelimoottori ennen käytännön osuudessa vertailtuja AR-alustoja. Vertailuun valitsin viisi AR-työkalua, joista dokumentoitiin alustojen perustoiminnot verkkolähteistä saaduilla tiedoilla.

Opinnäytetyössä tehtiin kappaleentunnistussovellus kahdella eri Augmented Reality -alustalla, jotka valittiin vertailutaulukosta saatujen tietojen perusteella. Sovelluksen tekeminen onnistui molemmille alustoille, mutta ei täysin niin kuin oli suunniteltu.

Opinnäytetyön aikana opin paljon asioita AR-alustojen käyttämisestä toiminnoista ja opin käyttämään unity pelimoottoria paremmin, jota olin käyttänyt vain muutamaan otteeseen ennen opinnäytetyön aloittamista. Toinen alustoista ei onnistunut tunnistamaan haluttuja kappaleita, joten tunnistusta testattiin muilla kappaleilla.

Avainsanat Unity, lisätty todellisuus, kappaleen tunnistus, pelimoottori

Sivut 37 sivua, joista liitteitä 2 sivua

Business Information Technology
HAMK Visamäki

Author	Santtu Lamminpää	Year 2019
Subject	Augmented Reality platforms for Unity game engine	
Supervisors	Tommi Lahti	

ABSTRACT

In this thesis Augmented Reality (later AR) tools were explored with the Unity Game engine. The commissioner of the work was HAMK Smart research unity. The actual goal of the thesis work was to test the object recognition functionalities with the selected AR tools.

In the functional part of the thesis there were designed several kinds of tables in which different features of AR platforms can be compared. With the data obtained from the tables, two AR platforms were selected to be tested with the object recognition feature.

In the theoretical part, reader will gain an understanding about augmented reality and game engines, how they work and where they can be used. Finally, five AR platforms were selected for comparison. Their basic functions are documented, based on the information which is obtained from online sources. Most of the sources which are used in this thesis are based on online articles, blogs and technical documentation written mainly in English, since there is no literature in Finnish available of the subject.

As a result of this thesis work, an object recognition application was developed with Vuforia and MAXST platforms. It was also found that object recognition worked with both platforms but not exactly as planned.

During the research process, a lot of useful information was learned and found out about Augmented Reality platforms and how to use them. Also, the knowledge of Unity game engine was increased as well as the research and planning skills which are essential and important in expert work.

Keywords Unity, Augmented Reality, Object recognition, Game engine

Pages 37 pages including appendices 2 pages

Käsitteet

AR	Augmented Reality, lisätty todellisuus
API	Application Programming Interface, ohjelmointirajapinta
C#	Ohjelmointikieli
GPS	Global Positioning System
GUI	Graphical User Interface, graafinen käyttöliittymä
JDK	Java Development Kit
SDK	Software Development Kit
SLAM	Simultaneous Localization and Mapping
VISLAM	Visual-Inertial Simultaneous Localization and Mapping
Unity	Pelimoottori
URL	Uniform Resource Identifier
UWP	Universal Windows Platform

SISÄLLYS

1	JOHDANTO.....	1
2	PELIMOOTTORI	2
2.1	Unity-pelimoottori	2
2.2	Unityn historia.....	3
2.3	Unityn ominaisuudet.....	4
2.3.1	Grafiikka.....	4
2.3.2	Ääni.....	4
2.3.3	Verkko-ominaisuudet	4
2.3.4	Fysiikka.....	4
2.3.5	Skriptaus	5
2.3.6	Graafinen käyttöliittymä (GUI).....	5
3	AUGMENTED REALITY (LISÄTTY TODELLISUUS)	6
3.1	Seuranta- ja tunnistusmenetelmät	6
3.1.1	Merkkipohjainen tunnistus	6
3.1.2	Merkitön tunnistus	7
3.1.3	SLAM.....	7
3.2	Käyttökohteet.....	8
3.2.1	Läketiede	8
3.2.2	Mainonta	8
3.2.3	Autoteollisuus.....	8
3.2.4	Opetus	9
4	AR-ALUSTAT UNITYLLE.....	10
4.1	Vuforia.....	10
4.2	Wikitude	12
4.3	MAXST	13
4.4	Kudan.....	14
4.5	EasyAR	15
4.6	Vertailu	16
4.6.1	Merkkipohjainen tunnistus	16
4.6.2	Merkitön tunnistus	17
4.6.3	Pilvipohjainen tunnistus	18
4.6.4	3D -objektin tunnistus	18
4.6.5	Vertailun ulkopuolelle jääneet AR-alustat	19
5	KAPPALEEN TUNNISTUS VALITUILLA ALUSTOILLA	20
5.1	Vuforia.....	20
5.1.1	Kappaleen skannaus ja tietokannan luominen	21
5.1.2	Sovelluksen rakenne ja testaus	23
5.2	MAXST	25
5.2.1	Kappaleen skannaus	26

5.2.2 Sovelluksen rakenne ja testaus	28
6 TULOKSET	32
7 YHTEENVETO	34
LÄHTEET	35

1 JOHDANTO

Lisätty todellisuus on viime vuosina kehittynyt huomattavasti ja sen tarjoamat mahdollisuudet ovat saaneet yritysten huomion. Suurin hyöty lisäystä todellisuudesta saadaan älypuhelimilla, sillä erilaisia AR sovelluksia löytyy paljon.

Tässä opinnäytetyössä tutkin Unity -pelimoottorilla toimivia lisätyn todellisuuden alustoja ja vertailen niiden ominaisuuksia. Tutkin myös Unityllä Augmented Realityn sovelluskehityksen eroja iOS & Android käyttöjärjestelmille.

Toisessa luvussa kerron lyhyesti pelimoottoreista, jotta lukijalla olisi käsitys siitä, mikä on pelimoottori ja miksi se on pelikehittäjän yksi tärkeimpiä työkaluja.

Seuraavaksi siirryn kertomaan Unity-pelimoottorista, jota käytän työkaluna tässä opinnäytetyössä. Kerron hieman Unityn historiasta ja pelimoottorin komponenteista.

Neljännessä luvussa käsitellään Augmented Reality teknologiaa ja käydään läpi järjestelmien käyttämät tunnistus ja seuraus menetelmät. Lisättyä todellisuutta hyödynnetään useilla eri aloilla, joista annan muutaman esimerkin.

Luvussa viisi tutkin erilaisia lisätyn todellisuuden työkaluja Unityllä ja dokumentoin niiden toimintoja olemassa olevan datan perusteella. Suunnitelen vertailutaulukon alustoille, jossa vertaillaan sisällön näyttämistä, pilvipohjaista tunnistusta ja 3d tunnistukseen liittyviä rajoitteita. Vertailutaulukosta saadun tiedon perusteella valitaan kaksi alustaa, joilla testataan kohteen tunnistus toimeksiantajan antamilla objekteilla.

Opinnäytetyön tavoitteena on vastata seuraaviin kysymyksiin:

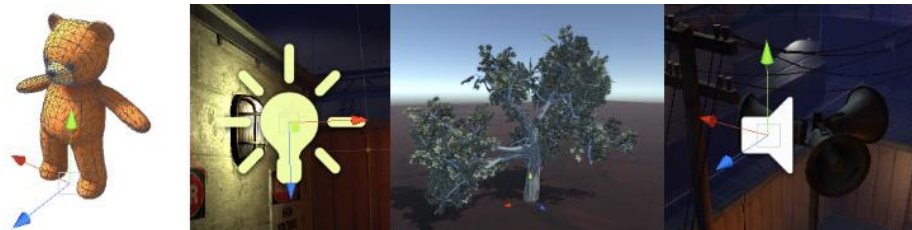
- Mitä lisätyn todellisuuden (Augmented Reality/AR) alustoja on olemassa Unity-pelimoottorille?
- Mitä eroavaisuuksia AR-alustoilla on?
- Mikä alusta toteuttaa kohteen tunnistuksen parhaiten?

2 PELIMOOTTORI

Pelimoottori on kehitystyökaluna käytettävä ohjelmistokehitysalusta (Software Framework), joka tarjoaa pelinkehittäjille tarpeelliset perustoiminnot nopeaan ja tehokkaaseen pelinkehitykseen. Nämä perustoiminnot ovat mm. grafiikka renderöinti, fysiikkamoottori, graafinen käyttöliittymä (GUI), äänet, komentosarjat (Scripting), tekoäly, verkko-ominaisuudet sekä animaatiot (Baker, 2016).

Pelimoottorien suurin hyöty käyttäjälle on se, että ne tarjoavat pelien rakentamiseen valmiit työkalut, jotta pelikehitys olisi mahdollisimman sulavaa. Hyvän kuvataajuuden (engl. Frames Per Second, FPS) saavuttamisesta tavallisten ominaisuusmuotojen tuomiseen pelimoottoreihin, pelimoottorit hoitavat pelikehittämisen ”raskaan työn”, joten kehittäjät voivat keskittyä pelin ilmapiiriin, tarinan ja muiden tekijöiden luomiseen, jotka ovat tärkeitä hyvän pelin luomisessa (Enger, 2013).

Pelikehyksessä pelimoottoriin kerätään kaikki tarvittavat objektit (engl. assets), objekteja ovat esimerkiksi hahmot, kerättävät esineet, valo, kamerat ja erikoisefektit. Objektit ei kuitenkaan itsenäään tee yhtään mitään, vaan niille täytyy antaa ominaisuuksia, jotta ne toteuttavat jonkin toiminnot.



Kuva 1. Erilaisia peliobjekteja: animoitu hahmo, valotus, puu ja äänilähde (Unity Manual b, n.d.).

2.1 Unity-pelimoottori

Unity on Unity Technologiesin luoma pelimoottori, joka tarjoaa työkalut 2D-, 3D-, VR- (Virtual Reality) ja AR- (Augmented Reality) pelien tehokkaaseen kehittämiseen. Pelimoottorilla on mahdollista luoda pelejä Windows, iOS, Linux ja Android käyttöjärjestelmille. Unity tukee myös Playstation4-, Xbox One-, Nintendo Switch- ja Nintendo 3DS-pelikonsoleita (Unity3d, 2018).

Pelimoottorin ohjelmointikielenä toimii C#-ohjelmointikieli, ennen C#:n lisäksi ohjelmointikielinä toimi myös BOO sekä UnityScript, joka oli Unityn versio JavaScriptistä. UnityScripting käyttö lopetettiin vuonna 2017 ja BOO-ohjelmointikielen tuki vuonna 2014 (Fine, 2017). Taulukossa 1 on listattu minimi vaatimukset sovelluskehitykseen.

Taulukko 1. Unityn järjestelmävaatimukset sovelluskehitykseen (Unity b, n.d.).

Käyttöjärjestelmä	Vaatimukset
Windows	Windows 7 SP1+, 8, 10 (vain 64-bit versiot) CPU: SSE2 tuki GPU: Näytönohjain DX10 tuella
iOS	Mac tietokone, jossa minimissään macOS 10.12.6 ja Xcode 9.0 tai uudempi
Android	Android SDK ja JDK
UWP	Windows 10 (64-bit), Visual Studio 2015

Unity-editorista on saatavilla ilmainen lisenssi, joka on mainio vaihtoehto harjoitteluun ja harrastus käyttöön. Kaikki pelimoottorin ominaisuudet ovat käytössä ilmaisessa versiossa. Tässä opinnäytetyössä käytän Unityn Personal lisenssiä ja viimeisin versiopäivitys on 2018.3.

2.2 Unityn historia

Tanskalainen Nicholas Francis pyysi vuonna 2002 apua Mac OpenGL-foorumilla varjostus järjestelmän käyttöönottoon omassa pelimoottorissaan, johon hän taas sai apua saksalaiselta Joachim Antelta. Tämän jälkeen kaksikko päätyi tekemään varjostusjärjestelmää yhdessä, sillä molemmat hyötyisivät siitä. He kuitenkin päättivät alkaa työskennellä saman projektin parissa. Myöhemmin tiimiin liittyi Tanskalainen David Helagson ja kolmikko perusti Unity Technologies yrityksen vuonna 2004 (Haas 2014, 4).

Aluksi yrityksen tarkoituksena oli myydä pelimoottoritekniologiaa muille kehittäjille käytettäväksi, mutta he päättivätkin kehittää pelimoottorin, joka on tarkoitettu nettipelien luomiseen. Esitelläkseen Unity Technologiesin pelimoottoria, he kehittivät pelin "GooBall", joka julkaistiin vuonna 2005. Pian sen jälkeen julkaistiin Unityn ensimmäinen versio 1.0 Mac OS X käyttöjärjestelmälle (Haas 2014, 5-8).

