

Terhi Lahtinen

# LISÄTYN TODELLISUUDEN ANDROID- MOBIILIPELIN KEHITTÄMINEN UNITY- PELIMOOTTORILLA

Opinnäytetyö

Liiketalouden ammattikorkeakoulututkinto

Tietojenkäsittelyn koulutus

2022



**Kaakkois-Suomen  
ammattikorkeakoulu**

Tutkintonimike	Tradenomi (AMK)
Tekijä	Terhi Lahtinen
Työn nimi	Lisätyn todellisuuden Android-mobiilipelin kehittäminen Unity-pelimoottorilla
Toimeksiantaja	Xamk / Game Studios -hanke
Vuosi	2022
Sivut	32 sivua
Työn ohjaaja	Jukka Selin

## TIIVISTELMÄ

Lisätty todellisuus mahdollistaa digitaalisten sisältöjen heijastamisen oikeaan maailmaan. Lisättyä todellisuutta on käytetty jo monta vuotta muun muassa viihteen ja opetuksen kehittämisessä. Teknologian työkalut ovat jatkuvassa muutoksessa, ja ne ovat kehittyneet alkuajoista helppokäyttöisemmiksi.

Opinnäytetyön tutkimusongelmana selvitetään, miten paikkatietoja hyödyntävän lisätyn todellisuuden mobiilipelin luominen toteutetaan. Työssä käydään läpi käytettävät työkalut ja ominaisuudet. Tässä opinnäytetyössä tullaan hyödyntämään Unityn tarjoamaa AR Foundation -kehystä, joka tarjoaa lisätyn todellisuuden luomiseen tarvittavat työkalut.

Xamk Game Studios -hankkeen tavoitteena on tutkia, kehittää ja testata erilaisia pelinkehitykseen liittyviä menetelmiä ja tekniikoita. Opinnäytetyössä on toteutettu hankkeelle toimeksiantona lisätyn todellisuuden pelin prototyyppi. Peliä voidaan käyttää pohjana lisätyn todellisuuden sovelluskehityksessä.

Opinnäytetyön alussa kerrotaan, kuinka lähdetään luomaan lisätyn todellisuuden sovellusta Unity-pelimoottorilla käyttäen AR Foundationia. Teoriaosudessa käydään läpi, mitä AR Foundation tarjoaa sekä kuinka Android-laitteelta saadaan sijaintitiedot ja miten niitä voidaan hyödyntää. Lopuksi käydään läpi lisätyn todellisuuden pelin kehittäminen ja pohditaan, kuinka sitä voisi lähteä parantamaan.

**Asiasanat:** lisätty todellisuus, mobiilipelit, paikkatietojärjestelmät, pelimoottorit, Android

Degree title	Bachelor of Business Administration
Author	Terhi Lahtinen
Thesis title	Developing an augmented reality mobile game for Android in Unity game engine
Commissioned by	Xamk / Game Studios project
Time	December 2022
Pages	32 pages
Supervisor	Jukka Selin

## ABSTRACT

Augmented reality enables new ways to offer entertainment and services. The objective of the thesis was to find out how an augmented reality mobile game was created with the Unity game engine. The results of the thesis benefitted the commissioning project by increasing the understanding of augmented reality. A prototype of an augmented reality mobile game was created for the thesis, and it can be used by the thesis commissioner as a basis for further development. The game would also utilize geographical data in its functions.

The first task was to figure out how to create an augmented reality mobile game in Unity. In this task all the necessary tools and changes for enabling augmented reality for the project were researched. The next step was to figure out how to get and utilize geographical location data from an Android mobile device and use it in the Unity mobile game. Finally, a prototype for the actual augmented reality mobile game was developed.

Augmented reality can be used in many ways for game development. The AR Foundation framework enables the use of augmented reality features in applications created with Unity. Results of the thesis can be used by the commissioner. The results of the thesis help increase the commissioner's own knowledge of augmented reality mobile game development.

**Keywords:** augmented reality, mobile games, geographical information system, game engines, Android

## SISÄLLYS

1	JOHDANTO.....	5
2	LISÄTYN TODELLISUUDEN ANDROID-MOBIILISOVELLUKSEN KEHITTÄMINEN UNITY-PELIMOOTTORILLA .....	6
2.1	Uuden Unity-projektin aloittaminen .....	6
2.2	Lisätyn todellisuuden pakettien lisääminen projektiin .....	8
2.3	Projektin asetusten konfigurointi lisättyä todellisuutta varten .....	10
2.4	Lisätyn todellisuuden sovelluksen luonti .....	12
3	AR FOUNDATION TARJOAA LISÄTYN TODELLISUUDEN OMINAISUUDET UNITY- PROJEKTEIHIN.....	14
3.1	Pintojen tunnistaminen ympäristöstä plane managerin avulla .....	15
3.2	Virtuaalisten sisältöjen kiinnittäminen 2D-kuviin Image tracking -ominaisuuden avulla	16
3.3	AR Foundation sisältää testaustyökalun AR-sovellusten kehittämiseksi .....	17
4	ANDROID-LAITTEEN SIJAINNITETÖJEN HYÖDYNTÄMINEN UNITY- PELIMOOTTORILLA .....	18
4.1	Sijaintitietojen saaminen Android-laitteelta .....	18
4.2	Valmiiden karttatyökalujen hyödyntäminen Unity-projektissa .....	20
5	OMAN AR-PELIN LUOMINEN.....	20
5.1	Projektin suunnittelu .....	21
5.2	Sijaintitietojen hyödyntäminen mobiilipelissä .....	21
5.3	Toimintojen rakentaminen .....	22
5.3.1	Kerättävien peliobjektien luominen .....	23
5.3.2	Pelin kohteiden luominen.....	24
5.3.3	Pelin käytettävyyden kehittäminen.....	26
5.4	Pelin testaus .....	29
6	YHTEENVETO .....	30
	LÄHTEET.....	32

## 1 JOHDANTO

Lisätty todellisuus (AR tai augmented reality) tarkoittaa digitaalisen sisällön lisäämistä oikeaan maailmaan. Lisätyn todellisuuden avulla voidaan heijastaa mikä tahansa virtuaalinen näkymä fyysiseen maailmaan mobiililaitteen tai lasien avulla. Lisätyn todellisuuden sovelluksia on ollut jo useita vuosia, ja työkalut, joilla lisätyn todellisuuden kokemuksia luodaan, ovat kehittyneet alkua ajoista huomattavasti.

Unity-pelimoottori sisältää omat työkalut lisätyn todellisuuden sovellusten luomiseen. Yksi tunnetuimmista työkaluista on Unityn oma AR Foundation -kehys, jonka avulla lisätyn todellisuuden sovelluksia voidaan ohjelmoida useille eri alustoille yhdellä koodipohjalla. Opinnäytetyössä lisätyn todellisuuden mobiilipeli tullaan toteuttamaan AR Foundationin avulla.

Opinnäytetyön tavoitteena on saada toimeksiantajalle prototyyppi lisätyn todellisuuden pelistä, jota pystyy tarvittaessa myöhemmin jatkokehittämään. Toimeksiantajana opinnäytetyölle on Xamkin Game Studios -hanke, jossa työn tuloksia tullaan hyödyntämään. Tavoitteena hankkeella on tutkia, kehittää ja testata erilaisia pelinkehitykseen liittyviä menetelmiä ja tekniikoita. Siten opinnäytetyön aihe on hankkeelle hyödyllinen.

Tutkimusongelmana opinnäytetyössä tullaan selvittämään, miten Unity-pelimoottorilla luodaan lisätyn todellisuuden mobiilipelejä Android-laitteelle. Unityn käyttöä opinnäytetyössä ei tulla esittelemään sen tarkemmin. Ainoastaan kaikki mitä lisätyn todellisuuden Android-sovellukseen vaaditaan. Opinnäytetyössä esitellään myös toimintoja, joita ei ole käytetty pelin teossa.

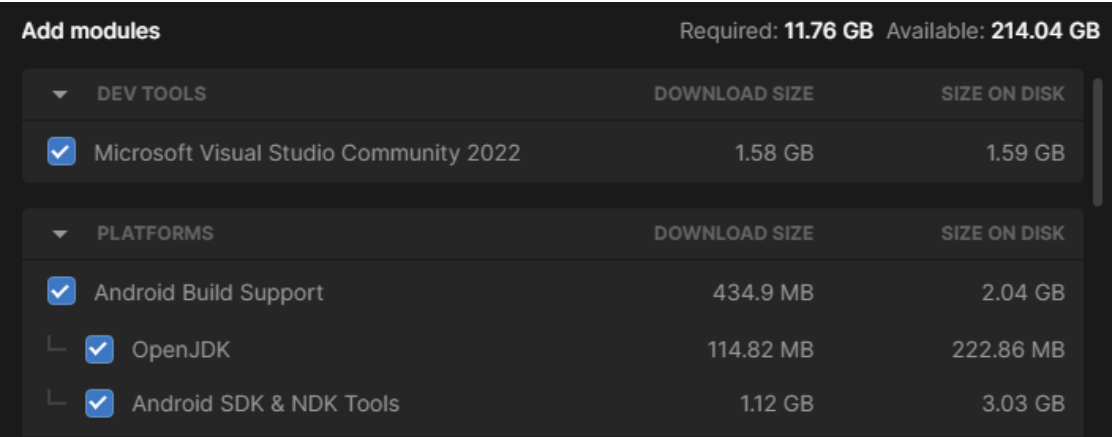
Opinnäytetyön luvuissa 2–4 käsitellään työn teoreettista osuutta. Näissä luvuissa neuvotaan lisätyn todellisuuden sovellusten perustan luonti. Lisäksi kerrotaan AR Foundationin ominaisuuksista ja selvitetään sijaintitietojen saantia Android-laitteelta. Luvussa 5 käydään läpi tutkimusongelmana kehitetyn lisätyn todellisuuden mobiilipelin prototyypin kehittäminen ja testaus. Viimeisenä lukuna on yhteenveto, jossa pohditaan prototyypin kehitysideoita ja AR-tekniikan mahdollisuuksia.

## 2 LISÄTYN TODELLISUUDEN ANDROID-MOBIILISOVELLUKSEN KEHITTÄMINEN UNITY-PELIMOOTTORILLA

Tässä luvussa kerrotaan kaikki tarvittava lisätyn todellisuuden sovelluksen luomista varten Unity-pelimoottorilla. Lisätyn todellisuuden sovelluksissa hyödynnetään laitteen omia ominaisuuksia. Näiden ominaisuuksien käyttöönotto vaatii tavallisen projektin aloittamisen lisäksi laajennusten asentamista ja asetusten konfigurointia. Opinnäytetyössä käytetään AR Foundationia, josta kerrotaan enemmän luvussa 3.

