

LOCATION-BASED MOBILE GAMES

Creating a location-based game with Unity game engine

Tiivistelmä

Tekijä(t) Korhola, Samuli	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 28	Valmistumisaika Kevät 2020
Työn nimi Sijaintipohjaisuus mobiilipeleissä Sijaintipohjaisen pelin kehitys Unity pelimoottorissa		
Tutkinto Tieto- ja viestintätekniikan insinööri.		
Tiivistelmä <p>Tämän opinnäytetyön aiheena oli sijaintipohjaiset mobiilipelit. Sijaintipohjaiset mobiilipelit ovat pelien tapa yhdistää oikea maailma virtuaalisen maailman kanssa ja täten ne luovat yhdessä aivan uuden pelikokemuksen.</p> <p>Tämä tutkimus syventyi teknologiaan ja työkaluihin, joilla kehitetään sijaintipohjaisia pelejä. Näihin sisältyy esimerkiksi GPS ja Bluetooth. Samalla työssä myös tutustuttiin yleisesti sijaintipohjaisten pelien ominaisuuksiin.</p> <p>Melkein kaikki tekniset ratkaisut, jotka oli esitetty opinnäytetyössä, olivat Moomin Move peliprojektin teknisiä ratkaisuja. Opinnäytetyön tuloksena tuli lisää mahdollisuuksia kehittää Moomin Move pelin sijaintipohjaisia ominaisuuksia, kuten tuomalla kamerapohjaisia sijaintitekniikoita.</p>		
Asiasanat Unity, sijaintipohjainen, mobiilipelit, GPS, Bluetooth		

Abstract

Author(s) Korhola, Samuli	Type of publication Bachelor's thesis	Published Spring 2020
	Number of pages 28	
Title of publication Location-based mobile games Creating a location-based game with the Unity game engine		
Name of Degree Bachelor of Information and Communications Technology		
Abstract <p>The subject of this thesis is location-based mobile games. Location-based mobile games are a way for mobile games to combine reality with virtual worlds and thus re-define the gaming experience.</p> <p>This thesis presents the technologies and tools that are used to create location-based games. This includes technologies like GPS and Bluetooth. The thesis also inspects the general features found in location-based games.</p> <p>All technical solutions presented focus on the design choices made on the Moomin Move game project. The thesis produced more solutions for developing the location-based services like the camera technology in the project.</p>		
Keywords Unity, location-based, mobile games, GPS, Bluetooth		

CONTENTS

ABBREVIATIONS

1	PREFACE.....	1
2	LOCATION-BASED GAMES	2
2.1	Introduction.....	2
2.2	Exploring location-based games.....	2
2.2.1	Pokémon GO.....	2
2.2.2	Ingress.....	3
2.2.3	Jurassic World Alive	4
2.2.4	The Walking Dead: Our World	5
2.2.5	Minecraft Earth	6
2.3	Possibilities with location-based games.....	7
2.4	Problems with location-based games.....	7
3	LOCATION-BASED TECHNOLOGIES	9
3.1	Global Positioning System (GPS)	9
3.2	Mapbox.....	10
3.3	GameSparks.....	11
3.4	Bluetooth technology	12
3.5	Augmented Reality and camera technology.....	14
3.5.1	Vuforia.....	14
3.5.2	ARCore.....	14
3.5.3	ARKit	15
4	DEVELOPING MOOMIN MOVE	17
4.1	Location-based services in Moomin Move	17
4.1.1	GPS locations.....	19
4.1.2	Bluetooth locations	20
4.2	Improving location-based systems.....	21
4.2.1	Augmented Reality locations	21
4.2.2	Server systems.....	23
4.2.3	Client systems	23
5	CONCLUSION.....	25
	REFERENCES	26

ABBREVIATIONS

Android	Mobile operating system
API	Application programming interface
AR	Augmented Reality
BaaS	Backend-as-a-service
CCU	Concurrent users. It is the total number of people who use a service at a given period
GPS	Global Positioning System
IOS	Mobile operating system by Apple
MAU	Monthly active users
SDK	Software Development Kit

1 PREFACE

Location-based games became popular when Pokémon Go hit the world. After Pokémon's success many other games have entered the location-based genre. New augmented reality technologies like ARKit and ARCore have been released, which offer new possibilities for these types of games. Location-based games bring new challenges and technologies, which need to be understood and learned when making games for this genre.

The aim of this thesis is to study the technologies behind location-based mobile games and to show what is needed to take into consideration when developing location-based games. The thesis focuses on the design decisions made with Moomin Move game to show a real-life example of the topic.

The practical case for this thesis is created with Tribered Oy to improve location-based features found in the Moomin Move game project. Tribered Oy has around 10 personnel and had a turnover of around 200 000 in 2017. Tribered Oy is based in Helsinki and provides services globally.

The objective of the thesis is to improve location-based services in the Moomin Move game and to investigate improvements to the location-based systems found in the game. The scope of the thesis is mostly limited to the technology and development choices in Moomin Move. Technology is also limited to the Unity game engine, GameSparks platform and Mapbox data as they are used in the Moomin Move game. There are lots of other game engines and server platforms available that can achieve similar results.

2 LOCATION-BASED GAMES

2.1 Introduction

Location-based games merge the game space with real life. This allows the game space to be unique depending on the real-world location. It also allows people to create communities simultaneously both in physical and digital spaces. (de Souza e Silva 2009, 410.)

Mobile location-based gaming is achieved with new technologies available on mobile devices. These include GPS positioning, cellular network and augmented reality technologies. (Buczowski 2012.)

2.2 Exploring location-based games

In summer of 2016 Pokémon GO was released, and it was the first real hit in the location-based games genre. Other games that have been released after Pokémon GO are Jurassic World Alive, The Walking Dead: Our World, Harry Potter Wizards Unite, Ghostbusters World and many others. Also, more games are coming like Minecraft Earth and Dragon Quest. (Carlton 2019.)

Location-based gaming started when phone manufacturers started providing GPS location technology in their smartphones. Parallel Kingdom was one of the first games in the genre released in 2008. Niantic's Ingress in 2012 was the next big step up in the genre. In 2016, with the brand of Pokémon, location-based games really had their first success. (Telfer 2018.)

2.2.1 Pokémon GO

Pokémon GO has been made by Niantic. Pokémon GO gameplay involves a lot of walking around and searching for Pokémon creatures. There are different types of Pokémon that can be found in different locations. After catching Pokémon, they are then trained to duel in Gyms and to defeat Raid bosses. (Carlton 2019.)