Versiossa 1.1, Unity lisäsi tuen pelien paketoimiseen Windows koneille ja internet-selaimille. Unityn 2.0 versio oli yrityksen suurin päivitys sen ominaisuuksien kannalta, joka julkaistiin vuonna 2007. Versiossa oli parempi tuki Windows-käyttöjärjestelmille ja internet-selaimille. Versioon lisättiin tuki DirectX-rajapinnalle, joka lisäsi pelien suorituskykyä huomattavasti. Unity sai vasta vuonna 2009 version Windows-koneille, jolloin koko editori piti rakentaa uudelleen (Haas 2014, 9-10).

2.3 Unityn ominaisuudet

2.3.1 Grafiikka

Grafiikka on yksi pelimoottorin tärkeimmistä elementeistä ymmärtää, jotta pelistä saadaan tehtyä näyttävän näköinen. Unity-editoriin voi tuoda eri objekteja (engl. assets), kuten 2D- ja 3D-malleja eri ohjelmista, esimerkiksi Maya, 3s Max tai Photoshopista. Kun objektit on järjestetty editorin näkymään ja ympäristöön, voidaan alkaa lisäämään valaistusta, ääntä, erikoistehosteita, fysiikka, animaatio ja pelin logiikka. Tämän jälkeen voidaan siirtyä jo sisällön testaukseen (engl. Debug) ja sisältö voidaan optimoida halutulle sovellusalustalle. (Unity Manual e, n.d.)

2.3.2 Ääni

Äänitehosteet voidaan suunnitella editorissa tyhjästä, aloittaen musiikin kokoamisesta, ääniefekteistä, ääninäyttelystä ja loppuun lisätään jälkituotanto. Unityssä voi hyödyntää Asset Storea, josta voi ostaa ääniefektejä, jotka voidaan yhdistää ja sekoittaa eri tavoin. Ääniä voidaan tuoda myös Unityn ulkopuolelta. (Unity Manual, n.d.).

2.3.3 Verkko-ominaisuudet

Unity-editorissa on kaksi erilaista tapaa luoda verkko-ominaisuuksia. Jos halutaan luoda multiplayer peli, tulisi pelikehittäjän aloittaa unityn NetworkManagerin tai High Level API:n (Application programming interface) käyttö. Network Manager on työkalu, jolla voidaan hallita multiplayer pelin verkko-ominaisuuksia. Unityn oma multiplayer High Level API(HLAPI), on järjestelmä, jolla voidaan rakentaa multiplayer sisältöä. HLAPI on joukko verkkokomentoja yhdistettynä Unityyn. Se on helppokäyttöinen kehityksen kannalta ja tarjoaa palveluita, jotka ovat hyödyllisiä multiplayer pelin luomiseen. (Unity Manual c, n.d.).

2.3.4 Fysiikka

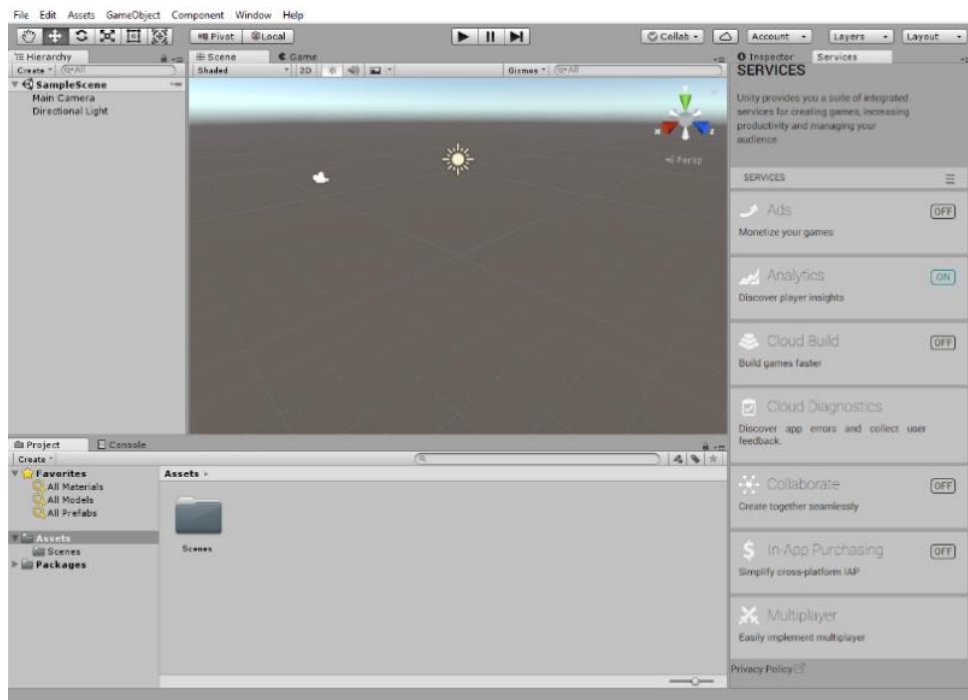
Unityn sisäänrakennetut fysiikkamoottorit tarjoavat komponentteja, jotka käsittelevät fyysistä simulointia. Muutaman parametrisetoksen avulla voidaan luoda objekteja, jotka käyttäytyvät realistisella tavalla. Koodin avulla voidaan kontrolloida fysiikkaa, kuten luoda ajoneuvoille tai koneelle sen dynamiikka. Unity käyttää kahta eri fysiikkamoottoria 3D- ja 2D fysiikan luomiseksi. Fysiikkamoottoreiden käsitteet ovat samanlaisia näiden moottoreiden välillä, mutta niiden toteutus tapahtuu käyttämällä eri komponentteja. (Unity Manual d, n.d.).

2.3.5 Skriptaus

Tärkeä osa pelimoottoreita on skriptaus. Sen avulla määritellään pelikomponenttien logiikka. Skriptaus on olennainen osa kaikkia pelejä, sillä sen avulla luetaan pelaajan tekemiä toimintoja ja järjestetään pelitapahtumat sellaisiksi, kun halutaan. Tämän lisäksi skripteillä voidaan luoda graafisia efektejä, kontrolloida objektien fyysistä käyttäytymistä tai jopa lisätä tekoälyä peliin. (Unity Manual a, n.d.).

2.3.6 Graafinen käyttöliittymä (GUI)

Useimmissa peleissä on jonkinlainen käyttöliittymä, joka sopii pelin teemaan. Tämän takia pelimoottorit antavat kehittäjille työkalut oman käyttöliittymän luomiseksi. Unityssä olevia käyttöliittymän komponentteja ovat: piirtoalue, painike, kuva, teksti, toiminnon käynnistys, siirtymiset.



Kuva 2. Unity käyttöliittymä

3 AUGMENTED REALITY (LISÄTTY TODELLISUUS)

Lisätty todellisuus tarkoittaa sitä, että tuodaan ihmisen havaitsemaan maailmaan digitaalisia elementtejä, joita käyttäjä näkee läpikatseltavien näyttöjen kautta. Elementtejä voivat olla esimerkiksi kuva, ääni, video, teksti, GPS-toiminnot ja tekoäly, joka yhdistelee edellä mainittuja elementtejä. AR toimii yleisimmin mobiililaitteilla ja tietokoneisiin ladattavilla ohjelmissa. Viime vuosina on kehitetty erilaisia laitteita, jotka tuovat lisätyn todellisuuden hyödyt uudella tavalla käyttäjälle koettavaksi, tästä hyvä esimerkki on älylasit.

AR-järjestelmä koostuu kolmesta yksinkertaisesta vaiheesta: Tunnistus, seuranta ja sekoitus. Tunnistuksen kohteena voi toimia mikä vain kuva, objekti, kasvot, vartalo tai tila, jonka päälle virtuaalista sisältöä voidaan asettaa. Kun järjestelmä on onnistuneesti tunnistanut kohteen, sen päälle lähetetään digitaalista sisältöä (Amin & Govilkar 2015, 12).

3.1 Seuranta- ja tunnistusmenetelmät

Jokaiselle AR-sovellukselle täytyy määritellä jonkinlainen logiikka etukäteen. Logiikka määrittelee, millaista digitaalista sisältöä ilmestyy näytölle, kun jokin kohde tunnistetaan. Yleisimmin AR-järjestelmät käyttävät kahta erilaista tyyliä tunnistukseen. Nämä tunnistustyyliä ovat merkkipohjainen ja merkitön tunnistus (Paladini, 2018). Nykyään AR-kehitystyökalut hyödyntävät SLAM-teknologiaa, joten merkitöntä tunnistusta hyödynnetään vähemmän. SLAM-teknologia on yleistynyt viime vuosina erityisen paljon ja siksi se onkin saanut huomiota AR- ja VR teollisuudelta. SLAM-teknologiasta kerrotaan enemmän kappaleessa 3.1.3.

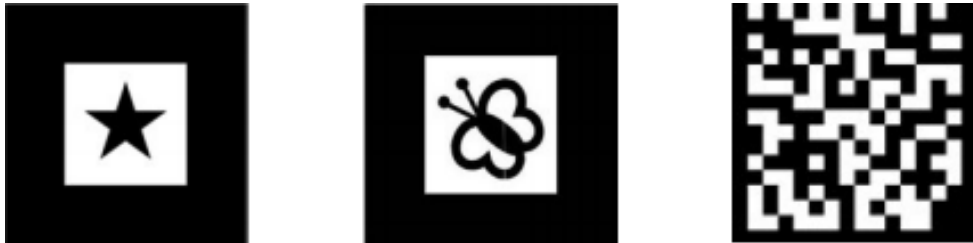
3.1.1 Merkkipohjainen tunnistus

Merkkipohjainen tunnistus (engl. marker-based Tracking), tunnistaa etukäteen määriteltyjä merkkejä fyysisestä maailmasta. Tunnistuksen avulla järjestelmä havaitsee merkin ja laskee sen koordinaatit, jonka jälkeen sisältöä ilmestyy merkin päälle. Markkereiden koko ja valaistus voi vaikuttaa tunnistuksen tarkkuuteen ja on suurin haaste AR-järjestelmille, jotka käyttävät tätä tunnistustyyliä (Amin & Govilkar 2015, 12).

Sanni Siltanen (2012) määrittelee merkkipohjaisen tunnistuksen prosessin seuraavasti:

1. Kuvan ottaminen
2. Kuvan esikäsittely
3. Potentiaalisten merkkien tunnistus ja "ei-merkkien" (engl. Non-marker) hylkäys
4. Merkkien tunnistus ja dekodaus
5. Merkkien sunnan ja asennon laskeminen

Kun kuva on otettu, se esikäsitellään poistamalla säröt ja etsimällä merk-
kien ääriviivat. Esikäsitelyn jälkeen valitaan potentiaaliset merkin tarkem-
paa tunnistusta varten ja poistetaan ei-merkit, jonka jälkeen selvät merkit
tunnistetaan ja dekodataan niiden sisältö. Viimeisenä järjestelmä laskee
merkkien sijainnin ja asennon (Siltanen 2012, 40). Kuvassa 4 muutama esi-
merkki merkkipohjaisessa tunnistuksessa käytettävistä merkeistä.



Kuva 3. Merkkipohjaisessa tunnistuksessa käytettäviä merkkejä (Silta-
nen 2012, 46).

3.1.2 Merkitön tunnistus

Merkitön (engl. Markerless) tunnistus on termi, jolla kuvataan AR-
sovellusta, joka ei tarvitse ennalta määriteltyä tietoa käyttäjän ympäris-
töstä sisällön näyttämiseksi (Schechter, 2014). Tätä tunnistustyyliä hyö-
dyntävät AR-järjestelmät käyttävät elektronisen laitteen gyroskooppia,
kompattia ja sijaintitietoja, jolloin järjestelmä saa käyttäjän tarkan sijain-
nin. Tämän datan avulla järjestelmä voi vertailla sitä tietokantaan, jolloin
sisältöä saadaan näkyville näytölle. Tässä tunnistustyyliä suurimmat
haasteet ovat ympäristön epätasaisuus, sekä kameran rajoitukset ja GPS
yhteyden vahvuus. Tunnistustapa ei toimi hyvin sisätiloissa (Amin & Govil-
kar 2015, 12-13).