### 2.1 Uuden Unity-projektin aloittaminen

Ennen projektin aloittamista on asennettava tarpeiden mukainen Unity-versio ja asennettaessa on valittava tarvittavat moduulit. Android-sovellusten kehittäminen vaatii kuvassa 1 näkyvät Unityn Android-työkalut. Moduuleja voidaan asentaa Unity-versioihin myös jälkikäteen. Versioiden moduuleja saadaan asennettua Unity Hubin Installs -välilehdeltä jokaisen version omista asetuksista.



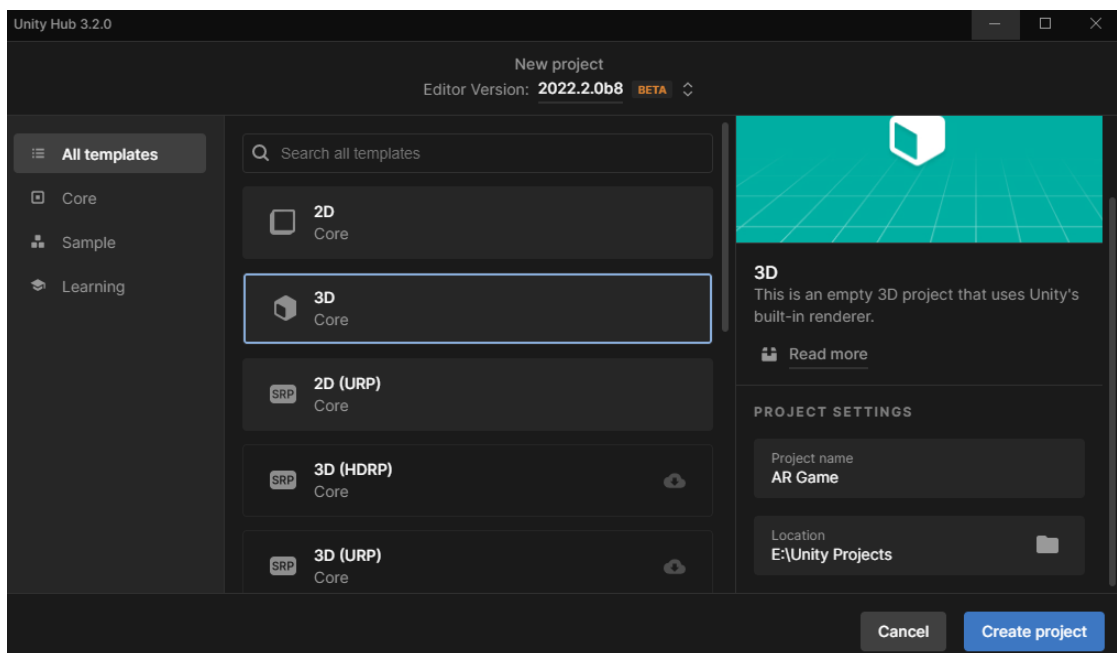
Add modules		Required: 11.76 GB	Available: 214.04 GB
▼	DEV TOOLS	DOWNLOAD SIZE	SIZE ON DISK
<input checked="" type="checkbox"/>	Microsoft Visual Studio Community 2022	1.58 GB	1.59 GB
▼	PLATFORMS	DOWNLOAD SIZE	SIZE ON DISK
<input checked="" type="checkbox"/>	Android Build Support	434.9 MB	2.04 GB
<input checked="" type="checkbox"/>	OpenJDK	114.82 MB	222.86 MB
<input checked="" type="checkbox"/>	Android SDK & NDK Tools	1.12 GB	3.03 GB

Kuva 1. Tarvittavien moduulien asennus

Kun Unity-versio ja tarvittavat moduulit ovat asennettu, voidaan aloittaa projektin luominen. Uuteen projektiin on valittava Unity-versio sekä template eli pohja, joita käytetään projektissa. Templatet ovat Unity-projektien pohjia, joissa on valmiiksi esiasennettuja paketteja, jotka sisältävät projektille tarpeellisia ominaisuuksia. Unity tarjoaa valmiit templatet kaksi- ja kolmiulotteisille peleille sekä tarkemmin tietyille pelien genreille, kuten ajopeleille tai ensimmäisen persoonan peleille.

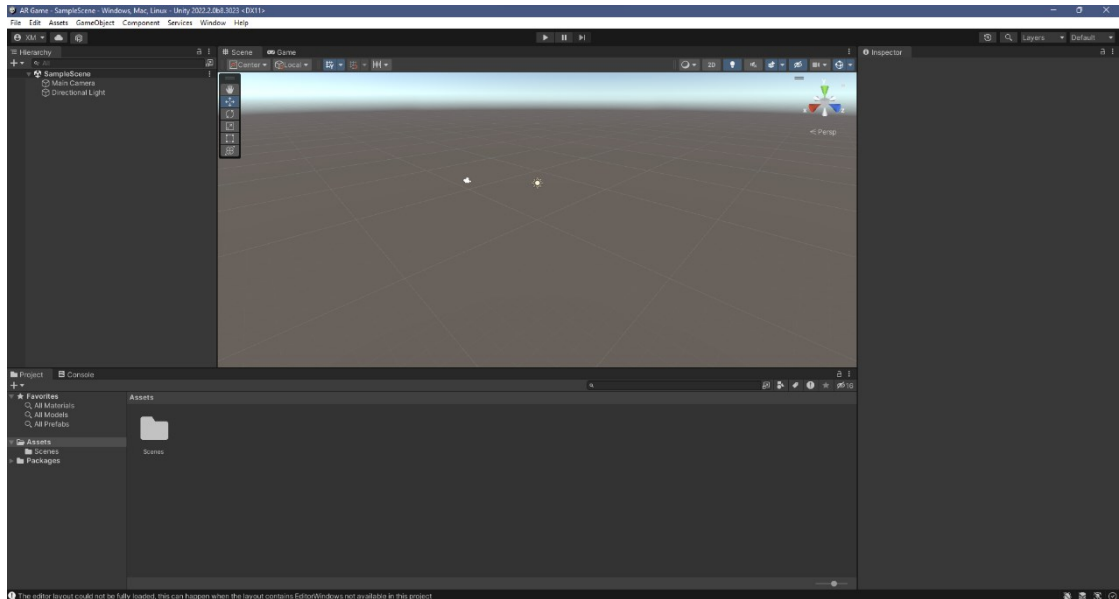
Kuvasta 2 näkee opinnäytetyössä käytetyn version sekä templatien. Unityssä on olemassa valmis AR-template lisätyn todellisuuden projekteille. Valmiin AR-templatien mukana voi kuitenkin tulla projektille turhia paketteja ja niiden versiot eivät välttämättä ole uusimpia. Projekti on helpompi aloittaa tyhjällä 3D-templatella ja asentaa tarvittavat paketit sekä konfiguroida projektin asetukset kuntoon itse.

Yleensä on suositeltavaa käyttää virallista, pitkään tuettavaa (Long Term Support/ LTS) Unity-version julkaisua, joka opinnäytetyön kirjoittamisen aikaan on 2021.3.11f1. Opinnäytetyössä on kuitenkin käytetty vielä beta-vaiheessa olevaa 2022.2.0b8-versiota. Jostain syystä 2021-version package manager ei asenna suoraan AR Foundationin 5.0.2-versiota, joka on uusien lisätyn todellisuuden ominaisuudet tuova paketti.



Kuva 2. Uuden projektin luominen

Projekti on vielä nimettävä ja sille on valittava tiedostosijainti, johon projekti tallentuu. Näiden jälkeen projekti luodaan painamalla Create project -painiketta. Unity lataa projektia hetken aikaa, jonka jälkeen näkyviin pitäisi tulla kuvan 3 mukainen aloitusnäky.



Kuva 3. Projektin aloitusnäky

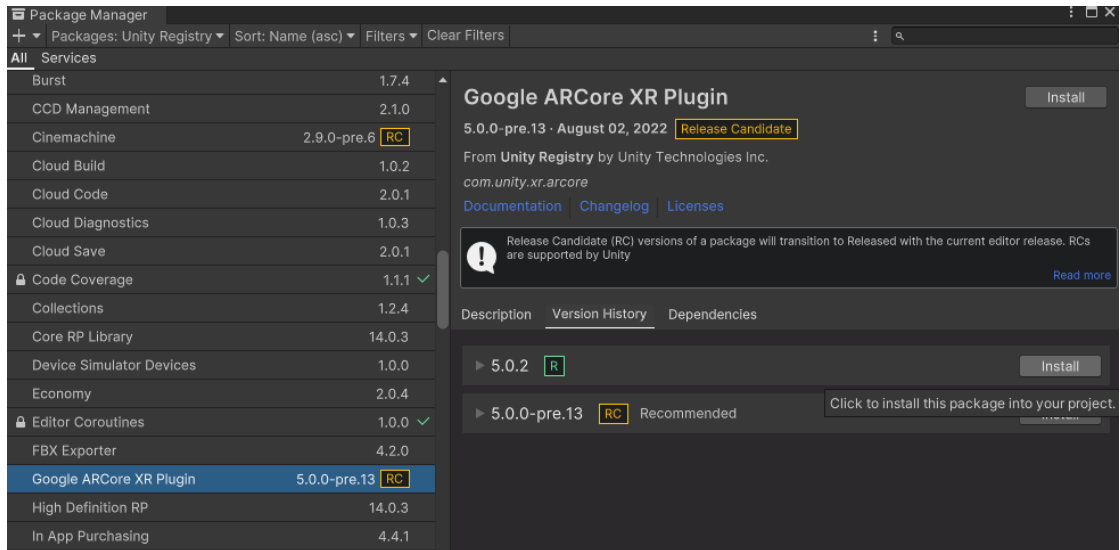
Projekti ei vielä tässä vaiheessa tue lisätyn todellisuuden ominaisuuksia. Seuraavaksi projektiin on lisättävä tarvittavat paketit, jotta lisätyn todellisuuden sovellusta on mahdollista ryhtyä tekemään.

## 2.2 Lisätyn todellisuuden pakettien lisääminen projektiin

Package managerista on asennettava paketit, jotka tuovat projektiin lisätyn todellisuuden ominaisuuksia. Asennettavat paketit ovat AR Foundation, Google ARCore XR Plugin ja XR Plugin Management. Paketteja asennettaessa on valittava tarpeenmukaiset versiot, jotka riippuvat käytetyistä Unity-versiosta ja halutuista ominaisuuksista.