The basic gameplay of Pokémon GO is that you walk around in the real world. As you get close to a Pokémon, your device will vibrate and you will see a Pokémon pop up in your vicinity. After finding a Pokémon, you can try to catch it with a Poké Ball. This is shown in the Picture 1 below. The Pokémon can escape the ball, so you cannot always catch all Pokémon immediately. Once the Pokémon gets caught in the ball, it will be added to your inventory, where it can be upgraded and evolved. (Pokemon.com 2018.)



Picture 1. Catching Pokémon in Pokémon GO. (Pokemon.com 2018.)

2.2.2 Ingress

Ingress was the first location-based game set in augmented reality from Niantic. Ingress was released seven years ago for Android and six years ago for IOS. In Ingress players learn about exotic matter that is coming to the world. Exotic matter is coming through portals, which the players can try to control. (Lagace 2020.)

Players in Ingress will need to choose between two factions: the Enlightened and the Resistance. The decision of faction is important, if you want to play with your friends, as you need to be in the same faction to play together. This has made an impact on the story of Ingress as the faction decision is now more about choosing to play with friends, instead of picking the faction based on a philosophic decision. The story of Ingress was more cryptic in the beginning, but as social features have superseded the importance of the story, the differences between factions have been simplified. (Lagace 2020.)

Five years after original Ingress release, Niantic released Ingress Prime, which was a big free update to the game. Ingress Prime improved mechanics from the original game. It also added new abilities and gave the game a new look with a redesigned user interface. Even though Ingress has not been as big a success as Pokémon GO, Ingress is still supported and updated seven years after its release. (Lagace 2020.)

2.2.3 Jurassic World Alive

Jurassic World Alive is a location-based dinosaur game that has been developed by Ludia. The game was released in 2018. The core gameplay consists of three parts. First you find dinosaurs on the map and collect DNA from the dinosaurs. The DNA then unlocks and upgrades dinosaurs. The dinosaurs are then used to battle other players. (Telfer 2018.)

Jurassic World Alive is designed to be less dependent on location-based gameplay than Pokémon GO. As Pokémon GO requires you to walk and traverse to have a proper game session, Jurassic World Alive allows you to battle and open chests everywhere you want. Dinosaurs also appear more in the same location, allowing you to play more without moving. This lowers the barrier for playing as it is very easy to have a short game session. When you are required to walk with the game, you will be more invested in the playing, as you will need to invest more time in playing. So, while it is easier to boot up the game and collect some dinosaur DNA, you will not be as invested into starting a long game session. (Telfer 2018.)

Jurassic World Alive was originally being created on an internal game engine, but Ludia decided to switch to the Unity game engine midway through the game development cycle. This decision came from needing to find a third-party solution for a game engine as their own game engine was not ready. Their own game engine would have required a lot of development before they could have started to work on the game. After they tested Unity with a small pilot project, they decided to stop work on their own engine and to migrate Jurassic World Alive to Unity. The transition to Unity went smoothly with Unity Enterprise Support. The new workflow allowed them to meet their deadline on their project. Unity also offered more autonomy for artists and streamlined the production pipeline. Internal game engine development and support is also costly, so to have someone else take care of it resulted in cost savings. The company's Unity games in general became more stable with less crashes than games that were developed on their internal game engine. (Unity.com 2020.)

Unity offered a more content-based development approach to the game development, as the tools allow artists to tweak the game visuals themselves. Ludia's own game engine required programmers to do most of the visual integration, which did not encourage as much creativity. Unity also offered much faster build times and easier building for different platforms. Creating an application build would previously take two to three times longer time. (Unity.com 2020.)

2.2.4 The Walking Dead: Our World

The Walking Dead: Our World is a Finnish location-based augmented reality game set in a world infested by zombies (Forde 2019). The Walking Dead: Our World landscape is filled with supplies, walkers and shelters. The game's map is powered by Google Maps. (Nedd 2018.)

The game's AR feature is that you shoot zombies that are mixed in with the real world. You get different weapons and a companion to help you slay the zombies. The AR is optional, and players can choose to play in a fully digital environment or in a mixed reality environment. (Nedd 2018.)

The Walking Dead: Our World also offers challenge boards. They are filled with tasks which are completed together with a group. The tasks provide variety for the gameplay as they might require you to switch characters or weapons for a task. Groups bring a social aspect to the game as you are working together for a common goal. Small rewards are given for individual tasks and larger rewards are given when the whole board is completed. These rewards are shared between the group and bring motivation to work together. Challenge boards can be seen in the Picture 2 below. (Nedd 2018.)



Picture 2. The Walking Dead: Our World challenge board. (Nedd 2018.)

2.2.5 Minecraft Earth

Minecraft Earth is new a location-based game that connects the Minecraft game with the real-world. Minecraft Earth is available for both Android and IOS. (Minecraft Earth FAQ 2020.)

In the game you collect blocks, mobs and items. After collecting the collectables you can build on tabletop Buildplates. Buildplates are small Augmented Reality areas where you can build, and place blocks like in the Minecraft game. The AR allows you to step into your buildplate as it is inserted into the real world. Buildplates are shown in the Picture 3 below. Friends can be invited to visit your Buildplates. The owner of the Buildplate needs to invite others for them to get access to the Buildplate. Also, when the owner stops playing, will all other player be disconnected. This way no one can mess in someone else's Buildplate when the owner is away. (Minecraft Earth FAQ 2020.)



Picture 3. Buildplates in Minecraft Earth. (Minecraft.net 2020.)

Minecraft Earth also offers survival Adventures. Adventures are found on the game map. The contents of Adventures are randomized and thus provide surprises and hidden treasures. Adventures are small AR Minecraft world pieces like the Buildplates. (Minecraft Earth FAQ 2020.)

2.3 Possibilities with location-based games

Games are a way to tell stories and to create worlds. Immersive world building is now a standard in games, and this creates a demand for large-scale realistic environments. Location data allows developers to develop large-scale worlds as they are created from the real-world data. Location-driven features not only add to the experience, they redefine it. (Chang 2019.)

Location-driven games also allow new social experiences. Not only can a player group up virtually, but they can also use their unique physical location to expand their multiplayer experience. Next Games the Walking Dead: Our World showed that player groups can send flares to allow other group members virtually join them at their physical location. That way they can complete missions together around the flare. (Chang 2019.)