3.1.3 SLAM

SLAM (engl. Simultaneous Localization And Mapping) on algoritmi, joka
kartoittaa käyttäjän ympäröimää aluetta ja seuraa kaikkea alueella tapah-
tuvaa liikettä. AR-sovellukset, jotka käyttävät tätä toimintoa muistavat fyy-
sisten objektien sijainnin suhteessa ympäristöön. Suurin osa nykyaikaisista
visuaalisista SLAM-järjestelmistä (VISLAM) perustuu pisteiden seurantaan
peräkkäisten kamerakuvien kautta. Kamerakuvien avulla järjestelmä kol-
miomittaa niiden 3D-sijainnin. Samanaikaisesti SLAM käyttää arvioituja
pistekohtia, joiden avulla lasketaan kameran asento. Tavoitteena on siis
kartoittaa ympäristö ja suhteuttaa se omaan sijaintiin, jotta navigointi olisi
mahdollisimman tarkkaa. Tämä on mahdollista yhdellä 3D-kameralla, kun-
han on olemassa riittävä määrä pisteitä, joiden avulla järjestelmä ymmär-
tää antureiden suunnan sekä ympäristön rakenteen.

Visuaaliset SLAM-järjestelmät toimivat reaaliaikaisesti, joten usein sijaintitiedot ja karttatiedot käyvät läpi nipun hienosäätöjä ennen kuin ne lopulta yhdistetään. Nämä säädökset suoritetaan erikseen sijainti- ja karttatiedoille, mutta ne tapahtuvat samanaikaisesti (Kudan, 2016).

3.2 Käyttökohteet

Lisätyn todellisuuden käyttö on saanut viime vuosina paljon suosiota, samalla kun elektronisten laitteiden koko on pienentynyt, lisätyn todellisuuden käyttö on lisääntynyt monilla eri aloilla.

3.2.1 Lääketiede

Yksi suosituimmista AR-ratkaisuista lääketieteen kehittämiseen on Google lasit. Vuonna 2013 joulukuussa Bostonin Beth Israel Deconness-terveyskeskuksessa lääkärit ottivat käyttöön Googlen älylasit testatakseen, että helpottavatko ne lääkärin ja potilaan välistä vuorovaikutusta tai tietojen syöttämistä. Terveyskeskukseen ripustettiin potilashuoneiden oviin isot QR-koodit. Lääkäri pystyy skannaaman QR-koodit astuessaan potilashuoneeseen, jolloin älylaseihin tulee näkyviin potilaan tiedot.

Ocra Health on mobiiliohjelmisto-yritys, joka on vuodesta 2010 lähtien kehittänyt monia lääketieteessä hyödynnettäviä sovelluksia. Yksi yrityksen sovelluksista on EyeDecide, jonka avulla lääkärit voivat näyttää simulaation potilaan näkökyvystä. Tämän avulla potilaalle saadaan parempi käsitys hänen terveyden tilastaan (The Medical Futurist, 2017).

3.2.2 Mainonta

AR tekee mainonnasta interaktiivista, jolloin markkinoijat ja mainostajat tavoittavat kuluttajan aivan uusilla tavoilla. Yksi markkinointia lisäävä sovellus on IKEA-place, jonka avulla käyttäjä voi esikatsella IKEA-huonekaluja omassa asunnossa. Leluyhtiöt ovat myös alkaneet hyödyntämään lisättyä todellisuutta esimerkiksi legopaketeissa, jotka kuluttaja voi skannata nähdäkseen millainen valmis rakennelma legoista rakentuu. Suomessa Valio on lisännyt maitopurkkeihinsa skannattavia merkkejä, joista pystyi katsomaan esimerkiksi Tuntematon sotilas-elokuvan trailerin.

3.2.3 Autoteollisuus

Autoteollisuudessa lisätyllä todellisuudella voidaan tehdä navigoinnista helpompaa. Tuulilaseihin on mahdollista näyttää ajo-ohjeita, nopeusmittari ja informaatiota muusta liikenteestä. Apple on patentoinut lisättyä todellisuutta käyttävän tuulilasin, jota kutsutaan nimellä ”Heads-up” näyttö.



Kuva 4. AR-järjestelmä tuulilasissa

3.2.4 Opetus

Opetuskäytössä lisättyä todellisuutta hyödyntäviä kohteita ovat esimerkiksi oppikirjat, joihin voidaan lisätä kuvia ja animaatioita, joita voidaan nähdä läpikatseltavien näyttöjen läpi. Tämä olisi kätevä tapa opettaa lapsia, sillä kuvien ja animaatioiden näkeminen tekee oppimisesta lapsille mielenkiintoisempaa ja voi nopeuttaa oppimista.

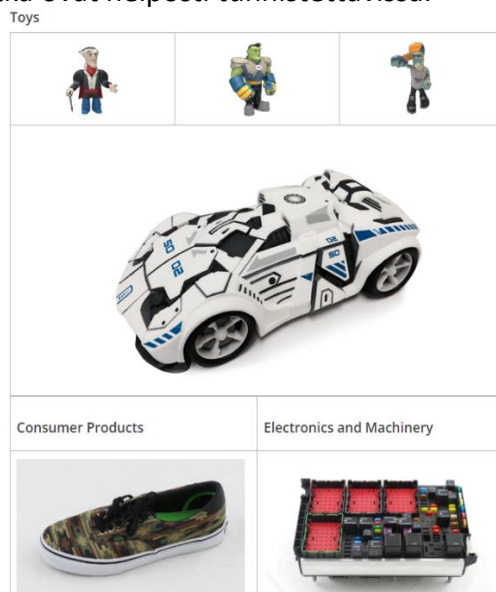
4 AR-ALUSTAT UNITYLLE

Kappaleessa tutkitaan AR-alustoja, jotka toimivat Unity-pelimoottorilla. Alustoja on saatavilla pelimoottorille hyvin paljon, joten tässä työssä ei voi käsitellä kaikkia alustoja. Kappaleessa käsitellään viittä eri alustaa ja näiden perustoimintoja. Lopuksi täytetään vertailutaulukot, jossa tutkitaan alustojen sisällönäyttämistä, merkitöntä tunnistusta, pilvipohjaisten palveluiden käyttöä ja kohteen tunnistusta.

4.1 Vuforia

Vuforia Engine on AR-kehitystyökalu, joka tukee johtavia älypuhelimia, tabletteja ja älylaseja. Alusta koostuu erilaisista komponenteista, kuten Target Management System, Cloud Target Database ja Device Target Database. Vuforia käyttää kohteen resursseja joko pilviyhteyden avulla tai suoraan älylaitteen paikallisesta tallennustilasta. Vuforia SDK tukee sekä iOS:n että Androidin alkuperäistä kehitystä ja mahdollistaa AR-sovellusten kehittämisen Unity Editorissa, joka on helposti siirrettävissä molemmille alustoille.

Kohteen tunnistus mahdollistaa monimutkaisten 3D objektien tunnistuksen ja seurannan. Tämän avulla voidaan rakentaa monipuolisia ja interaktiivisia AR-sovelluksia. Sovellusten sisältö voi olla esimerkiksi leluun lisättävää 3D-sisältöä, käyttöohjeiden lisäämistä kuluttajan elektroniseen laitteeseen tai uuden työntekijän kouluttamista työtehtäviin interaktiivisen koulutuksen avulla. Jotta kohteen tunnistus toimisi hyvin, fyysisen objektin tulisi olla läpinäkymätön, jäykkä ja siinä tulisi olla muutama liikkuva osa. Objektin pinnalla tulisi olla kontrasti pohjaiset ominaisuudet, vääntyneiden objektien käyttöä ei suositella. Kun kohdetta yritetään tunnistaa, sen ympäristön pitäisi olla kohtuullisen valoisa, jotta tunnistus olisi nopeaa ja tarkkaa. (Vuforia Development Library b, n.d.). Kuvassa 6 on esimerkkejä kohteista, jotka ovat helposti tunnistettavissa.



Kuva 5. Esimerkkejä helposti tunnistettavista kohteista (Vuforia Development Library b, n.d.).

Vuforian Cloud Recognition on palvelu, jonka avulla voidaan ylläpitää ja hallita kuvia. Tämä palvelu on saatavilla Vuforia Development, Cloud, Pro ja Enterprise-lisensseillä. Palvelu on myös helposti integroitavissa muiden sisällönhallintajärjestelmien kanssa. Käyttö määritellään kuvien tunnistusten määrällä, eli kuinka monta tunnistusta sovellus suorittaa kuukaudessa. Tämä palvelu sopii hyvin sovelluksille, jotka käyttävät useita kohteita tai kun kohteita täytyy päivittää usein. Sovellusta, joka käyttää Cloud Recognition-palvelua, voi vertailla kameran kuvia tietokantaan, jolloin sovellus pystyy suorittamaan paikallista tunnistusta ja seuranta.

Jokaiseen pilvipalvelussa olevaan kohteeseen voidaan liittää metadataa. Tämä data on käyttäjän määrittelemää tietoa, joka voidaan liittää kohteeseen. Tämä tieto voi olla esimerkiksi tekstiä tai URL-linkin kautta noudettava video. Metadata tiedoston maksimikoko on 2MB / kohde (Vuforia Development Library c, n.d.).

Vuforia Enginen tunnistus- ja seurantaominaisuuksia voidaan käyttää useissa kuvissa ja kohteissa. Vuforia Enginen viimeisin versiopäivitys mahdollistaa syvällisen oppimisen model targeteille, joiden avulla järjestelmä tunnistaa objekteja välittömästi. Engine käyttää myös VISLAM tunnistus teknologiaa, joka mahdollistaa tehokkaan merkittömän AR-tunnistuksen laitteisiin, jotka eivät alun perin tue teknologiaa. Taulukossa 2. on listattu Vuforia-moottorin tunnistus- ja seurantaominaisuudet.

Taulukko 2. Vuforia Enginen käyttämät tunnistus- ja seurantaominaisuudet (Vuforia Developer Library a, n.d.)

Toiminto	Selite
Model Targets	Kohteen tunnistus olemassa olevilla 3D-malleilla. AR sisältöä voidaan asettaa useisiin eri kohteisiin, kuten teollisuuslaitteisiin, ajoneuvoihin tai kodinkoneisiin.
Image Targets	Sisältöä voidaan kiinnittää litteisiin kuviin, kuten tulostusmateriaaleihin tai tuotepakkauksiin.
Object Targets	Luodaan skannaamalla jokin objekti. Hyvä vaihtoehto esim. leluille tai muille tuotteille, joilla on paljon yksityiskohtia.
Multi-Targets	Luodaan usealla Image Targetilla, jotka voidaan järjestää tavallisiin geometrisiin muotoihin (esim. laatikko)
Cylinder Targets	Tunnistaa kohteet, jotka ovat sylinterimäisiä (esim. juomapullot, kupit, tölkit)
VuMarks	Mukautettuja merkkejä, jotka voidaan koodata erilaisiin tietomuotoihin. Tukevat sekä yksilöllistä tunnistusta että seuranta AR-sovelluksissa.
External Camera	Ulkoinen kamera määrittelee kaikki Vuforia enginen tarvitsemat erityspiirteet ulkoisten kuvalähteiden käyttämiseksi.

Ground Plane	Mahdollistaa sisällön sijoittamisen horisontaalisille pinnoille (esim. pöytä, lattia)
--------------	---

4.2 Wikitude

Wikitudella voi luoda AR-sovelluksia, joilla voidaan tunnistaa, seurata ja lisätä kuvia, esineitä, kohtauksia, maantieteellisiä sijainteja ja paljon muuta. Wikitudella on mahdollista työskennellä natiivilla- tai JavaScript API:lla, tai käyttää Unityä, jolla voidaan luoda monipuolisia AR-kokemuksia älylaitteille, jotka käyttävät Android, iOS ja Windows alustaa.

Objektin tunnistus ja seurantatoiminto perustuu kehitysalustan SLAM -moottorin toimintaan ja sitä käytetään alustan jokaisessa tunnistus tyylissä. Objektin tunnistuksen avulla on mahdollista tunnistaa objekteja ja kokonaisia kohteita, jotka käyttäjä määrittää etukäteen. Sopivia kohteita on esimerkiksi lelut, patsaat, teollisuudessa käytettävät laitteet, työkalut ja kodinkoneet. Tunnistus toimii parhaiten, jos kohteella on rajallinen määrä liikkuvia ja kiinteitä osia. Kohteen tunnistusta voidaan käyttää myös suurempien paikkojen tai maisemien tunnistukseen, tätä toimintoa kutsutaan paikan tunnistukseksi (Wikitude Support b, n.d.).

Wikituden pilvipalvelu on suunniteltu projekteille, jotka tarvitsevat normaalia suuremman tallennustilan. Kehitystyökalu tarjoaa laitteen oman ja offline-tunnistuksen, eli sovellus voi tunnistaa jopa 1000 kuvaa ilman internet-yhteyttä. Pilvipalvelu tunnistus mahdollistaa tuhansien kuvien ylläpidon ja tarjoaa 100GB tallennustilaa.