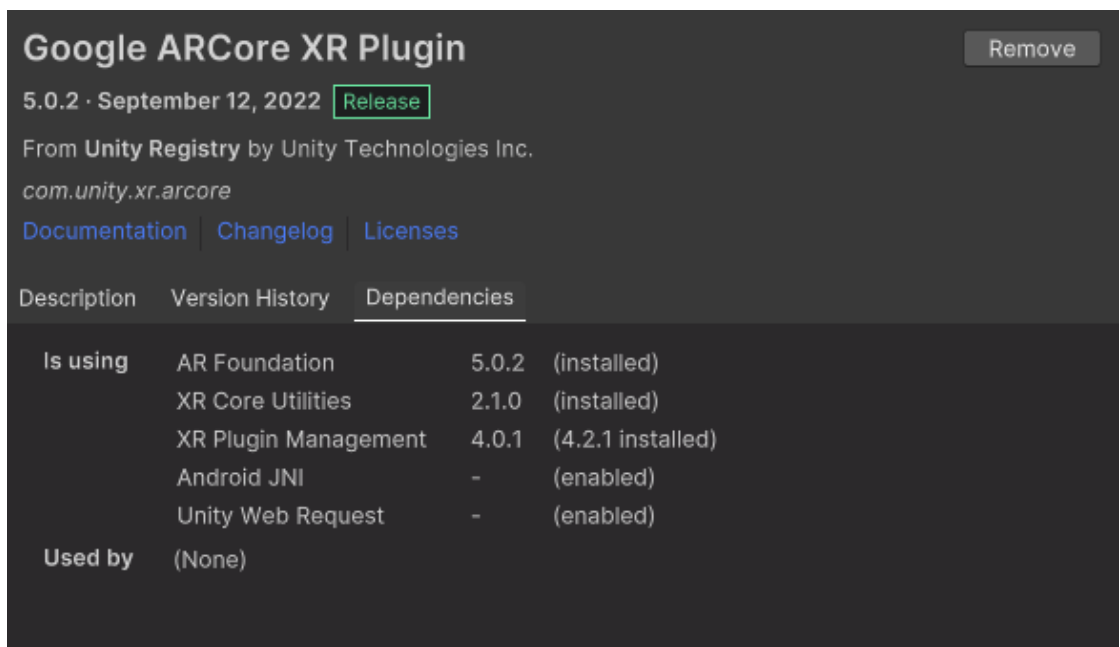
Esimerkiksi AR Foundation 5.0.2 -versio tuo uutena ominaisuutena simulaatioympäristön, jonka avulla AR ominaisuuksia voidaan testata editorissa. Ennen tätä lisätyn todellisuuden projekti on pitänyt testata asentamalla sovellus laitteelle. Opinnäytetyössä käytetyssä Unity-versiossa tarvittava uusin julkaistu versio on kuvan 4 mukaisesti valitun paketin Version History -välilehdellä.





Kuva 4. Paketin lisääminen

Opinnäytetyössä käytettävä Google ARCore XR Plugin asentaa automaattisesti myös AR Foundationin ja XR Plugin Managementin. Paketin Dependencies-välilehdeltä näkee, mitä muita paketteja asennettava paketti käyttää (kuva 5). On kuitenkin hyvä käydä katsomassa tarvittavat paketit vielä erikseen versiopäivityksiä varten, jotta saa uusimmat versiot käyttöön.



Kuva 5. Dependencies-välilehti

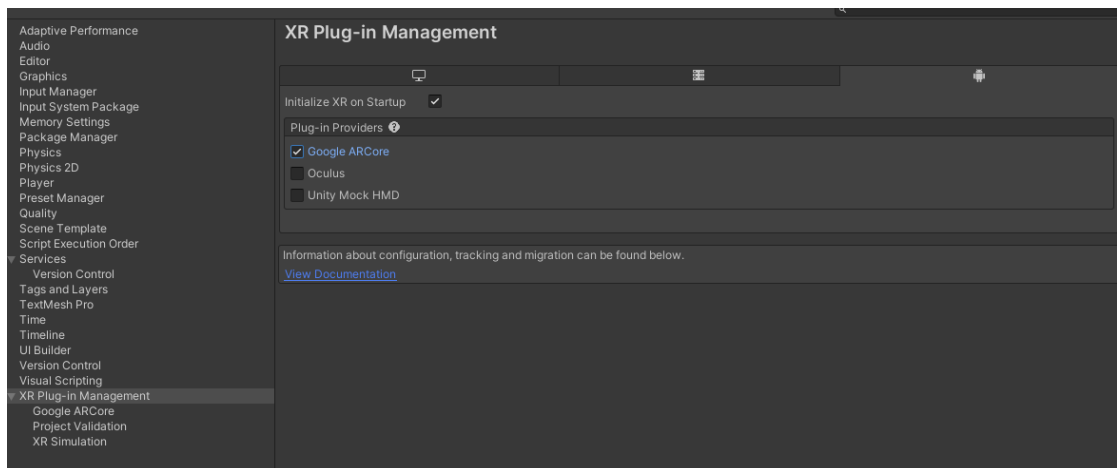
Pakettien asentamisen lisäksi projektin asetuksia on vielä konfiguroitava.

Opinnäytetyössä kehitetty lisätyn todellisuuden peli rakennetaan Android-laitteille, mikä vaatii muun muassa lisäosan Android-sovelluksen kääntämiseksi.

## 2.3 Projektin asetusten konfigurointi lisättyä todellisuutta varten

Projektin asetuksia on konfiguroitava, jotta sovellus voidaan rakentaa eri laitteille. Asetukset riippuvat siitä, mille laitteelle sovellusta rakennetaan. Opinäytetyössä peli rakennetaan Android-laitteelle ja seuraavaksi käydään läpi tarpeellisten asetusten konfigurointi.

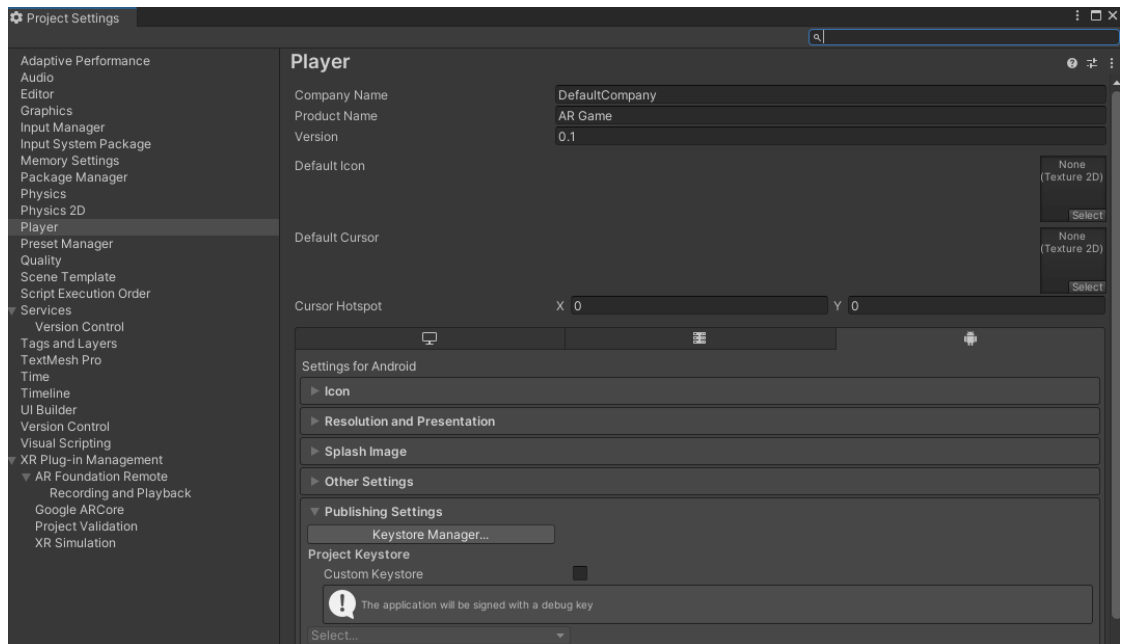
Project Settings -ikkunassa XR Plug-in Management -välilehdeltä on valittava Android-sovelluksen asetukset (kuva 6). Asennettavaksi liitännäiseksi valitaan Google ARCore. Tämä mahdollistaa sovelluksen käynnön Android-puhelimille.



Kuva 6. Google ARCoren valitseminen

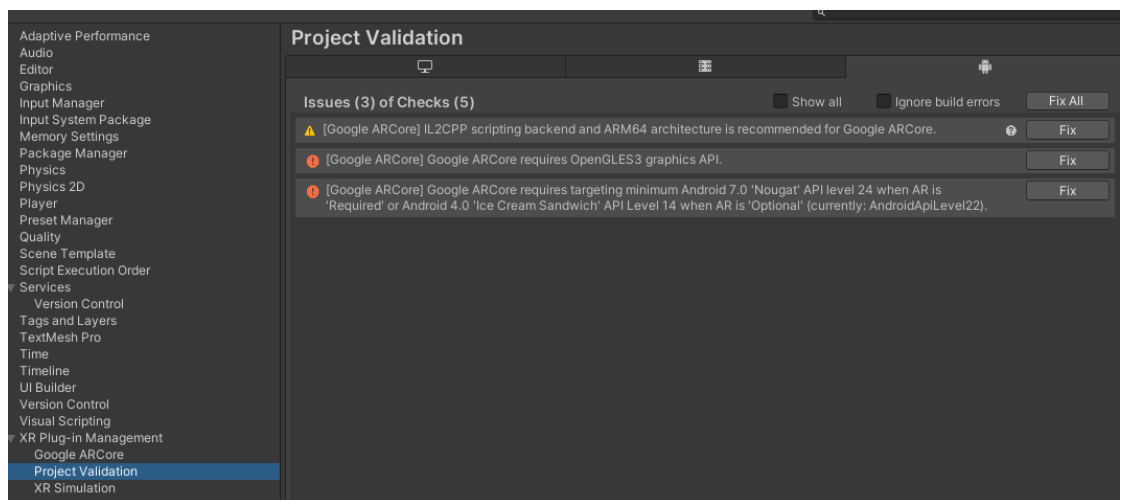
Project Settings -ikkunasta on muutettava Player Android-välilehden asetuksia (kuva 7). Render-asetuksista on otettava Auto Graphics API pois käytöstä, jotta Vulkan on mahdollista poistaa käytettävien grafiikkarajapintojen listasta.

Android-versio on määriteltävä Minimum API Level -kohdasta olemaan Android 7.0 (API level 24) tai uudempi. Vanhemmat Android-versiot eivät tue lisätyn todellisuuden ominaisuuksia. Scripting Backend -asetus on muutettava arvoon IL2CPP (Intermediate Language To C++), jolloin sovellus tukee laajemmin eri alustoja (IL2CPP Overview 2022). Target Architectures -asetuksista on aktivoitava ARM64, jolloin sovellus tukee myös 64-bittistä arkkitehtuuria.