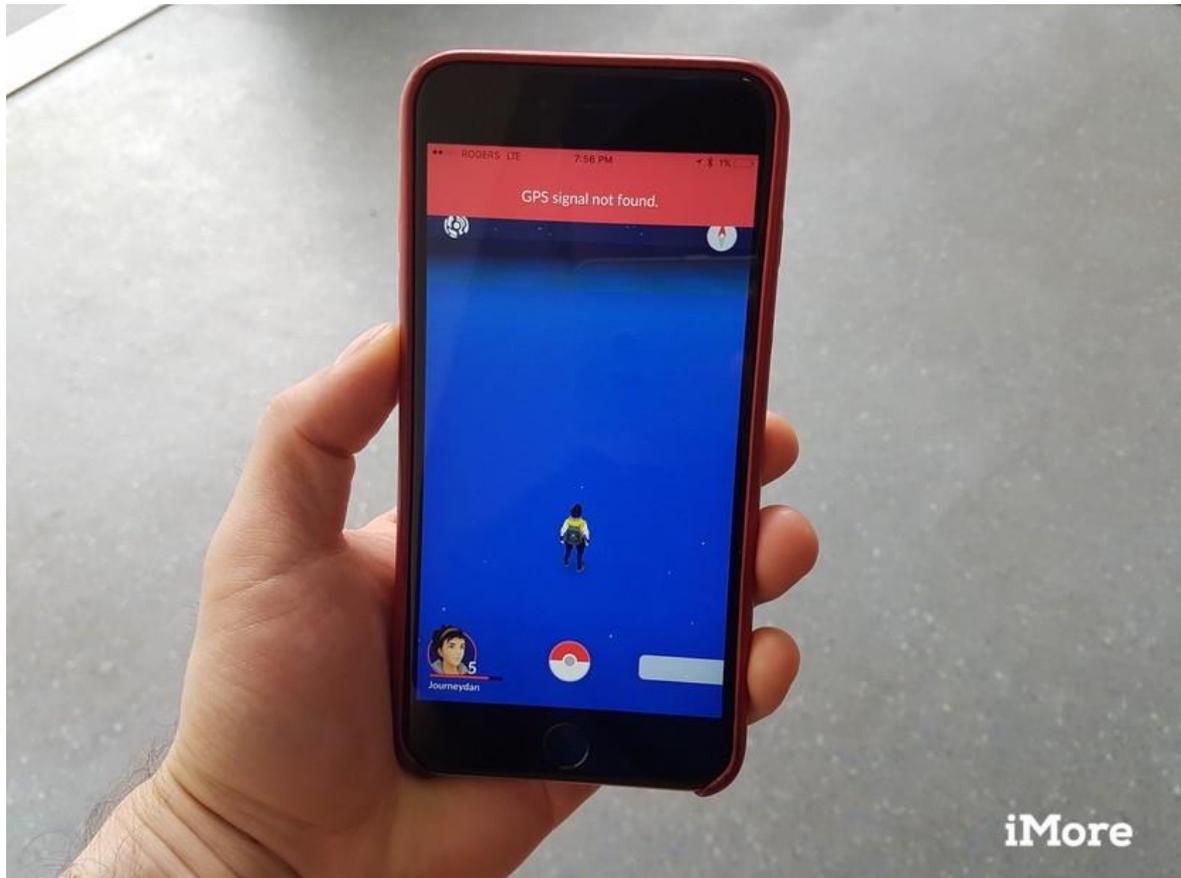
Location-based games provide the possibility to do business partnerships with brands and companies. This is achieved with sponsored and licensed locations. Brands like Starbucks and McDonalds have sponsored locations in Pokémon GO. In Pokémon GO these locations are raid locations in the game, where people gather around. This kind of marketing is good for locations as this creates traffic to their stores and locations. (Robles 2019.)

2.4 Problems with location-based games

The first technical problem with location-based games is the amount and quality of data that needs to be handled in the systems. Players can open the game in any country on any day. The data needs to be provided quickly and needs to update as relevant. The data content will also be very unpredictable as the location can be anywhere in the world. The location might not provide a good gaming experience and can make moving from one location to another difficult. The location might not even have the necessary location information available, thus making the game unplayable with an empty world as seen in the Picture 4 below. Analyzing game design issues in location-based games is difficult as the play area is the entire world. It is hard to reproduce the problem when you are on the other side of the world and your data, network and device can be hugely different. This often results in going through issues on a case by case analysis as it is hard to analyze so many different cases. In a normal digital game, you can just go to the same game section and try to reproduce the problem more easily. (Brimicombe & Li 2009, 139-140; Jacob & Coelho 2011.)

Relying on constant GPS and Internet connection limits the player base as playing with Wi-Fi does not allow the player to move around and thus requires playing with a cellular

network. Constant location detection and pulling data from servers will seriously affect the device's battery charge. (Jacob & Coelho 2011.)



Picture 4. Missing GPS signal and an empty world. (Daniel Bader 2016.)

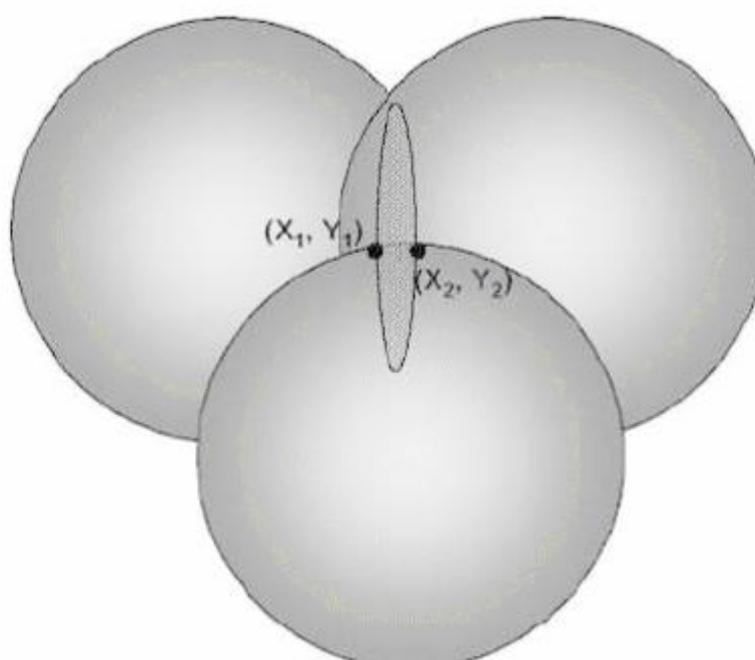
The player's fitness and physical condition also influences the game difficulty making the game harder for less fit and lazier people. This needs to be taken into consideration in the game's design as it might make the game unplayable to some. Players also need breaks and pauses while moving in the real world, so the game should give the player a break when it is needed. (Jacob & Coelho 2011.)

While augmented reality offers unique experiences with the real world, it might also have unintended side effects. If the game offers shooting mechanics like in *The Walking Dead: Our World*, where you shoot zombies in AR, real people can also walk into the frame of the camera. This can create disturbing situations for the players as they can shoot real people in the game, even if they are not their targets. (Nedd 2018.)