Unity liitännäinen on rakennettu Wikituden monipuolisen natiivi API:n päälle, mutta kaikkia ominaisuuksia ei voida hyödyntää. Taulukossa 3 on toiminnot, jotka toimivat Unityllä.

Taulukko 3. Unityllä toimivat Wikituden toiminnot (Wikitude, n.d.).

Toiminto	Selite
Instant Tracking	Digitaalisen sisällön lisääminen ilman merkkien käyttämistä. Wikitudella on oma SLAM Instant Tracking -tekniikka, joka voidaan yhdistää dynaamisesti ARKitiin ja ARCoreen
Object Recognition	Reaaliaikainen 360° tunnistus, jonka kohteet määritellään etukäteen
Scene Recognition	Mahdollistaa suurien esineiden, huoneiden tai nähtävyyksien tunnistamisen, seurannan ja lisäämisen
Image Recognition	Kuvan tunnistus, Wikituden AR-kirjastoon on mahdollista tallentaa 1000 kuvaa, jotka voidaan tunnistaa offline-tilassa käyttämällä pilvipalvelua
Seamless AR Tracking	SMART vaihtaa ARCore- tai ARKit-järjestelmään jos laite tukee sitä. Jos laite ei tue näitä järjestelmiä,

	Wikitude-tekniikka käynnistyy, jolloin Instant Tracking toimintoa voidaan käyttää monilla laitteilla
--	--

4.3 MAXST

MAXST on AR-kehitystyökalu, joka on järjestelmäriippumaton ja se tarjoaa kaiken tarvittavan AR-sovellusten kehittämiseen. MAXST tukee useita kehitysympäristöjä, kuten Android, iOS (Swift, Objective-C), Mac OS, Windows ja Unity. Kehitystyökalun algoritmi on optimoitu mobiilikäyttöön, joka takaa nopean ja vahvan tunnistuksen sekä seurannan. SIMD -rinnakkaiskäsitelymenetelmä nopeuttaa toimintoja ja alentaa laitteen muistin käyttöä, jotta moottori toimii hyvin myös alemman luokan älylaitteilla, joissa on vähemmän resursseja. MAXST 4.0 versio päivityksessä lisättiin pilvipalvelun kautta tunnistaminen, tällä palvelulla ei ole rajoituksia tunnistusmääriin.

Kehitystyökalu tarjoaa erinäisiä komponentteja, jotka helpottavat AR-sovelluksen rakentamisessa, taulukossa 4 on listattuna kehitystyökaluun kuuluvat komponentit (MAXST developer b, n.d.).

Taulukko 4. MAXST kehitystyökalun komponentit

License Manager	Lisenssien luonti ja hallinta työkalu
Target Manager	Työkalu, jonka avulla hallitaan kuvatunnistuksessa käytettäviä kuvia. Target Manager on optimoitu Chrome-selaimelle. Kuvat tulee olla JPG tai PNG formaatissa, eikä ne saa olla yli 3MB suurempia.
Map Manager App	Mobiilisovellus, jonka avulla kartoitetaan 3D-kohde ja hallitaan MAP-tiedostoja VISUAL SLAM ja objektin seuraus toiminnoille.

SDK on suoraan ladattavissa Unitylle, Androidille ja iOS:lle ja sen kaikki ominaisuudet ovat käytettävissä ilmaiseksi. Taulukossa 5 on kaikki MAXST kehitystyökaluun kuuluvat toiminnot (MAXST Developer a, n.d.).

Taulukko 5. MAXST perustoiminnot.

Toiminto	Selite
Image Tracker	2D -kuvantunnistuksen avulla sovelluksen on mahdollista seurata kohteita tarkasti haastavissa olosuhteissa, kuten silloin kun kamera liikkuu nopeasti tai kun kohde näkyy vain osittain näytöllä. Kohteita voidaan tunnistaa enintään kolme samaan aikaan.
Marker Tracker	MAXST tarjoaa 8192 valmista merkkiä, jotka järjestelmä tunnistaa helpommin ja tehokkaammin. Tällä toiminnolla voidaan hyödyntää markkereita myös suurissa tiloissa.

Instant Tracker	Tasohaku etsii välittömästi vaakasuoran tason kameran näkemästä tilasta. Tämän toiminnon avulla voidaan seurata ja sijoittaa 3D -objekteja suhteessa näille tasoille. Sovellus pystyy lisäämään sisältöä ilman etukäteen määriteltyjä merkkejä.
Visual SLAM	Kehitystyökalun avaruuskartoitusmoottori (engl. Space mapping engine) analysoi RGB-kamerasta saatua dataa, poimii yksityiskohdat ja tallentaa ne MAP-tiedostoon. Näitä tiedostoja voidaan hallita Map Manager työkalun avulla.
Object Tracker	Toiminto lataa Visual SLAM luodut MAP-tiedostot, joiden avulla niistä voidaan rakentaa AR-efektejä käyttämällä merkkien 3D-rakennetta.
QR/Barcode Scanner	QR- ja viivakoodin lukija on nopea tunnistuksessa, virhelukuja tulee vähemmän kuin merkeillä ja analyysit ovat tarkkoja. Käytössä avoimen lähdekoodin kirjasto Zxing-cpp.
Cloud Recognizer	Pilvipohjainen kuvantunnistuspalvelu on suunniteltu projekteille, jotka vaativat suurin määrin kohdekuvia ja niiden päivittämistä. Tunnistettavien kuvien määrä on rajaton ja tunnistamisen määrää voidaan lisätä tarpeen mukaan.

4.4 Kudan

Kudan AR-kehitystyökalu on iOS ja Android laitteille suunnattu moottori, jonka avulla sovelluksilla voidaan tunnistaa ja seurata fyysistä ympäristöä, hyödyntäen älylaitteen omia komponentteja. Alusta tarjoaa oman 3D-renderöinnin. Moottori voidaan liittää mihin tahansa projektiin, tarkoittaen sitä, että AR-sisällön luominen on yhtä helppoa uusissa sekä jo julkaistuissa sovelluksissa. KudanCV on komponentti, joka vastaa kuvien ja ympäristön tunnistuksesta. Tämä kehys on kirjoitettu C++ ohjelmointikielellä ja on helposti integroitavissa kolmannen osapuolen renderöintiohjelmaan tai alustaan.

Poiketen muista suosituista AR-kehitystyökaluista, Kudan ei tue pilvipohjaista tunnistusta, koska verkon viiveet vaikuttavat tunnistusnopeuteen. Vaihtona pilvipohjaiselle tunnistukselle, Kudaniin voidaan määrittää väliaikaisia markkereita tietyille tilanteille ja sijainneille. Tätä menetelmää käyttämällä markkereiden päivittäminen poistaa verkosta aiheutuvat viiveet ja mahdollistaa joustavan tavan päivittää markkereita ilman taustalla olevaa lähdekoodia. Kehitystyökalu mahdollistaa myös KARMarker-tiedostojen luomisen, joilla voidaan tallentaa suuria määriä markkereita yhteen pieneen tiedostoon (Kudan Guides, n.d.). Taulukossa 6 on Kudan-kehitystyökaluun kuuluvat toiminnot.

Taulukko 6. Kudan SDK toiminnot

Toiminto	Selite
Marker Recovery Mode	Uudelleen tunnistaa markkerit nopeammin, jos markkeri katoaa. Helpottaa tunnistamista vaikeista kulmista ja suurilta etäisyyksiltä. Tämä toiminto vie laitteen prosessorilta normaalia enemmän resursseja.
Extended Detection & Tracking	Järjestelmä luo sen havaitsemasta ympäristöstä automaattisesti tunnistettavia markkereita. Kun järjestelmä oppii sen ympäristön, se tunnistaa kohteet suuremmilta etäisyyksiltä.
Automatic Cropping	Ominaisuus havaitsee ympäristöstä automaattisesti kohteiden alueet ja käyttää niitä seuranta menetelmänä. Tämä mahdollistaa suurien kuvien käytön ilman seurantaresoluution menettämistä.

4.5 EasyAR

EasyAR-kehitystyökalusta on saatavilla kaksi eri versiota, jotka ovat basic ja pro. Pro versiolla on mahdollista tehdä kohteentunnistusta. Se tukee SLAM teknologiaa ja sovelluksiin on integroitu näppäin, jolla voi nauhoittaa näyttöä. Perusversiolla voi tehdä yksinkertaisia kuvantunnistustoteutuksia, videon toistoja, se tukee QR-koodeja ja usean kuvan samanaikaista tunnistusta. EasyAR julkaisi molemmille versioille sisäänrakennetun tuen pilvipalvelulle, jonka avulla voidaan tunnistaa kuvia pilvestä. Pilvipalvelua voi käyttää ilmaiseksi kuukauden ajan. Kehitystyökalun verkkosivuilta löytyy kattavat dokumentit eri toimintojen käyttöönottoon ja myös ohjeet Unity-pelimoottorille. Taulukossa 7 on esitetty EasyAR-kehitystyökalun toiminnot. (EasyAR, n.d.)

Taulukko 7. EasyAR-alustan toiminnot

Toiminto	Selite
3D kohteen tunnistus	Menetelmällä voidaan käyttää kohteen 3D-mallia tunnistukseen, jolloin sovellukselle ei tarvitse määritellä etukäteen rekisteröintivaiheita. Kohteen mallitiedostoa ei tarvitse ladata EasyAR sovellukseen tai verkkosivulle. Verkkodokumentaatioissa on hyvät ohjeet OBJ tiedoston luomiseen.
Usean kohteen tunnistus	Kehitystyökalu mahdollistaa usean kohteen samanaikaisen tunnistuksen. Pro-versiolla on mahdollista tunnistaa kuva ja objekti yhtä aikaa. Sovellukseen on mahdollista määrittää, kuinka monta kohdetta voidaan tunnistaa samanaikaisesti.

Target Configuration	Joustava kohteiden hallinta käyttöliittymä, jolla voidaan luoda kohteita ajonaikaisesti. Kirjautumista verkkosivustolle ei vaadita.
----------------------	---

4.6 Vertailu

Tutkittavat alustat valittiin niiden 3D-objektin tunnistusmahdollisuuksien ja pilvipalveluiden saatavuuden perusteella. Kehitystyökaluista tehtiin taulukot, joissa vertaillaan niiden tukemia tiedostomuotoja, ominaisuuksia ja kohteentunnistukseen sekä pilvipalveluun liittyviä rajoituksia.

4.6.1 Merkkipohjainen tunnistus

Taulukossa 8 vertaillaan AR-alustojen merkkipohjaisessa tunnistuksessa käytettäviä tiedostomuotoja, kokoja ja muita rajoituksia. Sisältö ladataan alustoiden omilla sisällönhallintajärjestelmillä tai suoraan laitteen omaan tallennustilaan. Alustoista tutkitaan myös sitä tarjoavatko ne valmiita markkereita. Yleensä alustat eivät tarjoa suoraan valmiita markkereita vaan selaimessa toimivan sisällönhallintajärjestelmän, joka luo kohteesta markkerin ja arvioi sen laadun tunnistukseen. Alustat, jotka tarjoavat markkereiden luomiseen tarkoitettun sisällönhallintajärjestelmän ovat Vuforia, Wikitude, MAXST ja EasyAR. Mikään alusta ei tarjoa valmiita markkereita.

Taulukko 8. Markkereiden rajoitukset

SDK	Kuva	Video	Teksti	QR- ja viivakoodi
Vuforia	8 tai 24bit JPG tai PNG RGB tai Greyscale Max. koko 2.25Mb. Ladataan Target managerin kautta.	Mp4 formaatti tehokain tunnistukseen. Videon rajoitukset Unityn käyttöjärjestelmä vaatimusten mukaan.	Tunnistaa valmiiksi 100 000 englanninkielistä sanaa, voidaan määrittellä tunnistettavia sanoja. UTF-8 standardi.	Voidaan käyttää markkereina.
Wikitude	Ladataan Wikituden Target managerin kautta. JPG tai PNG maksimikoko 10Mb. Kuvan leveys ja korkeus tulisi olla 500-1000 pikselin välillä.	Max. Resoluutio 720p (H.264) Mov, Mp4, Avi tai M4v Max. koko 100Mb Max. Bitrate: 1250 kbps Videoita voidaan toistaa suoratoistopalveluiden kautta kuten: Youtube ja Vimeo.	Anyline liitännäinen. (Wikituden oma työkalu) Mahdollista tunnistaa ja liittää sisältöä tekstiin, numeroihin, QR- ja viivakoodeihin.	Anyline liitännäinen.