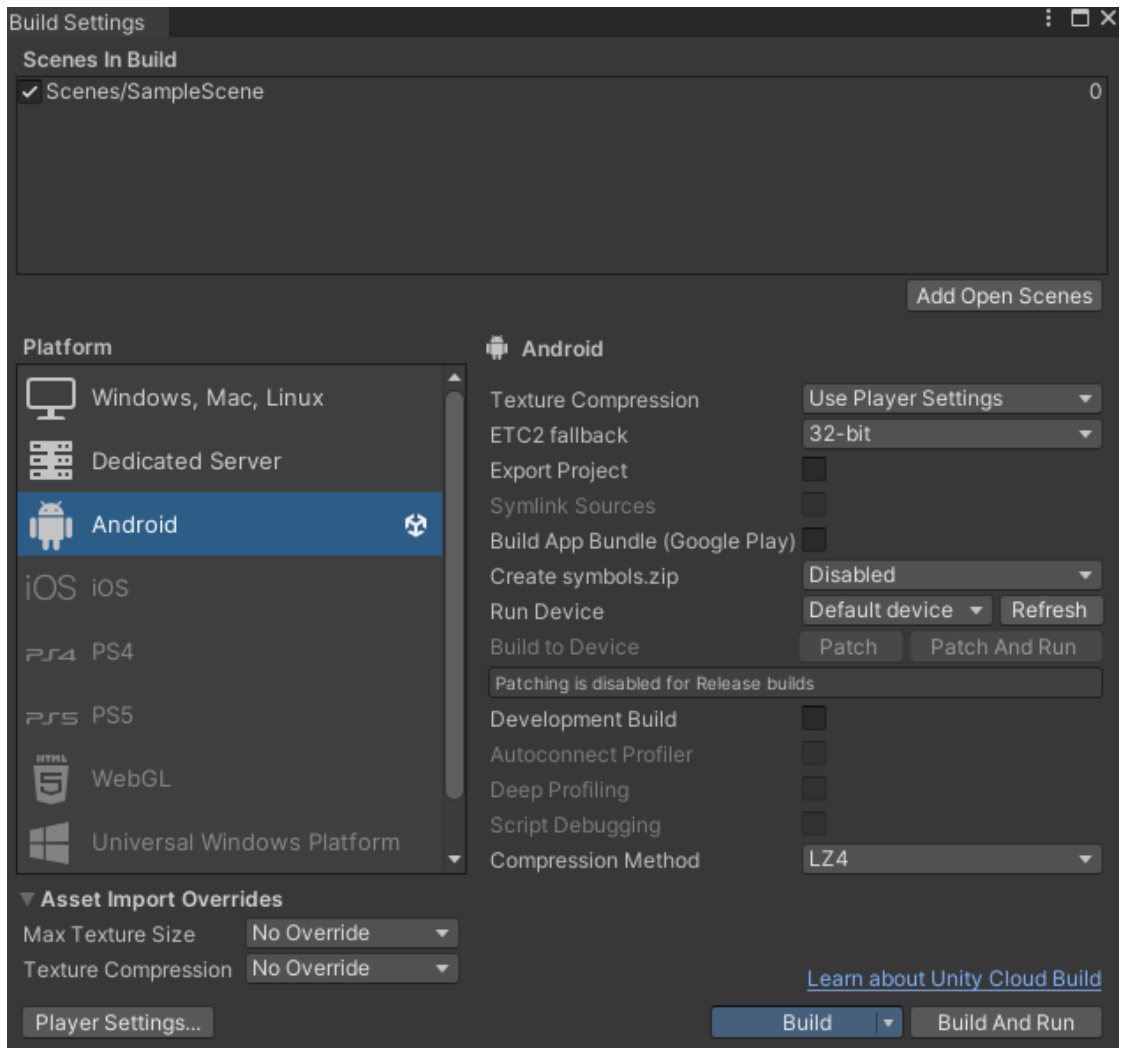
Kuva 7. Player Settings

Project settings -ikkunan XR Plug-in Managementin alta löytyy Project Validation -välilehti, josta näkee projektin asetuksiin vaikuttavat ongelmat. Myös tätä kautta on mahdollista korjata asetukset niin, että projekti toimii. Kuvasta 8 näkee samat ongelmat, jotka ylempänä on korjattu käsin.



Kuva 8. Projektin ongelmien korjaaminen

Projektin rakentamista varten on mentävä Build Settings -asetuksiin ja vaihdettava platform sekä lisättävä vaadittavat skenet (kuva 9). Android-laitteelle tehtäessä sovellusta on platformiksi valittava myös Android.



Kuva 9. Android buildin valinta

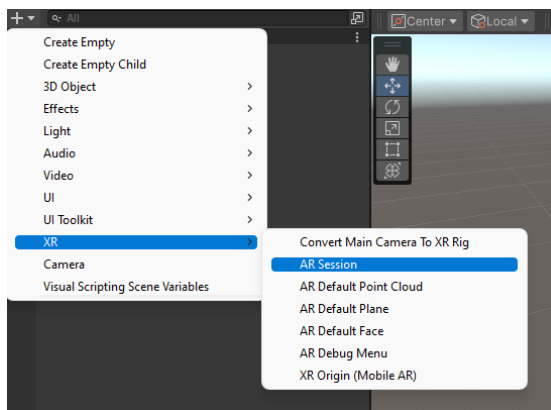
Project Settings -asetuksissa on paljon muitakin eri asetuksia, joita on mahdollista hyödyntää projektissa. Player Settings -välilehdeltä on mahdollista muuttaa muun muassa rakennetun sovelluksen aloitusruudun asetuksia.

## 2.4 Lisätyn todellisuuden sovelluksen luonti

Kun tarvittavat paketit on asennettu ja asetukset konfiguroitu, on mahdollista ryhtyä luomaan lisätyn todellisuuden sovellusta. Ensimmäisenä skeneen on lisättävä GameObject-valikon XR-kohdan alta AR Session- sekä XR Origin - peliobjektit, jotka kuuluvat AR Foundation -pakettiin (kuva 10). Skenen alkuperäinen Main Camera -peliobjekti voidaan poistaa, koska XR Origin sisältää Camera Offset -lapsiobjektin, jonka lapsiobjektina on oma lisätyn todellisuuden toimintoja tukeva Main Camera. AR Session ja XR Origin ovat tarpeellisia peliobjekteja, kun tehdään lisätyn todellisuuden sovellusta hyödyntäen AR Foundation -kirjastoa.

AR Session -komponentti hallitsee lisätyn todellisuuden toimintoja. Komponentin voi kytkeä päälle ja pois päältä vapaasti sovelluksen käynnissä olon aikana. Mikäli komponentti on pois päältä, ei sovellus kykene skannaamaan ympäristön yksityiskohtia. Käynnistämällä komponentin uudestaan sovellus pyrkii palauttamaan aiemmin skannatun datan. AR Session komponentteja voi olla vain yksi kerrallaan skenessä. (About AR Foundation 2022.)

XR Origin -komponentti muuntaa skannattavien yksityiskohtien sijainnin, kierroksen ja skaalan lopulliseen muotoonsa skenessä. AR-sessiolla on oma paikallinen avaruus, johon skannattavien yksityiskohtien peliobjektit luodaan. XR Origin -komponentti muuntaa näiden yksityiskohtien paikallisen avaruuden sijaintitiedot (transformit) Unity-skenen maailma-avaruuteen. (About AR Foundation 2022.)



Kuva 10. AR Session

Kun kaikki toimivaan lisätyn todellisuuden sovellusta varten vaadittavat muutokset asetuksiin ja skeneen on tehtynä, voidaan sovellusta ryhtyä tekemään. Mikäli tässä vaiheessa rakennettaisiin sovellus ja avattaisiin se Android-laitteella, näkyisi siinä vain kameran kuva. Android-laitteella on muistettava sallia kameran käyttö sovellusta käytettäessä, muuten lisätyn todellisuuden sovellukset eivät toimi laitteella. Myös sijaintitiedot on sallittava, mikäli sovellus tarvitsee niitä toimiakseen.

### 3 AR FOUNDATION TARJOAA LISÄTYN TODELLISUUDEN OMINAISUUDET UNITY-PROJEKTEIHIN

AR Foundation tarjoaa mahdollisuuden tehdä lisätyn todellisuuden sovelluksia eri alustoille Unity-pelimoottorilla. AR Foundation on joukko objekteja (Monobehavioureja) ja rajapintoja, jotka mahdollistavat eri laitteilla toimimisen. On kuitenkin tarkistettava, että laite tukee vaadittavia ominaisuuksia. Vaikka AR Foundation ei itsestään toteuta lisätyn todellisuuden ominaisuuksia, tarjoaa se rajapinnan, joka tukee useita eri alustoja. (About AR Foundation 2022.)

Eri laitteiden alustat tukevat hieman eri ominaisuuksia. Taulukossa 1 on lista eri ominaisuuksista ja alustoista, eli mitkä tukevat mitäkin ominaisuuksia. Tässä opinnäytetyössä hyödynnetään ARCoren tukemia ominaisuuksia kuten Device Trackingiä.

Taulukko 1. Lista ominaisuuksista mitä eri alustat tukevat (AR Foundation 2022)

	ARCore	ARKit	OpenXR
Device tracking	✓	✓	✓
Plane tracking	✓	✓	
Point clouds	✓	✓	
Anchors	✓	✓	✓
Light estimation	✓	✓	
Environment probes	✓	✓	
Face tracking	✓	✓	
2D Image tracking	✓	✓	
3D Object tracking		✓	
Meshing		✓	✓
2D & 3D body tracking		✓	
Collaborative participants		✓	
Human segmentation		✓	
Raycast	✓	✓	
Pass-through video	✓	✓	
Session management	✓	✓	✓
Occlusion	✓	✓	

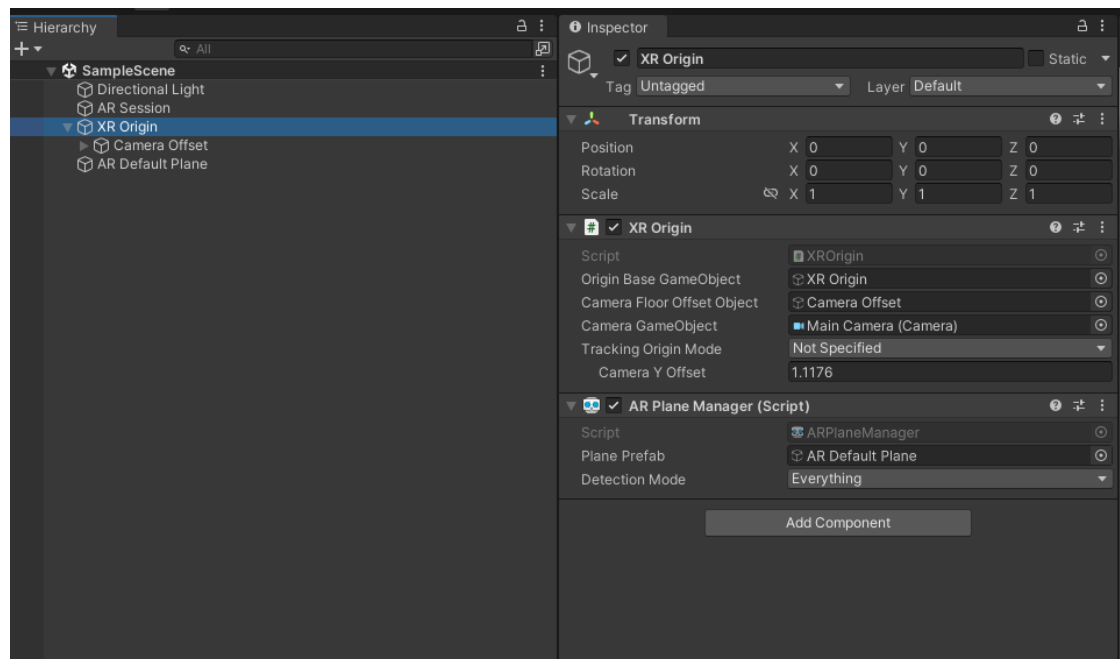
AR Foundation sisältää erilaisia manager-komponentteja, jotka tarjoavat lisätyn todellisuuden ominaisuuksien hyödyntämiseen ja niiden käyttäytymisen ohjelmointiin valmiita metodeja. Manager-komponentit lisätään skenessä XR

Origin -peliobjektiin. Erilaisia manager-komponentteja ovat muun muassa AR plane manager, AR tracked image manager ja AR face manager.