3 LOCATION-BASED TECHNOLOGIES

3.1 Global Positioning System (GPS)

A majority of today's mobile handsets are equipped with GPS chips that allow the devices to calculate their position in the real world with satellites, within ten meters or less. This requires at least three satellites for the location to be accurate, as shown in the Picture 5 below. The GPS signal can be obscured by houses and terrain, so a clear line-of-sight to the sky is necessary for accurate results. The location provided by the device is generally shown in coordinates (e.g. latitude and longitude). (Brimicombe & Li 2009, 182-186; Masters & Harvey 2014, 9.)



Picture 5. Locating the user with satellite measurements (Brimicombe & Li 2009, 175.)

The Unity game engine offers GPS location service inside the engine. The LocationService class in Unity offers simple functions and properties that can be used to get the devices location. LocationService only offers two functions: Start and Stop. These functions are used to start the service and to end it, if it is no longer needed. The accuracy of the requested location can be defined with the `desiredAccuracyInMeters` property. The default value of the property is 10 meters. If only a rough estimate of the location is needed, the value can be set to 500 or higher. These high values usually do not require the GPS chip to be enabled and thus can save battery power. (Unity LocationService 2020.)

3.2 Mapbox

Mapbox is a platform used for creating applications with real-world geographical data and spatial analysis. The tools can be used to create maps for websites or to build geoprocessing applications. (Mapbox 2020.)

Mapbox maps are created as layers. The layers depend on the tileset used for the map. Each tileset has its own set of data that will be used to create the map. Tilesets that are available in Mapbox are:

- Mapbox Streets
- Mapbox Terrain
- Mapbox Satellite
- Mapbox Traffic

Mapbox Street is the basic tileset variation that includes streets, water areas, buildings and various land data. Mapbox Terrain includes landcover and elevation data. Mapbox Satellite includes imagery from various sources like NASA, Maxar, Nearmap, National Land Survey of Finland and many others. Mapbox Traffic offers traffic information on top of the Mapbox Street tileset. (Mapbox 2020.)

One of the core features of Mapbox is allowing to create completely custom map designs. This means that the developer can freely customize every layer of the map. Layers can be configured with custom alignments, offsets, colors, weights and opacities. (Mapbox 2020.)

Mapbox offers Mapbox Maps SDK for the Unity game engine. The SDK allows the game engine to dynamically load map data at runtime. This optimizes the application, as the application only needs to load the latest data around the user. The SDK also allows to create 3D maps with the various tilesets. The map data is converted to meshes and game objects. This allows for more customization with custom textures and the map is easier to incorporate to the rest of the application. (Mapbox 2020.)

There are also other geographic data providers available like Google Maps. One key difference between Mapbox API ja Google Maps API is the price, as Google is more expensive. Mapbox is free for up to 50 000 monthly active users while in Google dynamic maps the service will start costing immediately. (Birchard 2019.)

3.3 GameSparks

The GameSparks platform is a backend-as-a-service designed to make game backend production easier. The premise is to have a game backend that is easy to build and maintain. (GameSparks study 2020.) GameSparks offers many SDKs for different game engines and platforms like Unreal Engine, Unity, Android and IOS. (GameSparks documentation 2020.)

The GameSparks platform components run on runtime clusters. The runtime clusters are built on Amazon Elastic Compute Cloud. This allows the clusters to be scaled up easily. That way the backend can have players from a handful to millions, depending on demand. AWS Lambda is used to automate provisioning and scaling of the environment. The strength in BaaS is that the platform manages these systems automatically and the game developer can focus on developing the game features instead of creating network structures themselves. (GameSparks study 2020.)

GameSparks offers three pricing models: Development, Standard and Enterprise. All models offer 100 CCU in preview environment. The development model is free to use, but it is only for development as it does not have live environment available. Standard offers live environment with up to 10 000 CCU for the price of 299 dollars per month. A high MAU amount will also increase the price if the 37375 monthly user limit is exceeded. Enterprise model prices and limits are negotiable and not publicly available. (GameSparks pricing 2020.)

Development on GameSparks is done through the GameSparks development portal. In the portal, games are managed on a game by game basis. GameSparks has live and preview environments. This means you can develop and test the backend in the preview environment. Once it is ready to deploy you can create a snapshot of the current preview environment and push it to live. This is crucial for games to work as a service as they need to be developed when the service is live for customers. Working on live environment directly is not ideal as it will often break the end user's experience. When you work on cloud code and features, the backend can become unplayable with the changes and if a user is simultaneously using the service, the experience will be bad for the user and the backend data might go into an unexpected state. Therefore, you should always have a separate environment for development and live users. (GameSparks documentation 2020.)

3.4 Bluetooth technology

Bluetooth technology can be used to locate user's mobile devices for indoor positioning. Bluetooth devices transmit signals containing device identity that can be picked up by the user's mobile device. This signal can be processed and used to detect where the user is located. (Brimicombe & Li 2009, 200-201.)

Bluetooth beacons are small hardware transmitters that transmit the Bluetooth signal around them. The idea of beacons is shown in the Picture 6 below. Smart devices then pick the signal up and detect what beacon they are close to. After finding out which beacons are in the vicinity, the beacon details are then used to get physical coordinates from inside the app or from an external database. The Bluetooth beacon itself has no location intelligence built into it. To determine the exact location of the user in an area, at least three beacon signals are required for trilateration. The received signal strength indicator, which represents the power of a received radio signal, is used to determine the distance of the device to each beacon. These distances are then calculated and correlated to the beacon positions in the real world and this then gives an estimate on the user's location in the space. (Proximi.io 2017.)



Picture 6. Bluetooth beacons used to detect user location. (Proximi.io 2017.)

iBeacon protocol beacons transmit UUID, major and minor fields to the smart device, which then identifies the beacon from these values. The UUID can be shared by all Bluetooth locations of the organization allowing the device to use a single identifier to recognize all the locations. Minor values can be configured to be the same in similar locations making it easier for an app to identify different types of locations as seen in the Table 1 below. (Apple.com 2014.)

Table 1. Example table of iBeacon ID configuration (Apple.com 2014.)

Store Location		San Francisco	Paris	London
UUID		D9B9EC1F-3925-43D0-80A9-1E39D4CEA95C		
Major		1	2	3
Minor	Clothing	10	10	10
	Housewares	20	20	20
	Automotive	30	30	30

Investigating what hardware to use is important when deciding to utilize Bluetooth beacons. There are many different beacon manufactures, which each offer different technical specifications in their beacons. Some offer beacons with irregular signal or weak battery life, which can lead to problems. Irregular signal especially can make location detection hard as the strength of the signal is used to determine the distance to the beacon. (Proximi.io 2017.)