MAXST	Target manager, optimoitu Chrome selaimelle. JPG tai PNG Max. Koko 3Mb. Tiedostonimi enintään 20 merkkiä. 100 kuvaa per sovellus.	Unity-pelimoottorin rajoitusten mukaan.	Ei tukea	Kehitysalustan oma - ja viivakoodin luku toiminto.
Kudan	JPG tai PNG, kuvat haetaan laitteen paikallisesta tallennustilasta. Ei muita rajoituksia määritetty.	Videotiedosto voi olla mikä tahansa alkuperäisen laitteen tukemuoto. Voidaan toistaa HD 30fps videoita.	Ei määritetty	Ei määritetty
EasyAR	JPG tai PNG suoraan ladatessa. Kuvan asetuksia voidaan muuttaa pakkaamalla se JSON formaattiin.	Voi toistaa videon laitteen paikallisesta tallennus tilasta mp4 formaatissa tai verkkoosoitteesta.	Ei tukea	QR- ja viivakoodeista voidaan näyttää tekstiä.

4.6.2 Merkitön tunnistus

Merkitön tunnistus tarkoittaa sitä, pystyykö alustalla tunnistamaan käyttäjän ympäristön ja hyödyntääkö se laitteen GPS-tietoja tai muita sensoreita käyttäjän liikkeen ja sijainnin havainnollistamiseksi. Näitä toimintoja ovat yleensä nimeltään Extended Tracking, Instant Tracking, Ground Plane tai jokin SLAM-tekniikkaa käyttävä työkalu. Alustat ovat alkaneet saamaan myös SLAM-tukea, joten ympäristön kartoittaminen onnistuu paremmin. Taulukossa 9 tutkitaan alustojen ympäristön- ja sijainnin tunnistus metodeja.

Taulukko 9. Alustojen ympäristön- ja sijainnintunnistus menetelmät

SDK	Ympäristön tunnistus & seuranta ominaisuus	Sijainnin tunnistus & seuranta ominaisuus
Vuforia	Kyllä Extended Tracking. Alustan Ground Plane toiminto aktivoi SLAM-tekniikan, jotta epätasaisten pintojen tunnistus olisi tehokkaampaa.	Ei käytä laitteen GPS-tietoja.
Wikitude	Kyllä Instant Tracking. Toimintoa käytetään ympäristön kartoittamiseen. Alusta tarjoaa myös SMART (Seamless AR Tracking) työkalun, joka mahdollistaa 3D-objektien pudottelemisen ympäristöön ilman etukäteen määritettyä tietoa.	Kyllä, alustalla voi luoda sijaintitietoihin perustuvia merkkejä. Alusta pystyy käyttämään laitteen GPS-tietoja, verkkoa ja merkkejä (engl. beacons).
MAXST	Kyllä Instant Tracker. Toiminto kartoittaa ympäristön käyttäen laitteen sensoreita. Ympäristöön voidaan pudotella valmiita 3D-objekteja.	Ei käytä laitteen GPS-tietoja.

Kudan	Kyllä Arbitrary Tracking toiminto.	Ei käytä laitteen GPS-tietoja.
EasyAR	Ei selviä suoraan alustan verkkosivuilla. Hyödyn- tää SLAM tekniikkaa kohteen tunnistuksessa.	Ei käytä laitteen GPS-tietoja.

4.6.3 Pilvipohjainen tunnistus

Taulukossa 10 vertaillaan alustojen pilvipohjaisen palvelun tunnistukseen liittyviä rajoituksia, kuten tietokantaan tallennettavien kuvien ja kuvien tunnistuksen määrää. Pilvipalveluista on saatavilla eri lisenssejä, joiden mukaan tallennus ja tunnistus määrät riippuvat. Lisenssien hintoja ei näytetä taulukossa. Taulukossa 10 vertaillaan alustojen pilvipohjaiseen tunnistukseen liittyviä rajoituksia.

Taulukko 10. Pilvipohjaisten palveluiden rajoitukset

SDK	Palvelu	Rajoitukset
Vuforia	Vuforia Cloud Recognition	1 tietokanta sovellusta kohden. Tietokantaan tallennettavien kohteiden määrä sekä tunnistusten määrä riippuu pilvipalvelimen lisenssistä. Palveluun voi tallentaa 3D-kohteentunnistuksessa käytettäviä tiedostoja, mutta on suositeltavaa käyttää laitteen omaa tallennustilaa nopeamman tunnistuksen saavuttamiseksi.
Wikitude	Wikitude Cloud recognition	Ei hyväksy kuvia, jotka ovat kooltaan yli 1024kbs. Ilmaissella lisenssillä tietokantaan voi tallentaa 50 000 kuvaa ja 1000 tunnistusta kuukaudessa. 3D-kohteen tunnistuksessa käytettävät tiedostot voidaan tallentaa palveluun.
MAXST	Cloud recognition Web API	Ilmaissessa lisenssissä vesileima sovelluksissa. Palvelusta voidaan tunnistaa vain kuvia. Sisällön näyttämisen taulukossa mainitut rajoitukset ovat käytössä myös pilvipohjaisessa tunnistuksessa.
Kudan	Ei tue pilvipohjaista tunnistusta	Alusta ei tue pilvipohjaista tunnistusta, joten sitä ei voida verrata.
EasyAR	EasyAR CRS	Palvelusta ladattu kohde voidaan tallentaa laitteen paikalliseen tallennustilaan ja käyttää tätä Offline -tilassa. 100 000 kohdetta tietokantaa kohden. Palveluun voidaan alustan verkkosivun mukaan tallentaa muutakin sisältöä kuin kuvia, mutta niistä ei kerrota tarkemmin. Kohteita voidaan lisätä ja päivittää palvelun avulla.

4.6.4 3D -objektin tunnistus

Alustojen tärkein vertailtava kohde on 3D -objektin tunnistus. Taulukossa vertaillaan tunnistamiseen liittyviä rajoituksia, kuten objektin fyysistä kokoa, tuettuja tiedostomuotoja sekä sitä, kuinka monta kappaletta voidaan tunnistaa samanaikaisesti. Ylempänä taulukossa on mainittu, jos kohteen tunnistuksessa käytettävät tiedostot voidaan tallentaa pilvipalveluun. Taulukossa 11 vertaillaan alustojen kappaleen tunnistuksen rajoituksia.

Taulukko 11. Kappaleen tunnistuksen vertailu

SDK	Objektin fyysisen koon rajoitukset	Tiedostokoko	Tuetut tiedostomuodot	Kappaleiden samanaikainen tunnistus
Vuforia	Suositus min. 12cm	Max. 35Mb	Object Scannerilla luotu object data, OD	2
Wikitude	Ei suositella alle 5cm kokoisia objekteja	Ei määritelty	.wt3 formaatti, luodaan Wikituden 3D encoderilla	1
MAXST	0,3m-1,5m	Ei määritelty	3dmap tiedosto, luodaan Visual SLAM työkalun avulla	Ei määritelty
Kudan	Ei määritelty	Ei määritelty	FBX ja OBJ	Ei määritelty
EasyAR	Ei rajoituksia kohteen fyysiseen kokoon.	Ei määritelty	OBJ ja MTL tiedosto. Texture map tiedosto JPEG tai PNG muodossa. Polygoni-malleja voi käyttää.	4-6

4.6.5 Vertailun ulkopuolelle jääneet AR-alustat

Unity on yhteensopiva monien AR-kehitystyökalujen kanssa ja kaikki eivät mahtuneet työssä vertailtavien alustojen joukkoon. Tässä taulukossa on listattuna osa alustoista ja syyt miksi niitä ei vertailtu.

Taulukko 12. Alustoja Unity-pelimoottorille, joita ei vertailtu.

SDK	Syy
ARCore	Ei tue suoraan 3D-kohteen tunnistusta, pilvipohjaisesta palvelusta voidaan hakea vain valmiita objekteja ja pudotella niitä ympäristöön.
ARKit	Sovelluksia voidaan kehittää vain iOS -käyttöjärjestelmille.
DeepAR	Kasvojen tunnistukseen erikoistunut työkalu.
Immersal	Ei heti saatavilla.
XZIMG	Kasvojen tunnistukseen erikoistunut työkalu.
ARToolKit	Epäselvät verkkosivut, muuttui avoimen lähdekoodin projektiksi, ei varmuutta 3D-kohteen tunnistuksesta.
Mapbox	Sijaintitietoihin perustuva alusta, ei 3D-kohteen tunnistusta

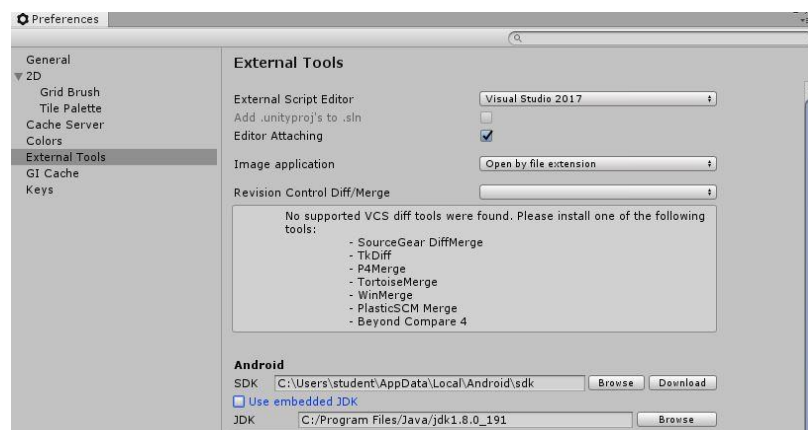
5 KAPPALEEN TUNNISTUS VALITUILLA ALUSTOILLA

Opinnäytetyön käytännön osiossa luodaan Vuforia ja EasyAR työkaluilla yksinkertainen mobiilisovellus, joka tunnistaa hissien kappaleet ja lisää niihin tekstimuodossa kuvauksen kappaleesta. Lopuksi vertaillaan työkalujen eroja.

Kappaleentunnistuksen testaukseen ja kappaleiden skannaukseen käytän Moto g5 ja OnePlus 5 Android-laitteita. Molemmista laitteista täytyy aktivoida USB-vianetsintä tila, jotta sovellusta voidaan testata kehityksen aikana ilman APK -tiedoston siirtämistä laitteeseen. Vianetsintätilan aktivoimiseksi laitteesta täytyy aktivoida myös kehittäjäasetukset.

Kehittäjäasetukset aktivoidaan Android laitteilla painamalla 7 kertaa ohjelmistoversion numeroa. Ohjelmistoversion numero löytyy ”tietoa puhelimesta” osiosta, joka sijaitsee asetukset-valikossa. Kun kehittäjäasetukset ovat aktivoitu, järjestelmä valikossa näkyy kehittäjäasetukset-valikko, sieltä pystytään aktivoimaan USB-vianetsintätila.

Android-sovelluksen rakentamiseen Unity editoriin täytyy määrittää Android Software Development Kit ja Java Development Kit, jotta koodi rakentuu oikein Android käyttöliittymälle. Työkalujen polku määritellään editorin External Tools välilehdellä.



Kuva 6. Unity External Tools-valikko

5.1 Vuforia

Unity editoriin täytyy määrittää Vuforian lisenssi avain, ennen kuin alustaa voidaan käyttää. Avain tilataan Vuforian Developer Portalin License Manager sivustolla. Valitsin opinnäytetyöhön ilmaisen avaimen, koska sillä voidaan suorittaa tuhat tunnistusta kuukaudessa ja uskon sen riittävän kappaleen tunnistuksen testaukseen. Kuvassa 7 on keskeneräinen lisenssin tilaus, josta näkyy myös muut rajoitukset ilmaiselle lisenssille.

License Manager
Target Manager

[Back To License Manager](#)

Add a free Development License Key

License Name *

You can change this later

License Key

Develop
 Price: No Charge
 Reco Usage: 1,000 per month
 Cloud Targets: 1,000
 VuMark Templates: 1 Active
 VuMarks: 100

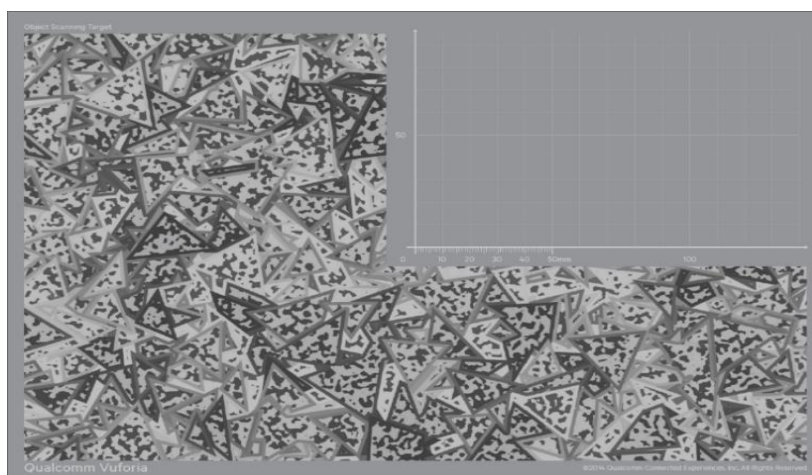
By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Kuva 7. Vuforian lisenssin tilaus.