### 3.1 Pintojen tunnistaminen ympäristöstä plane managerin avulla

Plane manager -komponentti tunnistaa oikean maailman tasaisia pintoja ja luo niille peliojekteja. Komponentti sisältää Detection Moden, josta voidaan valita, minkälaisia pintoja tunnistetaan. Vaihtoehtoina ovat ei mitään, vaakasuorat, pystysuorat tai molemmat pinnat. (AR Plane manager 2022.)

Plane manager otetaan käyttöön lisäämällä skenen hierarkiaan AR Default Plane -peliobjekti XR-välilehdeltä. XR Origin -peliobjektiin lisätään AR Plane Manager -komponentti. Komponentin sisältämään Plane Prefab -kohtaan vedetään aikaisemmin luotu AR Default Plane -peliobjekti. (Kuva 11.)



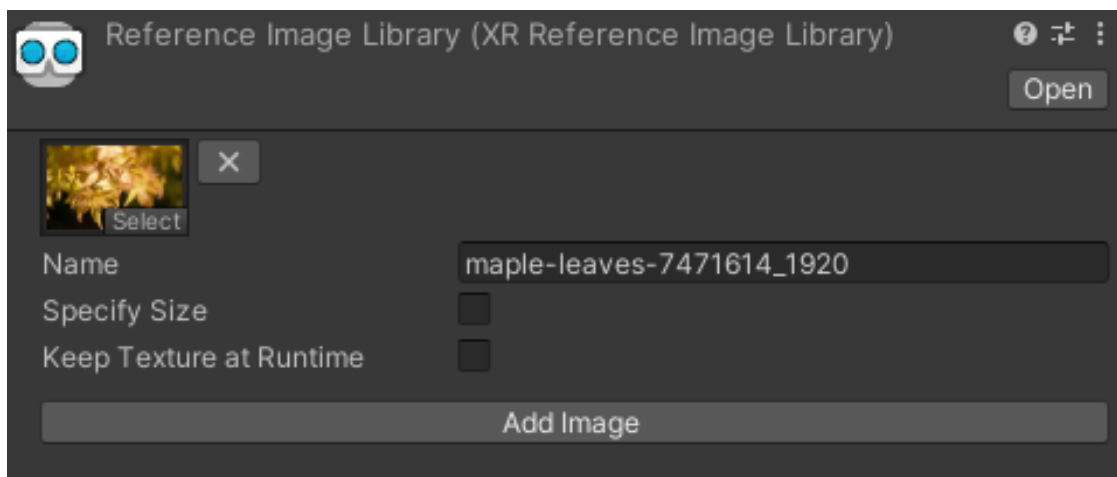
Kuva 11. Plane manager

Plane managerin luomien visuaalisten peliojektien materiaali on oletuksena tasainen väri, josta ei näe läpi. Materiaalia on mahdollista muuttaa ja laittaa itselle sopivamman näköiseksi. Parempi vaihtoehto voisi olla esimerkiksi pisteet läpinäkyvällä taustalla, jolloin olisi helpompi nähdä, missä kohtaa pinta on.

### 3.2 Virtuaalisten sisältöjen kiinnittäminen 2D-kuviin Image tracking -ominaisuuden avulla

Image tracking -ominaisuuden avulla voidaan seurata 2D-kuvia ja luoda peliobjekteja ympäristöstä tunnistettuihin kuviin. Tätä ominaisuutta voidaan hyödyntää muun muassa viihdesovelluksissa ja mainonnassa. 2D-kuvista voidaan esimerkiksi skannata hahmoja, jotka heräävät eloon 3D-malleina lisätyn todellisuuden tasolle.

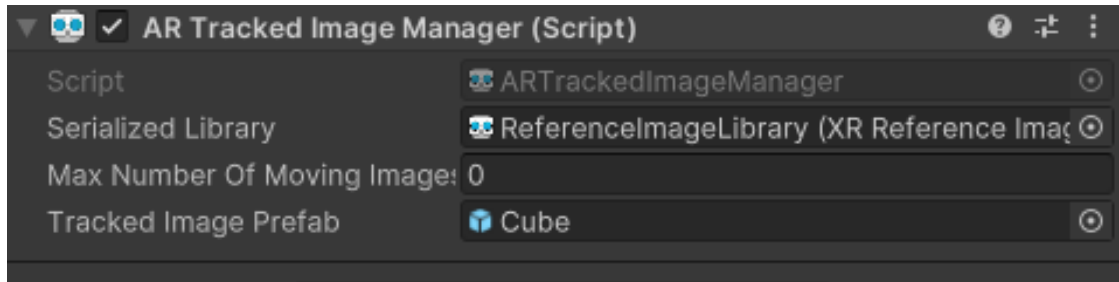
Kuvan tunnistamista varten on ensin määriteltävä kirjasto halutuista kuvista, joita Image tracking -komponentti yrittää tunnistaa ympäristöstä. Ensiksi Reference Image Library -sisältö on luotava Assets-kansioon, jotta kuvia voidaan lisätä siihen (kuva 12). Kuville voidaan määrittää nimi sekä fyysinen koko. Koon määrittäminen helpottaa kuvien tunnistamista. (AR tracked image manager 2022.)



Kuva 12. Reference Image Library

XR Origin -peliobjektiin lisätään komponentti AR Tracked Image Manager, johon lisätään aikaisemmin luotu kuvakirjasto kohtaan Serialized Library. Lopuksi vielä lisätään Tracked Image Prefab -kohtaan omavalintainen prefab, joka sitten ilmestyy sovellukseen kuvien kohdalle. (Kuva 13.) On myös mahdollista ohjelmoida itse toiminnallisuus, jossa määritellään, mitä tunnistetuille kuville tehdään. On esimerkiksi mahdollista määrittää jokaiselle kuvalle oma peliobjekti. (AR tracked image manager 2022.)





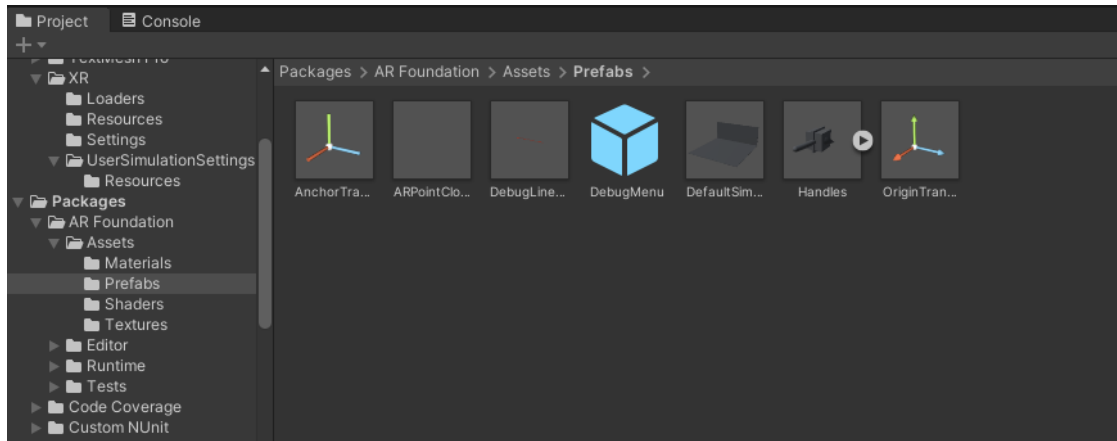
Kuva 13. AR Tracked Image Manager

Ominaisuuden avulla on myös mahdollista seurata kuvia, jotka ovat liikkeessä. Sitä varten voidaan rajoittaa liikkuvien kuvien määrää. Liikkuvien kuvien määrä asetetaan AR Tracked Image Manager -komponentissa lisäämällä haluttu luku kohtaan Max Numbers Of Moving Images.

### 3.3 AR Foundation sisältää testaustyökalun AR-sovellusten kehittämiseksi

AR Foundation sisältää versiosta 5.0.2 lähtien vianetsintään ja testaukseen Debug menu -nimisen työkalun, joka helpottaa sovelluksen kehittämistä. Työkalulla on muun muassa mahdollista tarkkailla ruudunpäivitystä ja session konfiguraatiota sekä visualisoida virtuaalisen maailman origo. Debug-valikon saa lisättyä mihin tahansa paikkaan skenessä, jossa on ARSession, XROrigin ja Camera. (AR Foundation Debug menu 2022.)

Työkalu lisätään skeneen valmiina prefab-objektina, joka sisältää toiminnallisuudet tarjoavan skriptin sekä käyttöliittymän työkalun käyttämistä varten. AR Debug Menu -prefabin voi lisätä skeneen GameObject-valikosta samasta listasta kuin AR Session- ja XR Origin -prefabit. Mikäli debug-valikon lisääminen ei toimi GameObject-valikosta, niin se on mahdollista lisätä myös Project-kansion kautta. Työkalun voi lisätä menemällä Project-ikkunan kansiorakenteessa kansioon Packages > AR Foundation > Assets > Prefabs ja vetämällä Debug-Menu-prefabin hierarkiaan (kuva 14).



Kuva 14. Debug menun tiedostosijainti

Mobiilisovellusta testattaessa voi olla tärkeää nähdä sovelluksen käynnissäolon aikana komentoriviin tulevat lokitiedot. AR Foundationin testaustyökalu ei kuitenkaan näytä komentorivin lokitietoja käyttöliittymässään. Tarvittaessa tällainen ominaisuus on vielä rakennettava itse ja tästä syystä opinnäytetyössä ei ole hyödynnetty Debug menu -työkalua.