Placing the beacons in the real world requires planning in the placement of the beacons. A good practice is to get the floor map of the environment where the beacons will be placed. This way the dimensions of the rooms can be easily measured, and the number of beacons can be calculated. For accurate positioning, at least one beacon should be put in every corner of the room. In larger rooms you will need to put additional beacons between the corners as beacons have a limited range and might not cover the whole room. The beacons should be placed on walls at two-meter height for optimal operation. Beacons should also be visible, so the signal will not be blocked by any obstacles. The signal strength can be adjusted in beacons. It should be tuned on site for optimal operation and to accommodate for any obstacles blocking the signal. When creating beacon solutions to multiple floor solutions you need to tune the signal strength accordingly, so the beacons from other floors will not interfere with the current floor. Beacons should be placed next to staircases and lifts, when the app needs to know when the floor needs to be changed. (Proximi.io 2017.)

3.5 Augmented Reality and camera technology

Augmented reality technology can be used to enhance the user's perception of the real world by overlaying virtual objects to the real world. The experience surpasses the reality by creating unreachable scenes in the real world with digital objects. (Park 2018.)

3.5.1 Vuforia

The Vuforia AR Software Development kit by Qualcomm is supported in the Unity game engine. It utilizes computer vision technology to analyze the camera viewfinder input of the real world. After tracking the world, it can place virtual objects on top of the background of the camera. (Park 2018.)

Vuforia offers support for image targets. Image targets are first configured to the database as a set of images that the application is trying to find from the camera input. When the application finds the image, its position and angle will be tracked. This information can then be used to create virtual content on top of the real-world feed as the application has a reference point in the real world. (Park 2018.)

The Vuforia engine will no longer be natively integrated in the Unity game engine and will require to be separately downloaded from the Unity asset store. Before Unity version 2019.3, Vuforia was included as an optional built-in package for the Unity game engine. (Fuad and Mahendran 2020.)

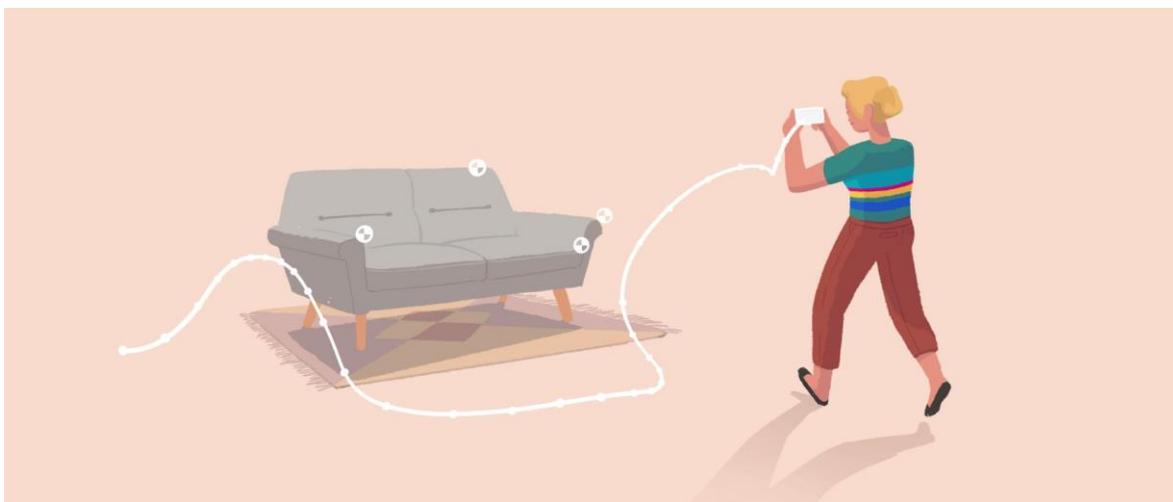
Vuforia is free for development, but when you publish your project you need to select a subscription plan. Vuforia offers three pricing tiers: Pro, Basic + Cloud, and Basic. Basic is the cheapest option costing 42\$ per month. Basic + Cloud adds large cloud image database support but increases the price to 99\$ per month. The pro license has no visible price tag but is required if you make content with Vuforia for companies with over 10\$ million turnover per year. If you make a project for a large client, you will need to pay for the pro license even if you are a small company. (Vuforia pricing 2020.)

3.5.2 ARCore

ARCore is platform for creating augmented reality experiences by Google. Some of the APIs can be used on both Android and IOS. ARCore is mainly designed for Android phones. (ARCore 2020.)

ARCore's two fundamental tasks are to track the positions of the mobile device as it moves around, and to understand the real world around the device. The way the device

understands the world is by identifying interesting points through the phone's camera. The point movements are then tracked, and this relative position is then used to calculate the movement of the user's device in the world. This is shown in the Picture 7 below. Internal sensors are used to enhance the information of the position and orientation. (ARCore 2020.)