Vuforia aktivoidaan Unityn XR-asetuksista, jonka jälkeen lisenssiavain voidaan syöttää Vuforia Configuration-ikkunasta License Key-kohtaan. Samasta ikkunasta määritellään myös, kuinka monta kappaletta sovellus voi tunnistaa samanaikaisesti. XR-asetuksien yläpuolelta löytyy muut asetukset-valikko, josta löytyy Android TV-yhteensopivuus. Tämä on vakiona päällä ja se tulisi poistaa käytöstä, sillä sovellus ei rakennu oikein älypuhelimille, eikä sovellusta voida testata.

5.1.1 Kappaleen skannaus ja tietokannan luominen

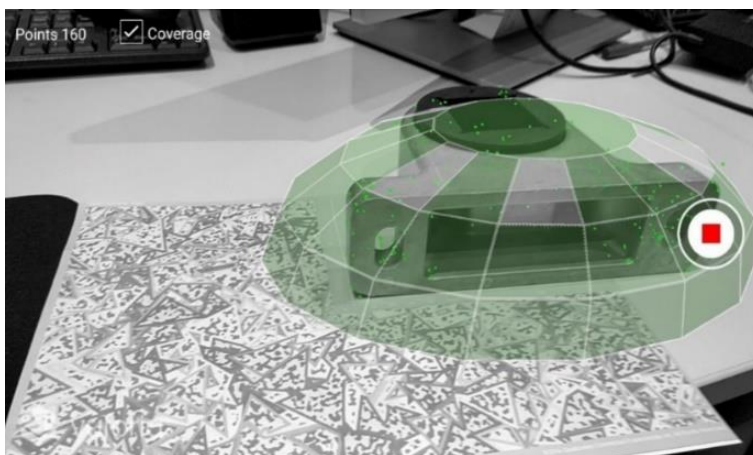
Vuforian kohteen tunnistukseen tarvitsee ladata Object Scanner-mobiili-sovellus, jonka avulla luodaan Object data -tiedosto tunnistusta varten. Mobiili-sovellus on ilmaiseksi ladattavissa Vuforian verkkosivujen kautta. Ennen skannauksen aloittamista tarvitsee tulostaa kohteen skannauksessa käytettävä pohja. Tätä pohjaa käytetään määrittelemään kohteen sijainti suhteessa sen paikalliseen alkuperään. Ruudukon mittayksikkö on millimetreinä ja sitä käytetään myös kohtausyksikköön ja arvioituu fyysisen kohteen mittakaavaan.



Kuva 8. Vuforia Object Scanning Pohja (Vuforia Developer Library, Vuforia Object Scanner n.d.).

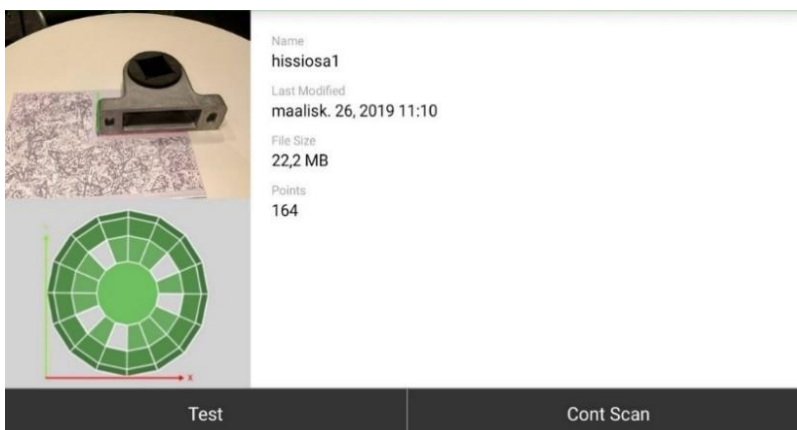
Skannaamisen ajaksi fyysisen kohde sijoitetaan pohjan oikeaan yläkulmaan. Skannaus ympäristön tulisi olla valoisa, eikä kappaleen päällä tulisi olla varjoja skannauksen aikana. Skannaus prosessi on yksinkertainen, kun kameraa liikutetaan kappaleen, sen ympärille muodostuu kehikko.

Vihreät palat ovat tunnistettuja yksityiskohtia ja harmaat ovat vielä tunnistamatta. Onnistuneen kappaleen tunnistuksen saavuttamiseksi kaikkia yksityiskohtia ei tarvitse skannata, eikä se ole välttämättä suositeltavaa, sillä object data tiedosto saattaa ylittää 35Mb rajan.



Kuva 9. Skannaus vaihe.

Kun kappaleesta on tunnistettu tarpeeksi yksityiskohtia, sitä voidaan testata Object Scanner-mobiilisovelluksessa. Testauksen aikana kappaleen ei tule olla skannaus pohjan päällä. Onnistuneessa testauksessa kappaleen viereen ilmestyy vihreä suorakuutio. Skannausta voidaan jatkaa vielä myöhemmin, jos kappaleesta halutaan poimia loput yksityiskohdat.





Kuva 10. Valmis skannaus.


Seuraava vaihe on luoda tietokanta "Object data" -tiedostoille ja se määritellään joko laitteen paikalliselle tallennustilalle tai pilveen. Valitsin paikallisen tietokannan, koska sovelluksessa käytetään vain muutamaa "Object data" -tiedostoa. Kuvassa 9 on näkymä tiedoston lisäämisestä tietokantaan. Valmis tietokanta ladataan Target Manager sivuston kautta Android Studio, Xcode, Visual Studio tai Unity Editor-kehitysalustoille.


Add Target

Type:









Single Image Cuboid Cylinder 3D Object

File:

File must be Vuforia Object Scanner data. For more information, see the Vuforia Object Scanner Application.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

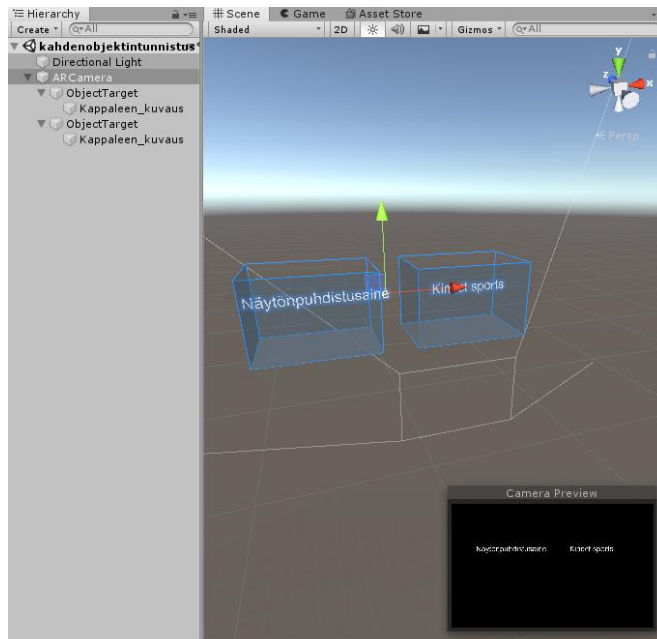
Kuva 11. Object Datan lisääminen Vuforian tietokantaan.

5.1.2 Sovelluksen rakenne ja testaus

Kun Unityllä aloitetaan uusi projekti, aloitusnäkyssä on aina Main Camera ja Directional Light -komponentit. Main Camera voidaan poistaa, sillä se korvataan AR Camera assetilla, joka aktivoi alustan AR toiminnot. Sovelluksen hierarkia muodostuu yksinkertaisuudessaan kolmesta eri komponentista, jotka ovat AR Camera, ObjectTarget ja 3D Teksti.

Kuvassa 12 näkyy sovelluksen komponenttien arvojärjestys. AR Cameran alle sijoitetaan kaksi ObjectTargettia ja niiden sisälle 3D tekstit. Valkoiset viivat kuvastavat kameran suuntaa ja näkyvyyttä ja siniset laatikot objektien sijaintia.

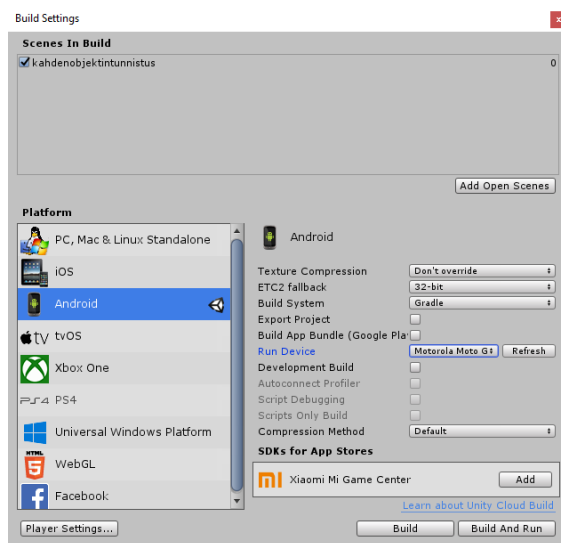
Testauksen aikana ilmeni ongelmia kiinnikkeiden tunnistamisessa, joten valitsin satunnaisesti muutaman kappaleen, jotka ovat eri kokoisia ja muotoisia testausta varten. Kerron tulokset-osiossa tarkemmin ongelmista ja havainnoista, joita ilmeni testauksen aikana.



Kuva 12. Kappaleen tunnistus sovelluksen hierarkia ja editorin näkymä.

ObjectTargetille täytyy määrittää oikea "Object data" -tiedosto. Tämä tapahtuu painamalla aktiiviseksi ObjectTarget komponentti, jonka jälkeen editorin oikealle puolelle aukeaa Inspector-ikkuna. Ikkunasta voidaan muuttaa myös objektin sijaintia, suuntaa ja kokoa tai lisätä objektille jonkin toiminnon suorittava skripti. 3D-tekstin asetuksista muokkasin vain sen kokoa.

Sovellusta voi testata Unityn Build asetuksista, kun testattava laite on yhdistetty USB-välityksellä tietokoneeseen ja kehittäjärooli on aktivoitu, sen tulisi löytyä "Run Device" -valikosta. Painamalla "Build And Run" -näppäintä, Unity rakentaa sovelluksen valitulle alustalle, jonka jälkeen sen tulisi aueta valitulla laitteella. Kuvassa 13 on unityn Build asetukset ja kuvassa 14 kuvankaappaus onnistuneesta kappaleen tunnistuksesta.



Kuva 13. Sovelluksen Build-asetukset.





Kuva 14. Onnistunut kappaleen tunnistus sovellus.

5.2 MAXST

Seuraava testattava alusta on MAXST. Unity paketit ovat ladattavissa MAXST:n verkkosivuilta. MAXST developer-sivuille täytyy rekisteröityä, jotta voi tilata lisenssin alustalle. Valitsin ilmaisen lisenssin, koska sillä saa kaikki alustan toiminnot käyttöön. Lisenssille täytyy määrittää sovelluksen ID, joka täytyy olla sama kuin Unityssä oleva Package Name. MAXST luo oman avaimen eri ohjelmistoversioille, kuten kuvassa 15 näkyy.

SDK License Details

License #	4212	License Type	Free Add
App Name	maxstobject 	App ID (Android Package Name or iOS Bundle Identifier)	com.test.object
Registration Date	2019-04-08	Next billing date (estimated)	-
Payment Method	-	Subscription Status	-
License Key	<div style="border: 1px solid #ccc; padding: 2px;"> - version - - version - 4.1.x 4.0.x </div>		

Kuva 15. MAXST Lisenssin luominen.

Kun MAXST Unity-paketti on viety editoriin, lisenssiavain voidaan syöttää AR Cameran configuration-asetuksista "License Key"-kohtaan. Tämän jälkeen sovelluksen ID määritellään samaksi, kuin lisenssiä tilatessa. ID määritellään editorin "Player Settings"-valikosta.