#### 4 ANDROID-LAITTEEN SIJAITITETOJEN HYÖDYNTÄMINEN UNITY-PELIMOOTTORILLA

Sijaintitietoja voidaan hyödyntää peleissä monin eri tavoin muun muassa näyttämällä käyttäjän sijainti oikeassa maailmassa. Sijaintitietojen avulla voidaan myös tarkastaa ympäristöä ja kertoa käyttäjälle, onko lähellä mielenkiintoisia kohteita, joita halutaan näyttää. (Glover & Linowes 2019, 311.)

Unity-pelimoottorilla tehdyissä sovelluksissa voi hyödyntää sijaintitietoja eri tavoin. Sijaintitietojen avulla voidaan muun muassa tarkastella käyttäjän sijaintia, jota voi hyödyntää erilaisissa sovelluksissa. Unityllä on mahdollista rakentaa esimerkiksi karttasovellus, jossa käyttäjän sijainti näkyy kartalla reaaliaikaisesti. Tässä opinnäytetyössä tullaan hyödyntämään sijaintitietoja verrattaessa käyttäjän sijaintia pelissä oleviin kohteisiin, jotka ovat sijainteja oikeassa maailmassa.

##### 4.1 Sijaintitietojen saaminen Android-laitteelta

Unityn dokumentaatiot sisältävät valmiin koodin, jolla saadaan tieto laitteen viimeisestä sijainnista (kuva 15). Koodin avulla saadaan tieto laitteen sijainnin

pituus- ja leveysasteista sekä korkeudesta. Tieto pituus- ja leveysasteista annetaan desimaalilukuina. Koodi lisätään projektiin luomalla ensin C#-skripti, joka liitetään tyhjään peliobjektiin komponenttina. Luotuun skriptiin kopioidaan dokumentaatiosta saatava koodi. (LocationService.Start 2022.)

```
public class TestLocationService : MonoBehaviour
{
    IEnumerator Start()
    {
        // Check if the user has location service enabled.
        if (!Input.location.isEnabledByUser)
            yield break;

        // Starts the location service.
        Input.location.Start();

        // Waits until the location service initializes
        int maxWait = 20;
        while (Input.location.status == LocationServiceStatus.Initializing && maxWait > 0)
        {
            yield return new WaitForSeconds(1);
            maxWait--;
        }

        // If the service didn't initialize in 20 seconds this cancels location service use.
        if (maxWait < 1)
        {
            print("Timed out");
            yield break;
        }

        // If the connection failed this cancels location service use.
        if (Input.location.status == LocationServiceStatus.Failed)
        {
            print("Unable to determine device location");
            yield break;
        }
        else
        {
            // If the connection succeeded, this retrieves the device's current location and displays it in the Console window.
            print("Location: " + Input.location.lastData.latitude + " " + Input.location.lastData.longitude + " " + Input.location.lastData.altitude + " " + Input.location.lastData.horizontalAccuracy);
        }

        // Stops the location service if there is no need to query location updates continuously.
        Input.location.Stop();
    }
}
```

Kuva 15. Koodi sijaintitietojen saamiseksi Android-laitteelta (LocationService.Start 2022)

Sijaintitietoja käyttävää sovellusta tehdessä voi olla hyödyllistä näyttää sijainnin tiedot sovelluksessa erillisessä käyttöliittymässä testaamista varten. Kuvan 16 print-komennot on muutettava siten, että print-komennon sijasta tiedot asetetaan käyttöliittymän tekstikomponenteille (kuva 16).

```
// If the service didn't initialize in 20 seconds this cancels location service use.
if (maxWait < 1)
{
    statusText.text = "Timed out";
    yield break;
}
```

Kuva 16. Tietojen asettaminen tekstikomponenttiin

Koodia on mahdollista muokata siten, että sijainnin tiedot päivittyvät sovelluksessa useammin. Oletuksena koodi päivittää sijaintitiedot 10 metrin välein. Metodin Input.location.Start()-parametreihin on lisättävä sijainnin tarkkuus metreinä sekä sijainnin päivittämiseen liikuttavan matkan määrä metreinä. (LocationService.Start 2022.)

## 4.2 Valmiiden karttatyökalujen hyödyntäminen Unity-projektissa

Monet lisätyn todellisuuden sovellukset hyödyntävät paikkatietojärjestelmää (GIS tai Geographical Information System), jonka avulla voidaan kerätä, tallentaa, hallita, analysoida ja esittää paikkatietoja. Monet paikkatietoihin perustuvat sovellukset hyödyntävät paikkatietojärjestelmää. Esimerkkeinä tällaisista sijaintia käyttävistä palveluista ovat Google Maps ja Pokémon Go. (Glover & Linowes 2019, 373.)

Useimmat sijaintitietoja hyödyntävät sovellukset tarjoavat käyttöliittymän ja kartan, josta käyttäjän sijaintia voidaan seurata. Sovellusten tekemistä varten on olemassa useita eri kehitystyökalukokoelmia. Nämä työkalut tarjoavat usein rajapintoja palveluntarjoajan järjestelmien hyödyntämiselle. Kehitystyökalukokoelmia karttatietojen hyödyntämiselle ovat muun muassa ArcGis ja Mapbox.

Mapbox sekä ArcGis tarjoavat Unitylle valmiit lisäosat kehitystyökalujen hyödyntämiselle. Molemmat tarjoavat muun muassa karttojen luomista ja paikkatietojen hyödyntämistä varten työkaluja sekä käyttöliittymän. (ArcGIS Maps... 2022; Maps SDK for Unity s.a.) Tässä opinnäytetyössä ei kuitenkaan tulla käyttämään karttatyökaluja.

## 5 OMAN AR-PELIN LUOMINEN

Opinnäytetyön kehittämistehtävänä on lisätyn todellisuuden mobiilipelin suunnittelu ja kehittäminen Unity-pelimootorilla. Peli hyödyntää Android-mobiililaitteelta saatavaa sijaintitietoa pelaajan sijainnin seuraamisessa. Peli vertaa pelaajan sijaintia ennalta määrättyjen paikkaperustaisten peliobjektien sijainteihin. Tässä opinnäytetyössä näistä peliobjekteista käytetään nimitystä kohde. Kun pelaaja on tarpeeksi lähellä kohdetta, hän voi kerätä siellä sijaitsevan peliobjektin.

Työssä selvitetään, miten lisätyn todellisuuden mobiilipeliä lähdetään tekemään ja kuinka mobiililaitteen sijaintitietoja voidaan hyödyntää. Tavoitteena on tuottaa hankkeelle lisätyn todellisuuden mobiilipelin prototyyppi, jonka päälle

toimeksiantaja voi myöhemmin toteuttaa omia ratkaisujaan. Toisena tavoitteena on lisätä hankkeessa AR-sovelluskehityksen osaamista.

## **5.1 Projektin suunnittelu**

Peli on toteutettu Unity-pelimootorilla, koska se mahdollistaa lisätyn todellisuuden sovelluksien luomisen ja tarjoaa kattavat dokumentaatiot. Lisätyn todellisuuden sovelluksen luomiseen Unity tarjoaa AR Foundation -kirjaston, joka mahdollistaa AR-ominaisuuksien lisäämisen projektiin helposti.

Unity-pelimootorilla on monia tapoja lähteä tekemään lisätyn todellisuuden sovellusta. On olemassa erilaisia ohjelmistokehityksen työkaluja (SDK), joita voi käyttää apuna tehtäessä lisätyn todellisuuden sovelluksia. Tässä opinnäytetyössä tullaan kuitenkin käyttämään Unityn tarjoamaa AR Foundation kehystä.

Pelin aihe sai idean tunnetusta lisätyn todellisuuden mobiilipelistä Pokemon GO ja opinnäytetyössä toteutettu peli on idealtaan samankaltainen. Tavoitteena oli luoda toimiva pelin prototyyppi, jota toimeksiantaja voi jatkokehittää tarpeidensa mukaan. Suunnitelmana oli mahdollisesti tehdä opaspeli Kaakkois-Suomen ammattikorkeakoulun Mikkelin kampukselle. Tässä opinnäytetyössä kehitettävä peli ei sisällä juurikaan grafiikoita eikä sisältöä, ainoastaan pelin toiminnallisuuden. Sijainti pysyy kuitenkin samana, eli peli sijoittuu Kaakkois-Suomen ammattikorkeakoulun Mikkelin kampukselle.

## **5.2 Sijaintitietojen hyödyntäminen mobiilipelissä**

Peli hyödyntää sijaintitietoja tarkastaessaan käyttäjän sijaintia todellisessa maailmassa. Pelissä olevat kohteet sijaitsevat oikeassa maailmassa ja pelaajan on käytävä niiden lähellä. Sijaintitietoja halutaan päivittää usein ja melko tarkasti, että peliä on mahdollista pelata.

Sijaintitietojen saamisessa on hyödynnetty Unityn tarjoamaa valmista koodia. Laitteelta halutaan saada sijaintitiedot päivittymään melko tarkasti, joten sijaintitietojen aloitus metodin parametreihin on laitettu luvut 0.5 ja 3 (kuva 17). En-

simmäinen luku 0.5 tarkoittaa sitä, että sijainnin tarkkuus on 0,5 metriä. Jälkimmäinen luku 3 kertoo, kuinka monta metriä pitää liikkua, jotta sijainti päivittyy.

```

1 reference
IEnumerator GPSLoc()
{
    // Check if the user has location service enabled.
    if (!Input.location.isEnabledByUser)
        yield break;

    // Starts the location service.
    Input.location.Start(0.5f, 3f);

    // Waits until the location service initializes
    int maxWait = 20;
    while (Input.location.status == LocationServiceStatus.Initializing && maxWait > 0)
    {
        yield return new WaitForSeconds(1);
        maxWait--;
    }

    // If the service didn't initialize in 20 seconds this cancels location service use.
    if (maxWait < 1)
    {
        statusText.text = "Timed out";
        yield break;
    }

    // If the connection failed this cancels location service use.
    if (Input.location.status == LocationServiceStatus.Failed)
    {
        statusText.text = "Unable to determine device location";
        yield break;
    }
    else
    {
        statusText.text = "Running";
        InvokeRepeating("LocationInfo", 0.5f, 1f); //Repeats LocationInfo function
    }
}

```

Kuva 17. Unityn tarjoama koodi muutettuna pelin tarpeiden mukaan

Testaamista varten on myös muutettu aikaisemmin koodissa olleet print-komennot. Aikaisemmin print-komentoina olleet tiedot lisätään nyt tekstikomponenttiin, jotta mobiililaitteella on mahdollista nähdä sovelluksen toimiminen. Koodista on vielä otettu pois sijaintitietojen tulostus ja lisätty siihen erillisen metodin ajaminen, kun sijaintitiedot on saatu.

### 5.3 Toimintojen rakentaminen

Toimintoja lähdetään rakentamaan luvussa 2 tehdyn lisätyn todellisuuden mobiilisovelluksen pohjan päälle. Kun sovellukseen on saatu sijaintitiedot toimimaan, voidaan muita toimintoja ryhtyä tekemään. Sijaintitietojen saannin jälkeen halutaan suorittaa metodi, joka hyödyntää laitteelta saatua sijaintia.