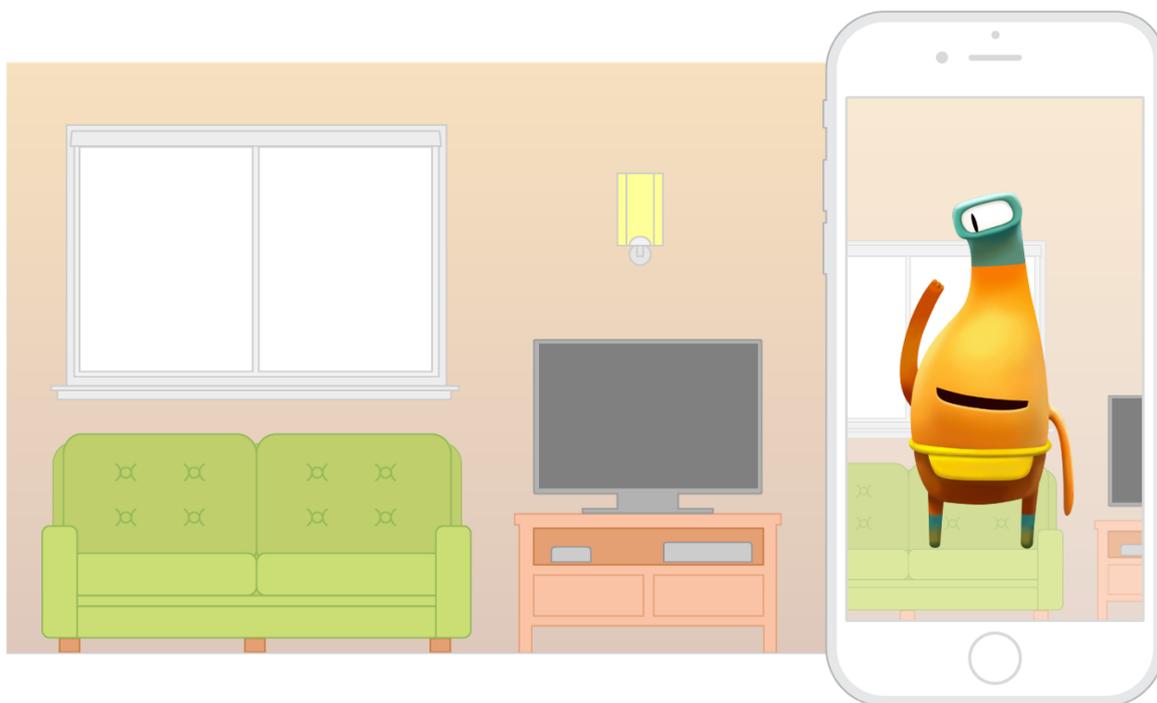


Picture 7. ARCore motion tracking and point detection. (ARCore. 2020.)

Once ARCore has analyzed the real world, the tracking data can then be used to place digital objects in the real-world locations. This data allows the users to move around and view the digital objects. Even if the location of the object gets out of the camera view, ARCore will be able to detect the same location again once the location gets back in the view. (ARCore 2020.)

3.5.3 ARKit

ARKit is an iOS device camera and motion feature API for producing AR experiences. ARKit can create and track correspondence between the real-world around the device and a virtual space where visual content can be created. The live camera image is combined with virtual content to create augmented reality. See Picture 8 below for a concept of virtual content in the real world. (ARKit 2020.)



Picture 8. Virtual content added to the real world. (ARKit documentation 2020.)

The tracking of the real world is a difficult process in AR experiences. The world that the ARKit maps is just an estimate of the world that it can perceive through the camera input. The physical environment will not always be consistent as lighting conditions can change. Blank walls also provide problems as they do not have details that the ARKit can try to track. That is why, when you are designing experiences for AR, you need to be aware of these limitations and indicate them to the user. Too fast motion easily breaks the experience, which is why users should be instructed to move slowly. Plane detection will require time to produce clear results, so it is good to give enough time for the detection to operate. (ARKit documentation 2020.)

4 DEVELOPING MOOMIN MOVE

Moomin Move is a location-based augmented reality game, which blends the lines between the real world and the world of Moomin. Key components of the game are a friendly attitude, social collaboration and discovering the Moomin Valley and its unique characters. One of the key elements of the game is to help Moomins and to build your relationships with them.

The core gameplay loop consists of four main parts:

- Collecting & Exploring
- Inventory management
- Quests & Challenges
- Upgrading and level progression

Exploring is done through traversing in the real world. Your character follows your movement in the real world and moves the character in the game. This way you can explore new places and find items and characters. Collecting happens in a number of simple mini games. Only pet collecting requires chance and effort, as other collectibles only require energy. Pets require luck for catching them as they have a percentage chance that can be improved by deepening your relationships with the Moomin characters.

Once items are collected, they are added to your inventory. Inventory requires constant management of items as it has limited slots and space. Players need to make decisions on what items to keep, when their inventory overflows.

Moomin characters give you quests once you become their friend. The quests require you to gather items and as a reward you get more special items. Groups in the game also have varied tasks that you need to complete for unique rewards.

Players gain levels from doing quests for Moomin characters. You also level up your friendship levels with the characters. Level-ups unlock more daily quests and improve players' chances for catching pets.

4.1 Location-based services in Moomin Move

In Moomin Move the general location of the user in the game world is determined by GPS. This is achieved with the Unity game engine's native location detection functionality. Unity allows you to choose between fine location and low accuracy location. Low accuracy is good when the application does not need to know the exact location of the device, but a rough idea where the device is located. The fine location can detect movement of a few

meters and with the low accuracy mode it will be 500 meters or more. Moomin Move requires the game character to move as the user moves in the real world, so this requires the fine location.

GPS is used in most locations of the game, but there are situations where it is not feasible to use, and other technologies are required to continue the experience. Indoor locations and moving locations like ships will be impossible for GPS to create a reliable experience. That is why specific indoor locations also use Bluetooth to confirm that the user has entered the location, when the GPS signal might not be available. GPS is also more unreliable for detecting small position changes and Bluetooth will more reliably confirm the exact location of the user, if the user is in its range. Moving locations like ships must fully utilize alternative methods for detection as the location itself will be constantly moving and would be a nightmare to track anyway with GPS. That is why Moomin Move uses only Bluetooth once the user boards a ship.

After determining the location of the user, the game will create the game world around the user. The stylized combination of the real world and Moomin Valley is created from the user's GPS location and Mapbox real-world data, as shown in Picture 9 below. The game world is then combined with custom and generated game-related content to fill the game world with gameplay content. The generated game content is shared with other players to create social experiences with other players. Normally the game world would require the map to be synchronized between players, but as the Mapbox data is derived from the real world, it will function in the same way for everyone even without synchronization. Using the location as the seed for generation of the non-gameplay content will allow the map location to be constantly the same. Not using the seed would make the trees and details of the environment change position every time the user starts the game. Also, by seeding all players will have the same random seed, and the world will be the same for everyone.



Picture 9. Moomin Move game world map and gameplay content.

4.1.1 GPS locations

Most of the locations found in Moomin Move are powered by GPS technology. The game automatically generates content around the player's GPS coordinates with generation algorithms. This allows the players to play anywhere in the world and to have game content available whenever they have GPS connection. The generated content is managed by the game's backend as it would not be shared with other players, if they were only generated locally on the user's device.

Normal gameplay content for players found on the map is collectibles, bushes and portals seen in Picture 9 above. Collectibles are items that the player can store, use and sell, and they are needed to complete quests for Moomin characters. Items are also required for events and group quests. Another type of objects found on the map is bushes. They are a way for the player to collect energy for actions. The most complex objects on the map are portals. They are the way for the player to enter Moomin Valley and to meet Moomin characters. There they can get quests from the characters and create relationships with the characters.

Some of the GPS locations are custom-made for partner locations. They have custom unique stories that provide special gameplay for players and customers in partner locations. This kind of special locations that do not have Bluetooth technology are faster to create as they do not need physical hardware in the location.