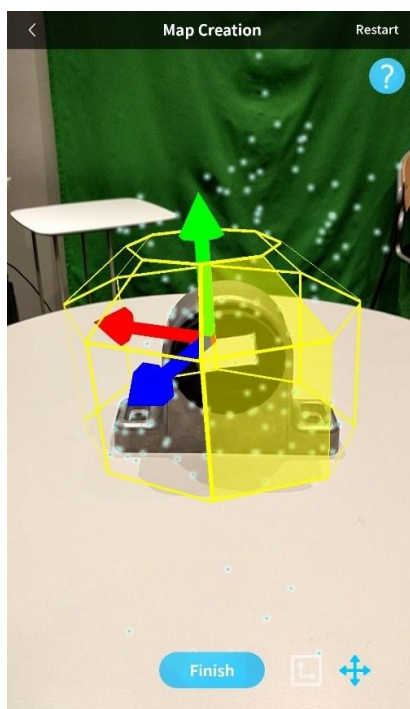
Identification	
Package Name	com.test.object
Version*	0.1
Bundle Version Code	1
Minimum API Level	Android 4.1 'Jelly Bean' (API level 16) ↓
Target API Level	Automatic (highest installed) ↓

Kuva 16. Sovelluksen ID:n määrittäminen.

5.2.1 Kappaleen skannaus

MAXST on vuoden 2019 alussa julkaissut uuden sovelluksen nimeltä Visual SLAM Tool, jolla luodaan kappaleen tunnistuksessa käytettävät 3d map tiedostot. Visual SLAM Tool sovelluksella luodut tiedostot toimivat ainoastaan MAXST uusimmassa ohjelmistoversiossa. Map Manager on alustan aiempi sovellus 3dmap tiedostoille, joten testaan näitä molempia. Molemmat sovellukset ovat ladattavissa Android laitteille Play kaupan kautta. Testatakseni molempia sovelluksia käytän MAXST ohjelmistoversioita 4.1.0 ja 4.0.4.

Visual SLAM Tool muistuttaa paljon Vuforiaan ”Object Scanner”-sovellusta. Skannatessa kohdetta, sen ympärille muodostuu myös kehikko, jonka palat muuttuvat keltaisiksi, kun ne on tunnistettu. Toisin kuin Vuforiaan sovelluksessa, kohde ei tarvitse pohjaa, jonka päälle kohde sijoitetaan skannauksen ajaksi. Kehikon sijaintia ja muotoa voi myös säätää nuolista.



Kuva 17. Skannaus Visual SLAM Tool-sovelluksella.

Työkalua käyttäessä huomasin, että sovelluksen Map Management-osio ei toimi oikein. Map Management on sovelluksen oma kirjasto luoduille 3dmap-tiedostoille, josta tiedostot voidaan siirtää toiselle järjestelmälle tai lähettää sähköpostiin.

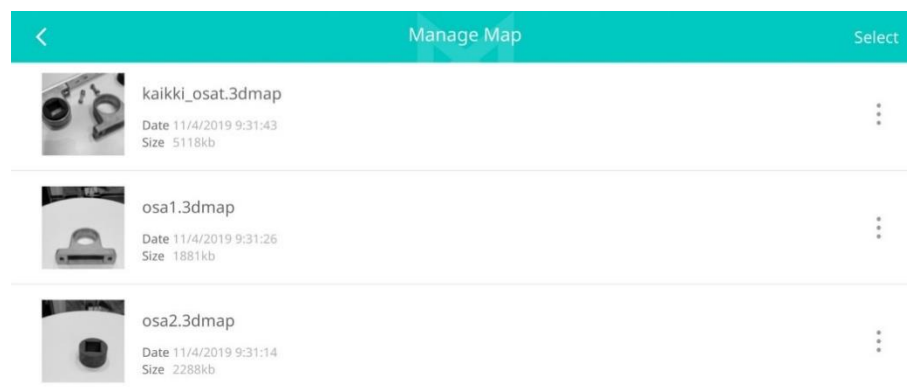
Seuraavaksi testaan, toimiiko MAXST alustan Map Manager sovellus, joka oli ensimmäinen sovellus objektien skannaukselle.

Map Manager sovellus toimi samanlailla kuin aiemmin työssä testatut skannaussovellukset. Ainoa ero muihin sovelluksiin oli, ettei Map Manager sovellus luo skannattavan kappaleen ympärille kehikon, jonka palaset muuttuvat värillisiksi, kun jokin osa kappaleesta tunnistetaan.



Kuva 18. MAXST Map Manager sovelluksella skannaus.

Skannasin kaikki kappaleet samassa näkymässä ja erikseen testatakseni, kuinka hyvin alustan kappaleen tunnistus toimii. Sovelluksen tiedostojen hallintaosio toimi oikein ja tiedostot voi lähettää suoraan sovellukselta sähköpostiin.



Kuva 19. MAXST Map Manager sovelluksen tiedostojen hallinta.

MAXST alustalla ei tarvitse luoda Target Managerin kautta tietokantaa kappaleen tunnistuksessa käytettäville tiedostoille. Tietokannat on tarkoitettu suurille kuva määrille, eikä niitä suositella käytettäväksi kuin pelkästään kuville. 3dmap-tiedostot voidaan siirtää suoraan työpöydältä Unityyn.

5.2.2 Sovelluksen rakenne ja testaus

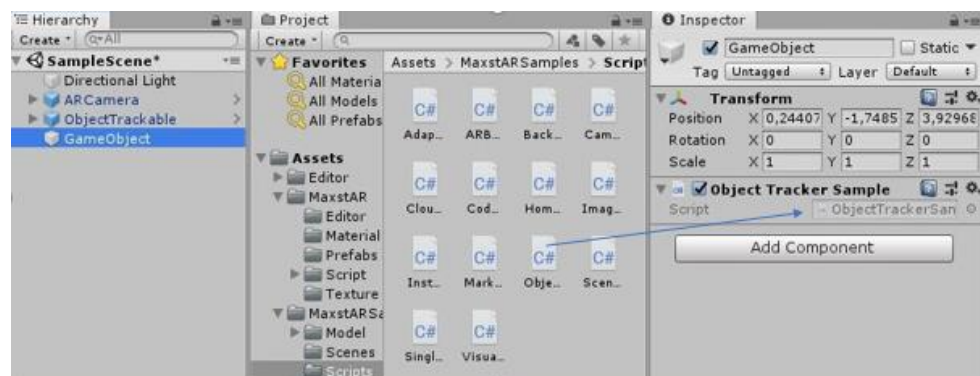
Käytän sovelluksen rakentamiseen MAXST alustan ohjelmistoversiota 4.0.4, koska Map manager sovellus ei tue alustan uusinta 4.1.0 ohjelmistoversiota. Oletusnäköymästä poistetaan taas Main Camera ja lisätään tilalle MAXST-alustan oma ARCamera. ARCamera-komponentin alle lisätään ObjectTrackable. Molemmat komponentit löytyvät "Prefabs"-kansioista, kuten kuvassa 20 näkyy.



Kuva 20. Komponenttien sijainti MAXST Prefabs-kansioissa.

Kun ARCamera on lisätty näkymään ja se painetaan aktiiviseksi "inspector"-ikkunasta. Painamalla "Configuration"-näppäintä päästään syöttämään aiemmin tilattu lisenssi. Lisenssin täytyy olla versiolle 4.0.4. Jos lisenssiä ei syötetä Unityyn ennen sovelluksen ajamista, se ei käynnisty oikein, eivätkä toiminnot ole aktiivisia.

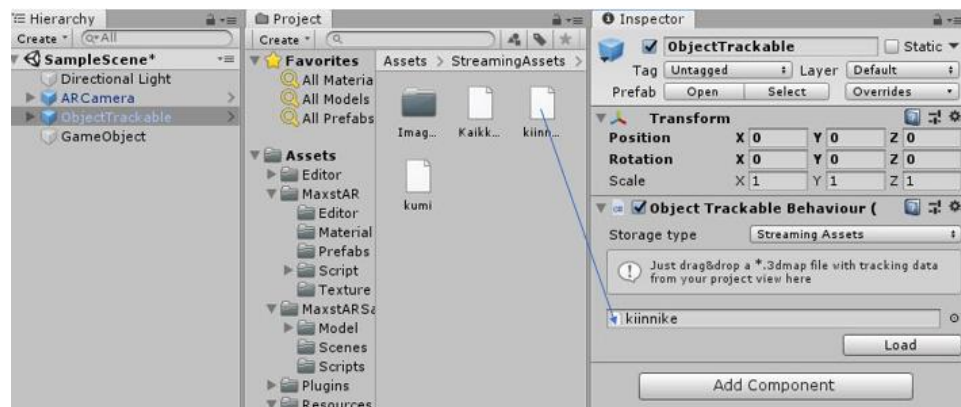
Näkymään lisätään tyhjä "GameObject"-komponentti ja sille lisätään MAXST ObjectTrackerSample-skripti kappaleentunnistusta varten. Skripti voidaan lisätä suoraan kansioista vetämällä se GameObjectin "Inspector"-ikkunassa sijaitsevaan Script kohtaan.



Kuva 21. Skriptin lisääminen tyhjälle GameObject-komponentille.

Aiemmin luodut 3dmap-tiedostot voidaan tuoda nyt Unityyn. Lähetin aiemmin Map Manager-sovelluksessa luodut tiedostot omaan sähköpostiini ja latsin ne tietokoneelle. Tiedostot voidaan vetää työpöydältä suoraan Unityyn. Lisäsin ne StreamingAssets-kansion alle.

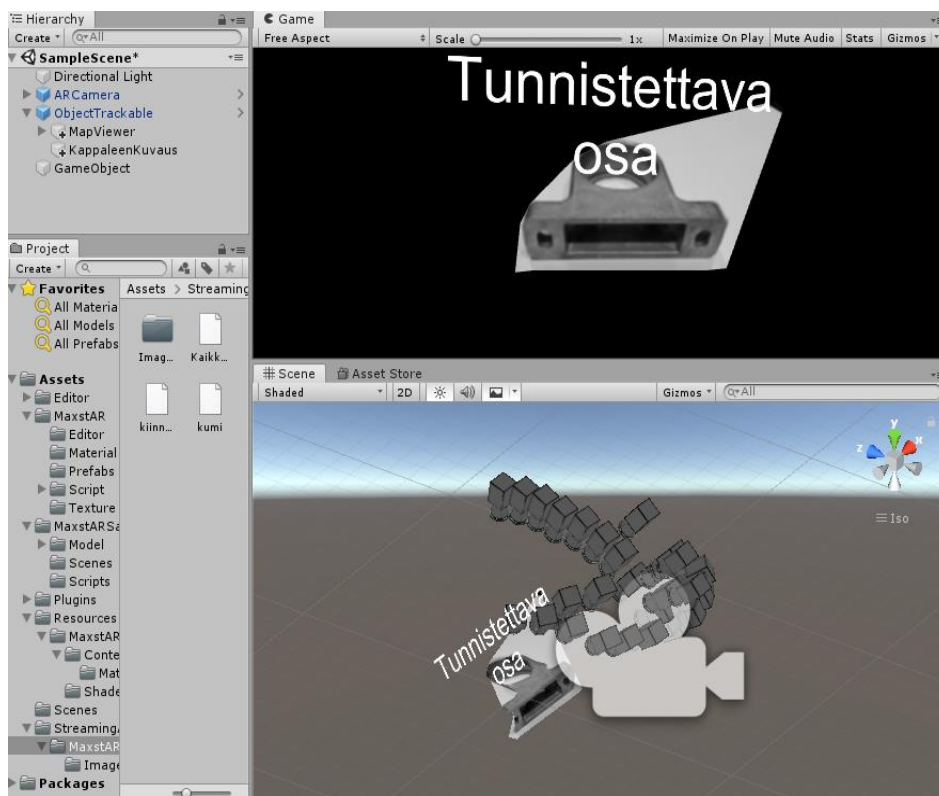
3dmap-tiedoston lisääminen ObjectTrackable-komponentille tapahtuu vetämällä tiedosto "Inspector"-ikkunassa näkyvään Object Trackable Behaviour-kohtaan ja painamalla "Load" -toimintoa.



Kuva 22. 3dmap tiedoston lisääminen ObjectTrackable-komponentille

Kun tiedosto on ladattu ObjectTrackable-komponentille, MapViewer-komponentti rakentuu automaattisesti ObjectTrackablen alle. MapViewer koostuu useista eri avainkehysistä (engl. Keyframe). Alusta luo avainkehysten jokaisesta kappaleesta skannatusta kuvakulmasta, joka täyttää jonkin alustan ehdoista.

Avainkehystä voidaan muuttaa MapViewerin Inspector ikkunasta liikuttamalla KeyFrame index-palkkia. Jokaiseen avainkehykseen voidaan sijoittaa virtuaalista sisältöä, jotta sovelluksessa virtuaalinen sisältö näkyisi tarkemmin tunnistettavassa kappaleessa. Kuvassa 23 näkyvät kamerat kuvastavat kaikkia kappaleesta otettuja avainkehysiksiä.



Kuva 23. MapViewer avainkehukset.