### 5.3.1 Kerättävien peliobjektien luominen

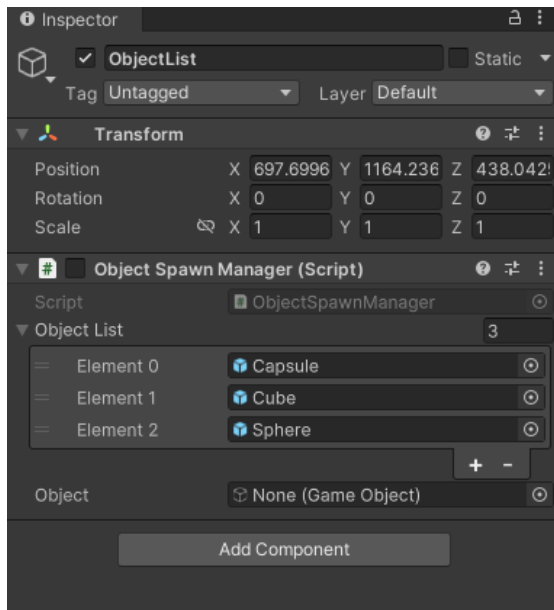
Opinnäytetyössä halutaan, että objektit tulevat näkyviin satunnaisesti listalta. Aluksi on luotava halutuista objekteista prefabit Assets-kansioon. Prefabeista on tehtävä lista, jolta aina yksi sattumanvaraisesti valittu peliobjekti tulee kerättäväksi näytölle.

Objektien pelimaailmaan lisäämistä varten on tehtävä skripti, jossa luodaan lista prefab-objekteista ja toiminnallisuus niiden luomiseksi peliin (kuva 18). Metodi `GetRandomObject` sisältää listalta satunnaisen objektin valinnan sekä sen luomisen peliin. Metodi `OnEnable` taas ajaa `GetRandomObject`-metodin joka kerta, kun skripti on laitettu päälle.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class ObjectSpawnManager : MonoBehaviour
6 {
7     public List<GameObject> ObjectList = new List<GameObject>();
8     public GameObject Object;
9
10
11     private void OnEnable()
12     {
13         GetRandomObject();
14     }
15
16     void GetRandomObject()
17     {
18         int index = Random.Range(0, ObjectList.Count);
19         Object = ObjectList[index];
20
21         Instantiate(Object, Camera.main.transform.position + new Vector3(0, 0, 1f), Quaternion.identity);
22     }
23 }
```

Kuva 18. Peliobjektin luomisen ja lisäämisen toteuttava skripti

Skeneen on luotava peliobjekti, johon lisätään komponentiksi ylempänä tehty objekteja luova skripti. Komponentin sisältämään listaan lisätään haluttu määrä prefabeja, joista voidaan valita satunnaisesti luotava objekti. Komponentin on oltava oletuksena pois päältä, koska sen tilaa hallinnoidaan toisesta skriptistä. (Kuva 19.)



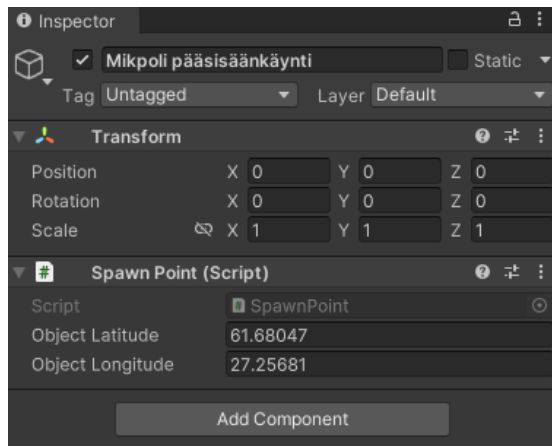
Kuva 19. Inspector-välilehti peliobjektista, joka sisältää skriptin objektien luonnille

Opinnäytetyössä käytettävät prefabit ovat Unityn primitiivejä 3D-malleja, joihin on lisätty eri väriset materiaalit. Lisätyn todellisuuden sovelluksissa objektien koko perustuu transform-komponentin scale-arvoon, jossa yksi Unityn yksikkö vastaa yhtä metriä oikeassa maailmassa. Tästä syystä mallien kokoa on muutettu pienemmiksi, jotta ne näkyvät mobiililaitteilla paremmin.

### 5.3.2 Pelin kohteiden luominen

Pelissä olevat kohteet, joihin objektit tulevat näkyviin, ovat ennalta määrättyjä. Kohteet ovat peliobjekteja, joihin on liitetty komponenttina skripti. Skripti sisältää muuttujat, joihin lisätään kohteen pituus- ja leveysaste. Kohteiden sijainnin lisäys tehdään manuaalisesti jokaiseen kohteeseen erikseen. Kuvasta 20 näkee, kuinka kohteen pituus- ja leveysasteet merkitään desimaalein skriptikomponenttiin.





Kuva 20. Mikpoli pääsisäämäkäynti -peliohjelman Inspector-välilehden näkymä

Kun kohteet on luotu ja asetettu niille koordinaatit, voidaan niitä vertailla laitteen sijainnin kanssa. Vertailu luodaan samaan skriptiin, missä laitteen sijaintitiedot on otettu. Vertailua varten on luotava leveys- ja pituusasteita varten float-tyyppin muuttujat, joihin tulee laitteen sijainnin ja kohteen sijainnin välinen etäisyys. Skriptiin on luotava myös lista, johon tulee kaikki luodut kohteet.

LocationInfo-metodi sisältää tiedon siitä, missä sijainnissa pelaaja on. InvokeRepeating-komennon avulla pystytään suorittamaan LocationInfo-metodia toistuvasti. Metodi kutsutaan ensimmäisen kerran 0,5 sekunnin kuluttua ja sen jälkeen sekunnin välein. CompareLocations-metodia kutsutaan LocationInfo-metodin sisällä. Kutsuttavan metodin parametreihin laitetaan laitteen leveys- ja pituusasteet, jotta niitä voidaan käyttää. (Kuva 21.)

```

66         else
67         {
68             statusText.text = "Running";
69             InvokeRepeating("LocationInfo", 0.5f, 1f); //Repeats LocationInfo function
70         }
71     }
72
73     0 references
74     public void LocationInfo()
75     {
76         if (Input.location.status == LocationServiceStatus.Running)
77         {
78             statusText.text = "Running";
79             Latitude = Input.location.lastData.latitude;
80             Longitude = Input.location.lastData.longitude;
81
82             CompareLocations(Latitude, Longitude);
83             locationText.text = $"Current Latitude: {Latitude}\nCurrent Longitude: {Longitude}";
84         }
85         else
86         {
87             statusText.text = "Stopped";
88         }

```

Kuva 21. LocationInfo-metodi

Sijaintien vertailu tapahtuu metodissa CompareLocations. Ensimmäisenä otetaan laitteen- ja kohteen sijainnin leveys- ja pituusasteiden erotukset. Erotuksesta otetaan vielä luvun itseisarvo, jotta ei tule negatiivisia lukuja. Laitteen etäisyyttä kohteesta voidaan nyt verrata haluttuun etäisyyteen if-lauseessa. Koska etäisyys on koordinaattien desimaalilukuna, on myös haluttu etäisyys merkattava desimaalein. Kun pelaaja on tarpeeksi lähellä kohdetta, suoritetaan if-lauseen sisällä oleva koodi. (Kuva 22.)