4.1.2 Bluetooth locations

Moomin Move uses Bluetooth beacons to detect whether the player is inside a special location in the game. Bluetooth beacons are configured with a unique ID and then placed inside the licensee location. Then they are configured in the backend and can be found in the game. The backend is constructed in a way that allows the game to add new locations on the fly without needing an update on the game client.

Special Bluetooth locations are created in partnership with other companies for mutual benefit. Partnership locations get visibility in the game and potential customers visiting their stores. Moomin Move gets advertisement, visibility in partner locations and special gameplay for players.

Silja Line cruises had the first Bluetooth locations for the game. This was a special case as GPS cannot detect your position inside a ship at all. That is why the game has a custom ship map that is navigated with Bluetooth technology. The map can be seen in Picture 10 below. Bluetooth beacons are placed around the ship to provide location detection inside the ship.



Picture 10. Moomin Move Silja Line custom map.

Other partner locations utilize both GPS and Bluetooth to detect that the player has entered the given location. GPS is used to show the location on the game map and Bluetooth is used to detect whether the player has entered the location.

4.2 Improving location-based systems

The systems found in Moomin Move serve their purpose quite well as they have been in development for three years. Still, due to the small team size and limited resources, there is much room for improvements.

Player location is detected in Unity with its native systems. All location-based game items are generated, stored and distributed from the GameSparks platform. Geographical real-world data is pulled from Mapbox databases.

4.2.1 Augmented Reality locations

Moomin Move has great potential for making augmented reality-based locations in the game. These locations would utilize the camera of the user's phone to track the places around them and to enhance them with virtual elements.

Moomin as a brand has many licensed real-world locations that could utilize augmented reality features. Locations like Moominworld, Moomin Museum and Moomin Valley Park in Japan have many key locations in them that could be detected with AR. These locations can then be enhanced with living virtual characters, who can give information and quests to you.

Moomin Move has Vuforia implemented and it is utilized only for tracking camera rotation and for creating a camera image background while collecting in AR mode. Vuforia offers support for image, cylinder and model targets. Image targets are simple images that the application can recognize. After being recognized, the targets can be used as trackable points for digital objects or for unlocking content in the game once the image is recognized. Cylinder targets are like image targets, but as the name suggests they are for images that are wrapped around objects in cylindrical shape. These can be used for instance in coffee cups that are popular in the Moomin brand. Model targets are for detecting real-world objects that are in a known shape. This could be useful for detecting real-world objects like in the Picture 11 below.



Picture 11. Moominhouse at Moomin Valley Park Japan. (Moomin 2019.)

Vuforia itself is an easy option for starter users of the Unity engine as it is well integrated in the game engine. Vuforia has had some questionable design choices that can lead to development problems, if the app needs to have non-AR content too. Vuforia forces the game scene to have an active Vuforia camera upon startup that will require some extra work if you do not want the AR experience to be active right away on startup. The Vuforia license can easily lead to problems when using it for customer-related projects. Even if the project is low budget and small, Vuforia will require a pro license, when creating an application for a customer that has more than 10\$ million revenue. The price of Vuforia in the pro license is high and will render most small projects unprofitable for bigger companies as the license will take most of the budget.

4.2.2 Server systems

One of the biggest tasks that has yet to be tackled is a proper implementation of the geo data to the backend, which generates the game content. Currently the real-world data is barely used when generating the game content and this could be greatly enhanced. This would give more gameplay variety to the game based on the location from where the game is played. This would also allow to differentiate the items more as they would require more specific locations. Now everything can be found everywhere at random.

GameSparks does not provide tools for directly calling the Mapbox data. This means that the data needs to be collected with custom call structure for getting the necessary information to GameSparks. After receiving the data, GameSparks can create more tailored content for the players. This would allow the servers to spawn different collectible and character types depending on the type of the location. For instance, near water the player could find more fish. This would also provide more gameplay choices for players as they could influence the items they find by relocating themselves to a more suitable location.

The Unity client already utilizes fully the data provided by Mapbox to create a Moomin-like version of the real world. This includes getting data of streets, buildings, landscapes etc. to build a geographically accurate world that the players can navigate by walking in the real world. Players can easily share their experience with other players as the world is created from the same data.

4.2.3 Client systems

Player location detection for Moomin Move is created with Unity location detection, which is simple to use, but is lacking in features. The system provided by Unity does not allow to handle the system after starting as it has only Start and Stop functions. This is not optimal as GPS systems can have problems due to bad connection, old device or other factors. So, you only have a choice of restarting the system or hoping that the system solves the problems itself.

Accuracy data provided by the system can hugely overestimate the margin of error. This results in the system thinking that the data can be very inaccurate even when it is not the case. For instance, the system can say that the coordinates have a margin of error of 10 000 meters, when the coordinates given by the system are mostly within 300 meters of the correct location. That is why the native accuracy values need to be bypassed with a custom solution for better results.

The Player GPS coordinates provided by Unity are rounded to floats, which results in them being truncated and losing accuracy. Coordinates are large decimal values that need at least 5 decimals to show locations in an accuracy of 10 meters. Floats can only store 7 significant digits and thus are prone to losing data when having latitudes of larger than 100 degrees. This will result in the player location jumping around as the location can change tens of meters when there are not enough decimals presented for the location.

Bluetooth locations in Moomin Move do not use triangulation, which could allow the character in the game to move freely in indoor locations. Now the location of the character is set to the nearest beacon, which will result in the character jumping from one location to another. Triangulation would provide more persistent gameplay experience with the GPS experience, but it would require at least three times the number of beacons in every location. This would also require additional client systems that can calculate the distances of nearby beacons and to use these distances to get the real-world location of the device. The individual locations of the beacons would also need to be more precisely cataloged and managed for them to be functional reference points for location detection.

Camera-based indoor locations are an alternative to Bluetooth locations, but it will require many gameplay-related changes. The user needs to have the camera active and to look around the environment through the camera view for the location detection to work. This will be difficult to manage for players as they need to be in the right location in the first place for the detection to work. If they try to use the camera to detect a location somewhere else, they will be frustrated as nothing can be found. This is a major disadvantage compared to Bluetooth as Bluetooth can be active in the background and once it detects something it can then notify the user that something is nearby. A combination of these technologies could be utilized. Bluetooth would detect when an area is near and then the user would be advised to activate the camera view to look for certain areas. The more technologies users need to use, the harder and more unpleasant it is for them. This will also require more tutorials for the users on how to use the technologies and might result in loss of user retention as it might be too much for them to handle. Using many technologies together will also consume more power and reduce the battery life.

5 CONCLUSION

Location-based services can provide exiting features and social experiences for players. They can enhance our world with virtual elements and the real world can be used to enhance the virtual elements.