Virtuaalinen sisältö sijoitetaan MapViewerin alle ja tässä sovelluksessa lisäksi kappaleeseen tekstiä, samalla tavalla kun Vuforia alustaa testatessa tasapuolisten tulosten vertailemiseksi. Kuvassa 24 näkyy sovelluksen komponenttien arvojärjestys.



Kuva 24. MAXST sovelluksen komponenttien järjestys.

En onnistunut tunnistamaan alustalla kuin yhden kappaleen kerrallaan. Valmistajan sivuilla ei ole määritelty kuinka monta kappaletta voidaan

tunnistaa samanaikaisesti. Toisin kun Vuforia-alustalla, MAXST Map manager sovelluksella pystyy skannaamaan monta kappaletta samassa näky-
mässä, mutta yksittäisille kappaleille ei pysty määrittämään omia virtuaalisia objekteja. Sain kappaleentunnistuksen onnistuneesti toimimaan jokaisella yksittäisellä osalla muutaman mutkan kautta. Kerron havainnoista enemmän tulokset-osiossa. Kuvassa 25 kuvankaappaus MAXST-alustalla onnistuneesta kappaleentunnistuksesta.



Kuva 25. MAXST onnistunut kappaleentunnistus.

6 TULOKSET

Opinnäytetyön käytännön osassa oli tarkoituksena tehdä kaksi samanlaista kappaleentunnistussovellusta eri AR-alustoilla. Valitsimme toimeksiantajan kanssa aluksi testattaviksi alustoiksi Vuforia ja EasyAR-alustat. Sovelluksen tarkoituksena oli tunnistaa halutut kappaleet ja lisätä tekstimuo-dossa kuvaus kappaleesta, joka tunnistetaan.

Vuforia-alustan käytössä ilmeni aluksi ongelma, kun sovellusta testatessa kamera ei tarkentanut mihinkään objektiin. Selvisi, että Vuforia käyttää joi-tain ARCoren toimintoja, jonka seurauksena automaattinen tarkennus ei toimi kaikilla laitteilla, koska ARCore ei tue automaattista tarkennusta tällä hetkellä. Sain tarkennuksen toimimaan lisäämällä Vuforian ARCamera komponenttiin kamerantarkennus skriptin, jonka löysin Unityn fooru-meilta.

Vuforian ”Object scanner”-sovellus oli helppokäyttöinen ja skannaustulokset olivat tarkkoja. Skannasin toimeksiantajan haluamat kappaleet monta kertaa, eikä alusta onnistunut tunnistamaan näitä kappaleita. Testasin myöhemmin kappaleen tunnistusta eri muotoisilla ja kokoisilla objekteilla ja tunnistus toimi todella hyvin. Vuforia suosittelee verkkosivuillaan tunnistuksessa käytettävien objektien sisältävän mahdollisimman paljon yksityiskohtia nopean tunnistamisen saavuttamiseksi. Tästä voi siis päätellä, etteivät metallipintaiset valossa kiiltävät kappaleet ole paras mahdollinen tunnistettava kappale Vuforia-alustalle. Onnistuin tunnistamaan kaksi kappaletta samanaikaisesti, joka oli alustan maksimimäärä tunnistettaville esi-neille. Vuforian verkkosivuilla löytyi kattavat ohjeet kappaleen tunnistukseen.

Alkuun toinen testattava alusta oli EasyAR. Alustan kappaleen tunnistuk-sessa käytettävät OBJ, MTL ja JPEG-tiedostot täytyi luoda valmiista malli-kuvista. Tämä onnistui 3Ds Max-ohjelmalla. En kuitenkaan onnistunut rakentamaan toimivaa sovellusta alustalla, sillä se vaati JSON osaamista. EasyAR alustan testaaminen vaihdettiin MAXST alustaan.

MAXST skannaus sovellukset toimivat samalla periaatteella kuin Vuforian sovellus. Uudempi sovellus Visual SLAM Tool muistutti hyvin paljon vuforian ”Object scanner”-sovellusta. Molemmissa sovelluksissa skannattavan esineen ympärille muodostui kehikko ja tarkoituksena oli skannata kaikki kehikon osat. Visual SLAM Tool ei kuitenkaan vaatinut skannauspohjaa. So-velluksella luotuja tiedostoja ei voitu hyödyntää työssä, sillä ne ei tallentu-neet sovelluksessa. Tästä syystä jouduin lataamaan MAXST-alustan van-hemman ohjelmistoversion ja skannaus sovelluksen.

Map Manager sovellus toimi paremmin kuin Visual SLAM Tool. Kappaleita skannatessa näkymään ilmestyi vain pisteitä tunnistettujen kohtien mer-kiksi, joten alkuun jäi epäselväksi, onko kappale skannattu tarpeeksi hyvin.

Sovelluksella pystyi myös skannaamaan useamman kappaleen samassa näkymässä. MAXST verkkosivuilta löytyi unitylle selkeät ohjeet kappaleen tunnistuksen rakentamiselle. Alkuun sovellusta ajaessa ilmeni ongelma, kun unitystä avatussa sovelluksessa näkyi vain musta ruutu. Tämä korjautui muuttamalla Unity projektin laatuasetuksista "anti-alias"-metodia alemmas.

Toisin kuin Vuforia, MAXST onnistui tunnistamaan kaikki toimeksiantajan haluamat kappaleet, mutta tunnistus onnistui vain yhdellä kappaleella kerrallaan. Skannasin kaikki kappaleet samaan tiedostoon ja testasin liittää siihen virtuaalisia kohteita. Kohteet näkyivät oikein, mutta jos jotain kappaletta liikutti, kaikki virtuaalinen sisältö katosi. Tunnistus toimi parhaiten ympäristössä, missä kappaleet oli skannattu.

7 YHTEENVETO

Opinnäytetyöni onnistui mielestäni hyvin. Opinnäytetyön tarkoituksena oli tutkia Augmented Reality alustoja unity pelimoottorille, miten alustat eroavat toisistaan ja mikä niistä toteuttaa kappaleen tunnistuksen parhaiten.

Haastavinta ja eniten aikaa vievää oli suunnitella vertailutaulukot alustojen ominaisuuksien tutkimiseksi. Taulukon täyttö osoittautui myös odotettua hankalammaksi, sillä luotettavaa tietoa oli vaikea löytää muualta kuin valmistajan verkkosivuilta.

Kappaleentunnistus onnistui Vuforiailla hyvin, vaikkei alun perin määrättyjä kappaleita onnistuttu tunnistamaan. MAXST-alustalla tunnistus kuitenkin onnistui kaikille halutuille osille, mutta en onnistunut rakentamaan sovellusta, joka tunnistaisi usean kappaleen samanaikaisesti. Sovelluksien tarkoituksena oli lisätä tekstinä kuvaus siitä kappaleesta, joka tunnistetaan onnistuneesti. Mielestäni Vuforia suoritti tunnistuksen nopeammin kuin muut alustat.

Opinnäytetyö lisäsi ymmärrystäni lisätystä todellisuudesta ja siitä, mitä seuraus ja tunnistusmenetelmiä alustat käyttävät. Opin myös käyttämään paremmin Unity pelimoottoria, sekä tiedonhakua.

LÄHTEET

Amin, D. & Govilkar, S. (2015). Comparative Study of Augmented Reality Sdk's. Haettu 23.1.2019 osoitteesta https://www.researchgate.net/publication/276855764_Comparative_Study_of_Augmented_Reality_Sdk's

Baker, M. (2016). How Game Engines Work. Haettu 18.1.2019 osoitteesta <https://interestingengineering.com/how-game-engines-work>

Cheng, M. (2018) Apple Files Patent for Augmented-reality Windshield. Haettu 1.2.2019 osoitteesta <https://www.futurecar.com/2564/Apple-Files-Patent-for-Augmented-reality-Windshield>

EasyAR Documentation a. (n.d.). EasyAR 3D Object Tracking. Haettu 18.2.2019 osoitteesta <https://www.easyar.com/doc/EasyAR%20SDK/Guides/EasyAR-3D-Object-Tracking.html>

EasyAR. (n.d.). What is EasyAR CRS. Haettu 19.2.2019 osoitteesta <https://www.easyar.com/view/crs.html>

Enger, M. (2013). Game Engines: How do they work? Haettu 18.1.2019 osoitteesta <https://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/>

Fine, R. (2017). UnityScript's long ride off into the sunset. Haettu 22.1.2019 osoitteesta <https://blogs.unity3d.com/2017/08/11/unityscripts-long-ride-off-into-the-sunset/>

Haas, J. (2014). A History of the Unity Game Engine. Haettu 21.1.2019 osoitteesta https://web.wpi.edu/Pubs/E-project/Available/E-project-030614-143124/unrestricted/Haas_IQP_Final.pdf

Kudan. (2016). An introduction to Simultaneous Localization and Mapping. Haettu 30.1.2019 osoitteesta <https://www.kudan.io/post/an-introduction-to-simultaneous-localisation-and-mapping>

Kudan Guides. (n.d.). Framework. Haettu 12.2.2019 osoitteesta <https://kudan.readme.io/docs/framework>

MAXST Developer a. (n.d.). Everything's possible with MAST AR SDK. Haettu 14.2.2019 osoitteesta <https://developer.maxst.com/Features>

MAXST Developer b. (n.d.). MAXST tools. haettu 15.2.2019 osoitteesta https://developer.maxst.com/MD/doc/4_1_x/intro

Paladini, M. (2018). 3 Different types of AR explained: Marker-based, Markerless & Location. Haettu 24.1.2019 osoitteesta <https://www.blippar.com/blog/2018/08/14/marker-based-markerless-or-location-based-ar-different-types-of-ar>

Schechter, S. (2014). What is markerless Augmented Reality? Haettu 28.1.2019 osoitteesta <https://www.marxentlabs.com/what-is-markerless-augmented-reality-dead-reckoning/>

Siltanen, S. (2012). Theory and applications of marker-based augmented reality. Haettu 26.1.2019 osoitteesta <https://www.vtt.fi/inf/pdf/science/2012/S3.pdf>

The Medical Futurist. (2017). The Top 9 Augmented Reality Companies in Healthcare. Haettu 31.1.2019 osoitteesta <https://medicalfuturist.com/top-9-augmented-reality-companies-healthcare>

Unity a. (n.d.). Game engines-how do they work? Haettu 18.1.2019 osoitteesta <https://unity3d.com/what-is-a-game-engine>

Unity b. (n.d.). System Requirements for Unity 2018.3. Haettu 18.1.2019 osoitteesta <https://unity3d.com/unity/system-requirements>

Unity Manual a. (n.d.). Scripting. Haettu 19.1.2019 osoitteesta <https://docs.unity3d.com/Manual/ScriptingSection.html>

Unity Manual b. (n.d.). Gameobjects. Haettu 19.1.2019 osoitteesta <https://docs.unity3d.com/Manual/GameObjects.html>

Unity Manual c. (n.d.). Multiplayer Overview. Haettu 21.1.2019 osoitteesta <https://docs.unity3d.com/Manual/UNetOverview.html>

Unity Manual d. (n.d.). Physics. Haettu 21.1.2019 osoitteesta <https://docs.unity3d.com/Manual/PhysicsSection.html>

Unity Manual e. (n.d.). Graphics Overview. Haettu 21.1.2019 osoitteesta <https://docs.unity3d.com/Manual/GraphicsOverview.html>

Vuforia Developer Library a. (n.d.). Vuforia Engine Features. Haettu 6.2.2019 osoitteesta <https://library.vuforia.com/content/vuforia-library/en/features/overview.html>

Vuforia Developer Library b. (n.d.). Object Recognition. Haettu 6.2.2019 osoitteesta <https://library.vuforia.com/articles/Training/Object-Recognition>

Vuforia Developer Library c. (n.d.). Cloud Recognition. Haettu 7.2.2019 osoitteesta <https://library.vuforia.com/articles/Training/Cloud-Recognition-Guide>

Vuforia Engine. (n.d.). Develop with Vuforia Engine. Haettu 7.2.2019 osoitteesta <https://vuforia.com/content/vuforia/en/tools-and-resources.html>

Wikitude. (n.d.). Wikitude Augmented Reality SDK. Haettu 10.2.2019 osoitteesta <https://www.wikitude.com/products/wikitude-sdk/>

Wikitude Support a. (n.d.). Introduction to the Wikitude SDK. Haettu 9.2.2019 osoitteesta <https://www.wikitude.com/external/doc/documentation/latest/unity/>

Wikitude Support b. (n.d.). Object and Scene Recognition. Haettu 10.2.2019 osoitteesta <https://www.wikitude.com/external/doc/documentation/latest/android/objecttracking.html#object-and-scene-recognition>