```

89 private void CompareLocations(float lat, float longi)
90 {
91     foreach (var location in LocationList)
92     {
93         LatitudeDiff = Mathf.Abs(location.ObjectLatitude - lat);
94         LongitudeDiff = Mathf.Abs(location.ObjectLongitude - longi);
95
96         if (LatitudeDiff < 0.00005f && LongitudeDiff < 0.00005f)
97         {
98             infoText.text = "Sijainti: " + location.name;
99             currentLocation = location.name;
100
101             if(!spawnButtonPressed)
102             {
103                 StartScanning.gameObject.SetActive(true);
104                 spawnButtonPressed = true;
105             }
106             break;
107         }
108
109         else
110         {
111             infoText.text = "Sijainti: Ei tiedossa";
112             spawnButtonPressed = false;
113
114             if (StartScanning.gameObject.activeInHierarchy)
115             {
116                 StartScanning.gameObject.SetActive(false);
117             }
118         }
119     }
120 }

```

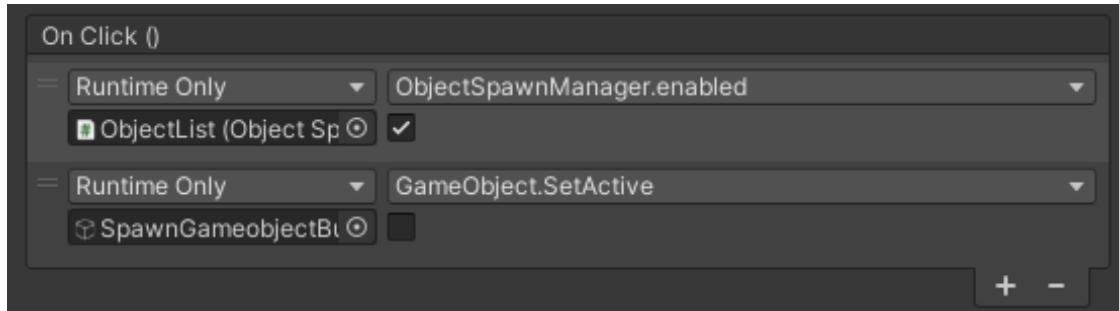
Kuva 22. CompareLocations-metodi

Pelaajan ollessa kohteessa aktivoidaan nappi, jolla pelaaja voi aloittaa peliohjelman luomisen. Myös sen hetkinen sijainti poistetaan listalta, jotta pelaaja ei voi samasta sijainnista kerätä kaikkia objekteja.

### 5.3.3 Pelin käytettävyyden kehittäminen

Kun pelaaja saapuu johonkin kohteeseen, sovelluksen käyttöliittymään ilmestyy nappi, josta kohteen kerättävän peliohjelman luominen tapahtuu. Luomisen jälkeen pelaaja voi kerätä objektin painamalla sitä näytöllä. Pelaajaa auttaamaan on luotu myös käyttöliittymästä löytyvä karttakuva, johon on merkattu määritettyjen kohteiden sijainnit.

Napille voidaan lisätä tapahtumia, jotka suoritetaan automaattisesti nappia painettaessa. Tässä napille on lisätty kaksi metodia, joista ensimmäisessä peliohjelman luova skripti aktivoidaan. Toisessa metodissa nappi poistetaan käytöstä, jotta se ei jää näkyviin. (Kuva 23.)



Kuva 23. Objektin lisäämisen tarvittavat napin metodit

Objektin keräämisen voi suorittaa, kun on painanut nappia, joka aktivoi peliohjelman luomisen. Peliobjektia voi joutua hetken etsimään oikeasta maailmasta, koska se ei liiku laitteen mukana luomisen jälkeen. Jos pelaajan laite heiluu objektin luonnin aikana, voi objekti jäädä piiloon kameranäkymän ulkopuolelle. Objekti luodaan laitteen kameran sijainnista yhden metrin verran eteenpäin. (Kuva 24.)

Kun pelaaja painaa objektista, se katoaa ja näytössä olevaa laskuria kasvatetaan yhdellä. Objektin painaminen deaktivoi objektin luonnissa käytettävän skriptin sekä kutsuu metodia, joka poistaa kohteiden listasta tämänhetkisen aktiivisen kohteen.



Kuva 24. Rajattu kuva pelinäköymästä, kun objekti on luotu

Pelissä on karttanäkymä, jotta kohteet on mahdollista löytää. Karttanäkymän saa päälle painamalla näytön reunassa olevaa Kartta-nappia. Pelissä nappi avaa kartan Mikkelin kampuksesta ja näyttää sinisillä pisteillä, missä kohteet ovat. Kartan voi sulkea näytön yläkulmassa olevasta ruksista. Kartta ei liiku eikä näytä pelaajan tämänhetkistä sijaintia. (Kuva 25.)



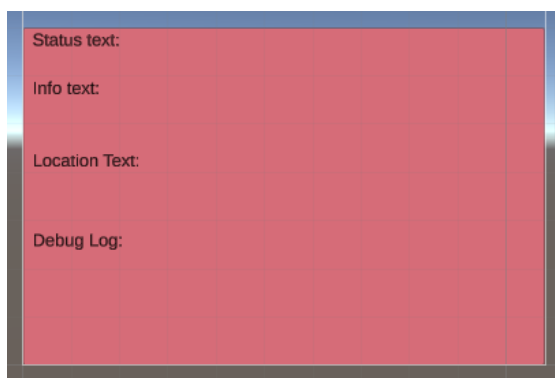
Kuva 25. Rajattu kuva kartasta

Opinnäytetyön peliä kehitettäessä haastavinta oli kokonaisuuden jäsentely helposti ratkaistaviksi paloiksi. Kehityksen aikana tulleiden ongelmien ratkaiseminen saattoi paikka paikoin olla myös haasteellista ja yksi ongelma jäi vielä kokonaan ratkaisemattakin.

#### 5.4 Pelin testaus

Unityllä mobiililaitteelle kehitettyä lisätyn todellisuuden sovellusta on hieman hankala testata. Mikäli haluaa testata kaikkien toiminnallisuuksien toimintaa, on sovellus ensin rakennettava. AR Foundation tarjoaa debug menun, joka helpottaa sovelluksen testaamista, mutta sekin vaatii sovelluksen rakentamisen.

Pelin testaamisen helpottamista varten työssä on käytetty tekstikenttiä, joihin voi syöttää tarvittavat tiedot (kuva 26). Status text kertoo, onko sijaintitietojen etsiminen käynnissä. Jotta nähtäisiin, onko käyttäjä oikeassa paikassa, tulee Info text -kohtaan tekstinä näkyviin ennalta määrätyn alueen nimi tai "Sijainti: Ei tiedossa". Sijainnin leveys- ja pituusasteet näkyvät Location Text -kohdassa. Debug Log -kohtaan tulee näkyviin, testattavan ominaisuuden testausloki.



Kuva 26. Testausta varten tehdyt tekstikentät

Pelin testaaminen tapahtui kahdella eri merkkisellä puhelimella, jotta nähtäisiin, onko niiden välillä eroja. Peliin sijoitetut kohteet ovat ennalta määrättyjä, joka helpottaa pelin testaamista. Pelissä on testattava, toimivatko halutut toiminnot ja näkyykö laitteella tarvittavat peliobjektit.

Suoritin pelin kehittämisen aikana yksikkötestausta jokaiselle erilliselle ominaisuudelle. Tarkka testaus on tärkeää vikojen ja ratkaisujen löytämisen kannalta. Ensimmäisenä testasin Unityn sijaintipalvelun toimivuutta ja miten tarkaksi sen päivittäminen on laitettava. Sisätiloissa testaamista varten sijaintitietojen päivittäminen oli laitettava tarkemmaksi, kuin ulkona oltaessa.

Kävin Kaakkois-Suomen ammattikorkeakoulun Mikkelin kampuksella testaamassa, kuinka peli toimii käytännössä useamman kohteen kanssa. Testatesani D-rakennuksen ja kirjaston välissä sijaitsevaa kohdetta huomasin, että peli ei toimi oikein. Objektin luomista varten tuleva nappi näkyi ja sitä painamalla objekti tuli näkyviin, mutta nappi ei deaktivoitunut. Selvitellessäni, mistä ongelma johtui, päädyin lopputulokseen, että listan viimeinen kohde ei toimi oikein.

Selvitin ongelmaa vaihtamalla listan kohteiden paikkoja ja huomasin, että aina listan viimeinen kohde oli viallinen. Tähän ongelmaan en kuitenkaan keksinyt mitään ratkaisua ja lopulta lisäsin listan viimeiseksi jäseneksi ylimääräisen kohteen, joka ei sijaitse pelialueella. Tämän avulla pelissä käytettävät kohteet toimivat oikein eivätkä siten häiritse sen pelaamista.

## **6 YHTEENVETO**

Opinnäytetyön suunnittelu ja toteutus onnistuivat mielestäni melko hyvin. Peliin olisi kyllä voinut tehdä hieman enemmän pelillisiä ominaisuuksia. Myös pelissä ilmenneet ongelmat olisi ollut hyvä saada korjattua. Opinnäytetyössä toteutettu lisätyn todellisuuden pelin prototyyppi onnistui kuitenkin siinä mielessä hyvin, että sen päälle on helppo luoda uusia ominaisuuksia ja kehittää vanhoja.

Koska opinnäytetyössä toteutettu peli on prototyyppi, on paljon lisättäviä ominaisuuksia, jotka sopisivat siihen. Pelistä puuttuu käytännössä täysin kunnollinen käyttöliittymä ja reaaliaikainen karttanäkymä, missä pelaaja näkisi oman sijaintinsa. Pelissä olevat kohteiden sisältämät objektit ovat Unityn primitiivejä 3D-malleja, jotka olisi hyvä muuttaa eri pelitoteutuksissa peleihin sopiviksi.

Pelin prototyyppiin luodut kohteet on tehty manuaalisesti ja niiden luontiprosessia voisi parantaa. Kohteet voisivat olla esimerkiksi satunnaisesti luotuja halutulle alueelle. Mikäli objektit luodaan satunnaisesti, on otettava huomioon alueen tiet, maastonmuodot ja rakennukset, jotta pelaajan on mahdollista liikkua kohteille. Kohteiden luonnissa on hyvä huomioida myös se, että ei luo niitä liian lähelle toisiaan, jotta pelialueesta ei tulisi liian ahdas.

Opinnäytetyössä luodussa pelissä kartalta ei näe pelaajan sijaintia, mutta kohteet on merkattu. Karttaa olisi hyvä kehittää siten, että siinä näkyisi myös pelaajan sijainti. Tämän ominaisuuden luomista varten olisi mahdollista hyödyntää jotain valmista karttatyökalupakettia, joista mainittiin luvussa 4.2.

Tämänhetkiseen käyttöliittymään olisi hyvä lisätä yhtenäisiä grafiikoita. Käyttöliittymää tehtäessä on huomioitava muun muassa, että käyttää yhtenäisiä värejä, typografia on selkää ja ei käytä liikaa eri fontteja. Käyttöliittymän tulisi olla myös skaalautuva eri kokoisille näytöille. Tätä ei ole tehty opinnäytetyössä tehdyssä pelissä.

Peliä kehitettäessä oli ideana luoda nuoli, joka näyttää missä kerättävä peliohjekti on. Nuolta ei peliin koskaan tullut, mutta se parantaisi käyttökokemusta huomattavasti. Nuoli tulisi esiin siinä vaiheessa, kun peliohjekti ei ole ruudulla näkyvissä ja siten helpottaisi pelaajaa löytämään kerättävä peliohjekti.

Lisätyn todellisuuden ominaisuuksien hyödyntäminen oli todella helppoa AR Foundationin avulla. Unityn tarjoamat dokumentaatiot AR Foundationin hyödyntämiseen olivat hyvät ja selkeät. AR Foundation tarjoaa paljon muitakin kuin opinnäytetyössä käytetyt ominaisuudet.

Lisätty todellisuus tarjoaa paljon muitakin käyttökohteita kuin viihdettä ja pelejä. Teknologiaa voi hyödyntää muun muassa opetuksessa sekä matkailu- ja kulttuurialalla. Tämänkin opinnäytetyöhön luodun mobiilipelin prototyyppin voi helposti jatkokehittää siten, että se soveltuu esimerkiksi opetuskäyttöön. Sovelluksesta voisi esimerkiksi kehittää Mikkelin kampusta ja sen toimintaa ja tiloja esittelevän opassovelluksen.

## LÄHTEET

About AR Foundation. 2022. Unity Technologies. WWW-dokumentti. Saatavissa: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html> [viitattu 19.10.2022].

ArcGIS Maps SDK for Unity. 2022. ArcGIS Developers. WWW-dokumentti. Saatavissa: <https://developers.arcgis.com/unity/> [viitattu 04.11.2022].

AR Foundation Debug menu. 2022. Unity Technologies. WWW-dokumentti. Saatavissa: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/ar-debug-menu.html> [viitattu 27.10.2022].

AR Plane manager. 2022. Unity Technologies. WWW-dokumentti. Saatavissa: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/plane-manager.html> [viitattu 24.10.2022].

AR tracked image manager. 2022. Unity Technologies. WWW-dokumentti. Saatavissa: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/tracked-image-manager.html> [viitattu 24.10.2022].

Glover, J. & Linowes, J. 2019. Complete Virtual Reality and Augmented Reality Development with Unity: Leverage the Power of Unity and Become a Pro at Creating Mixed Reality Applications. Birmingham: Pact Publishing. E-kirja. Saatavissa: <https://kaakkuri.finna.fi/> [viitattu 01.11.2022].

IL2CPP Overview. 2022. Unity Technologies. WWW-dokumentti. Saatavissa: <https://docs.unity3d.com/Manual/IL2CPP.html> [viitattu 21.10.2022].

LocationService.Start. 2022. Unity Technologies. WWW-dokumentti. Saatavissa: <https://docs.unity3d.com/ScriptReference/LocationService.Start.html> [viitattu 21.10.2022].

Maps SDK for Unity s.a. Mapbox. WWW-dokumentti. Saatavissa: <https://docs.mapbox.com/unity/maps/guides/> [04.11.2022].