When making location-based games, you need take into consideration many technologies and game design elements that are not typical in other mobile games. GPS, Bluetooth and cellular data play a huge role on the user's device and experience. These technologies require more technical design decisions than is required in a normal mobile game.

Mapbox and other services like Google can be used to get location data that is required when creating the world map into game. Unity provides off-the-shelf solutions to some client location-based systems, but they are not always adequate on their own. GameSparks offers a solution for creating a game's backend, allowing the developers to focus more on gameplay instead of server structures.

REFERENCES

- Apple.com. 2014. Getting started with iBeacon [referenced 29.2.2020]. Available: <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>
- ARCore. 2020. ARCore overview [referenced 16.2.2020]. Available: <https://developers.google.com/ar/discover/>
- ARKit. 2020. ARKit overview [referenced 16.2.2020]. Available: <https://developer.apple.com/augmented-reality/>
- ARKit documentation. 2020. Understanding World Tracking [referenced 16.2.2020]. Available: https://developer.apple.com/documentation/arkit/understanding_world_tracking
- Bader, D. 2016. No GPS signal picture [referenced 27.2.2020]. Available: <https://www.imore.com/how-fix-gps-signal-not-found-error-pokemon-go>
- Birchard, T. 2019. Geographic Data Visualization with Mapbox [referenced 14.3.2020]. Available: <https://hackersandslackers.com/map-data-visualization-with-mapbox/>
- Brimicombe, A. & Li, C. 2009. Location-Based Services and Geo-Information Engineering [referenced 16.2.2019]. Available: <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/detail.action?docID=437465>
- Buczowski, A. 2012. Location-based Gaming [referenced 16.2.2019]. Available: <http://geoawesomeness.com/location-based-gaming/>
- Carlton, B. 2019. Location-Based AR is Changing How The World Looks At Gaming [referenced 23.2.2020]. Available: <https://vrscout.com/news/location-based-ar-gaming/>
- Chang, S. 2019. Google maps platform. Take mobile gaming to the next level with location [referenced 18.3.2019]. Available: <https://cloud.google.com/blog/products/maps-platform/take-mobile-gaming-to-the-next-level-with-location>
- de Souza e Silva, A. 2009. Hybrid Reality and Location-Based Gaming: Redefining Mobility and Game Spaces in Urban Environments [referenced 16.2.2019]. Available: https://www.researchgate.net/publication/240280878_Hybrid_Reality_and_Location-Based_Gaming_Redefining_Mobility_and_Game_Spaces_in_Urban_Environments
- Fuad, M. & Mahendran, AC. 2020. Unity XR platform updates [referenced 16.2.2020]. Available: <https://blogs.unity3d.com/2020/01/24/unity-xr-platform-updates/>
- Forde, M. 2019. One Year on: Next Games on the evolution of location-based title The Walking Dead: Our World [referenced 2.3.2020]. Available:

<https://www.pocketgamer.biz/interview/71276/one-year-on-next-games-on-the-evolution-of-location-based-title-the-walking-dead-our-world/>

GameSparks pricing. 2020. GameSparks Pricing [referenced 28.3.2020]. Available: <https://www.gamesparks.com/pricing/>

GameSparks documentation. 2020. GameSparks documentation [referenced 28.3.2020]. Available: <https://docs.gamesparks.com/documentation/key-concepts/development-process.html>

GameSparks study. 2020. GameSparks Case Study [referenced 24.3.2020]. Available: <https://aws.amazon.com/solutions/case-studies/gamesparks/>

Jacob, J. & Coelho, A. Portugal. 2011. Issues in the Development of Location-Based Games [referenced 17.2.2019]. Available: <https://www.hindawi.com/journals/ijcgt/2011/495437/>

Lagace, M. 2020. Ingress: Everything you need to know in 2020 [referenced 15.3.2020]. Available: <https://www.androidcentral.com/ingress>

Mapbox. 2020. How Mapbox works [referenced 14.3.2020]. Available: <https://docs.mapbox.com/help/how-mapbox-works/>

Masters, Harvey P. New York: Nova Science Publishers. 2014. Location-based Services: Developments and Privacy Issues [referenced 16.2.2019]. Available: <http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=776110&site=ehost-live>

Minecraft Earth FAQ. 2020. Minecraft Earth FAQs [referenced 15.3.2020]. Available: <https://help.minecraft.net/hc/en-us/articles/360033744412-Minecraft-Earth-FAQs>

Minecraft.net. 2020. Minecraft Earth [referenced 27.2.2020]. Available: <https://www.minecraft.net/fi-fi/about-earth>

Moomin.com. 2019. Moominvalley Park Japan [referenced 23.2.2020]. Available: <https://www.moomin.com/en/blog/the-first-moomin-theme-park-outside-finland-is-simply-magical-and-officially-open-see-the-full-photo-story-of-moominvalley-park-japan/#80c1af73>

Need, A. 2018. New 'Walking Dead' mobile game has a killer social element, but fairly pointless AR [referenced 2.3.2020] Available: <https://mashable.com/article/the-walking-dead-our-world-ar-game-review/?europa=true>

Our World. 2020. The Walking Dead Our World [referenced 29.2.2020]. Available: <https://www.thewalkingdeadorworld.com/>

Park, D. 2018. Application Development with Augmented Reality Technique using Unity 3D and Vuforia [referenced 16.2.2020]. Available: https://www.ripublication.com/ijaer18/ijaerv13n21_33.pdf

Pokemon.com. 2018. Pokémon GO [referenced 27.2.2020]. Available: <https://www.pokemon.com/fi/app/pokemon-go/>

Proximi.io. 2017. How to do accurate indoor positioning with Bluetooth beacons [referenced 16.2.2019]? Available: <https://proximi.io/accurate-indoor-positioning-bluetooth-beacons/>

Robles, P. 2019. Are sponsored locations in games good business for SMBs [referenced 21.3.2020]? Available: <https://econsultancy.com/are-sponsored-locations-in-games-good-business-for-smbs/>

Telfer, A. 2018. Can Jurassic World Alive stand up to Pokémon GO [referenced 17.3.2020]? Available: <https://mobilefreetoplay.com/can-jurassic-world-alive-stand-up-to-pokemon-go/>

Unity LocationService. 2020. Unity LocationService Documentation [referenced 29.3.2020]. Available: <https://docs.unity3d.com/ScriptReference/LocationService.Start.html>

Unity.com. 2020. Ludia's Jurassic World Alive: A Unity case study [referenced 24.2.2020]. Available: <https://unity.com/case-study/jurassic-world-alive>

Vuforia Pricing. 2020. Developer Vuforia Pricing [referenced 24.3.2020]. Available: <https://developer.vuforia.com/pricing>