

Outdoor Augmented Reality

Ethan Zammit[4802L]
ethan.zammit.19@um.edu.mt
University of Malta

June 19, 2021

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as “the unacknowledged use, as one’s own work, of work of another person, whether or not such work has been published” (Regulations Governing Conduct at Examinations, 1997, Regulation 1 (viii), University of Malta).

I / We*, the undersigned, declare that the [assignment / Assigned Practical Task report / Final Year Project report] submitted is my / our* work, except where acknowledged and referenced.

I / We* understand that the penalties for making a false declaration may include, but are not limited to, loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.


Work submitted without this signed declaration will not be corrected, and will be given zero marks.

* Delete as appropriate.

(N.B. If the assignment is meant to be submitted anonymously, please sign this form and submit it to the Departmental Officer separately from the assignment).

Ethan Zammit

Student Name



Signature

Student Name

Signature

Student Name

Signature

Student Name

Signature

ARI2201

Course Code

Individual Assigned Practical Task - Outdoor AR

Title of work submitted

14/06/2021

Date

Contents

1 Introduction

1.1 Brief of Project

This project aims to delve into the effects of the exploitation of Augmented Reality Techniques on tourism and heritage.

A system is to be developed using Google's ARCore, enabling complex computer vision-based functions to be easily embedded within an android app.

1.2 Aims & Objectives

There are three main objectives to this project, which are to be explored and implemented to analyse the effects of their combination. Firstly, Augmented Reality is to be used and implemented in the setting of an outdoor environment, using several technologies provided by ARCore, we may explore different ways that this can be exploited for the best experience. This may even be combined with other sensory data such as the location for a better touch with reality. Secondly, an android application is to be developed, which will allow ease of use by most people, avoiding hurdles of installations and such. This application is to provide some method of direction to landmarks, and also incorporate the AR experience when appropriate. Thirdly, The application is to get information from a server, which will allow for a centralised and controlled method of managing landmarks, descriptions and their locations.

1.3 Functionality Developed

1.3.1 The Application

An android app was developed that the users can easily install and access. The app constantly uses the location service to consult the API server, providing the device's current location and retrieving a list of explorable landmarks. These landmarks are displayed on the application, as a list of potentially explorable landmarks, also giving a general bearing of where the user should head to reach the site.

Upon entering the geofence, which means the location of the user is within relatively small proximity of a landmark (defined server-side), the app enables a landmark to be selectable, which when selected, the device enters an AR mode. When in AR mode, the user can get a floating 3D informational window that contains details about the landmark selected.

1.3.2 The API Server

The server is to contain a list of landmarks, including their names, locations, a description and maybe even a set of images. The server should allow a device to consult it with location-based information and a list of landmarks (within some proximity) is returned to the device, where the device uses the information given and lists them as potential landmarks to explore.

The landmark information is to be stored on the server, so anything can be easily changed by changing the configurations of the server, and the mobile apps simply obtain newer information, without needing to rebuild or update the applications. As this technology may be adapted to other uses, such as games, it would be really useful to be able to have an idea of where players are.

2 Background Research

2.1 ARCore and Unity

Google ARCore is a framework for the creation of AR experiences on android and IOS devices, and comes with great capabilities out of the box, and simplifies the workflow by removing the need of reinventing everything from scratch. A simplified framework enables lower-cost projects [?], lower-qualified developers and faster integration of AR projects, providing a gateway to the mainstream acceptance of AR.

Unity 3D technologies combined with Google ARCore enhance the usability of ARCore, as developers are enabled to keep using existing, familiar tools to develop an Augmented Reality Experience. As unity has been a longstanding game development engine, it also has a wide range of support, add-ons and a helpful community. Also enabling the development of different platforms such as desktop, web and mobile applications [?].

2.2 Augmented Reality and Tourism

Augmented Reality in the tourism domain is no new application works such as [?] show promising results of the applications in the tourism domain. Furthermore, [?] also presents an insight into the future of this domain application, and results show that the area is still blooming. By combining Augmented Reality with other features, we can change and improve the experience, tailoring it to specific applications.

2.3 Augmented Reality Enhancement

Augmented Reality encapsulates a wide are of technologies which ultimately aims at enhancing, changing and manipulating some aspect of reality as we know it. AR can also be combined with other novel technologies, [?] delves into how the implementation and variation of these technologies can change the resultant experience, including technologies such as location, routing, interactive views and more.

When these two technologies are combined, the experience is taken to another level, as the Augmented Reality experience can shift based on the device's real conditions and positioning. A level of immersion is reached, as users need to move and visit heritage sites in order to experience the Augmented Reality effects, and in return, they gain context and heritage information about the

landmarks visited.

3 Implementation Details

3.1 The Server

An API server was written in python, as due to the small scale of the application, this was ideal to meet the requirements whilst keeping the implementation simple enough. The server provides several endpoints which may be pinged, but only two particular endpoints are used.

3.1.1 Location Updates

The server keeps track of a list of active devices, (though a unique identifier provided on requests), and their last known location. The device regularly updates the sever with location information, and then requests nearby landmark data. The server loops through all landmarks, and calculates the distance between the device longitude and latitude positioning, and the landmark. If the distance is below some threshold, it is added to a list of potentially explorable landmarks, which is returned as a JSON response to the device. Each landmark entry also contains a geofence region, which when is larger than the distance, the landmarks are considered near the user, and the device can know that the AR mode can be enabled.

3.2 Landmark Menu

3.2.1 General User Interface

Unity provides a wide range of features that can be used to implemented UI in the application. Modern solutions make use of a canvas gameObject which promises several features such as user interaction, screen scaling/responsiveness, and list managing features such as scrollable lists.

A scrollable list was used to show a list of the landmarks returned by the server. The title, a short description, the raw distance from the location and the bearing to the location is shown for each entry, so the user may have some basic information on how to reach a target. Whenever some landmark is very close, the entry becomes interactable, and the user can press it to enter AR mode near the landmark.

3.2.2 Location Service

On the device, the location is updated every second, with an intended accuracy of 0.1 metres (usually not met, but we try to be as accurate as possible). The integrated unity function is used, and a listener is used to check for updates, which update the server, and the landmarks list accordingly.

3.2.3 Geometric Distance & Geofencing

3.2.4 Geometric Bearing

3.3 AR Mode Technologies

In this mode, the camera is shown to the user, and a 3D translucent floating window is spawned in 3D space. The user may move around the panel, and observe the panel stays locked in 3D space. The panel shows some deeper description about the near landmark. A carousel allows the user to see some images of the place (as provided through the API).

3.3.1 Scaling & Interaction

When combined with unity, ARCore gives the developer access to the AR camera and the pose driver, which are responsible for synchronizing movements carried out in real life, with the movement inside the game engine. A system is used where 1 unit in the unity engine corresponds to one meter of scale when viewed in AR mode. Thus If we need to create a 30cm box, we can just set the size in the engine to 0.3, without needing to do any manual conversion, etc.

A 'World Canvas' gameObject was used so that we can use UI features in 3D space, allowing for a simpler implementation when handling image placement, text sizing, and any other form of user interaction with the panel. By appending the information panel to this type of canvas, we also inform ARCore that we want to synchronise the locations between the real world and the virtual unity world space.

3.3.2 Information Panel Instantiation

Using the aforementioned techniques, we first create a generic prefab for the Information Panel. This includes the colouring, text placement, title placement, and other things such as button placement and functionality etc. We also provide some relative sizing, which will be scaled accordingly when the prefab is used.

Upon Clicking on the desired entry from the landmarks menu, the information of the landmark is transferred to the AR Mode scene, this data includes things such as the title, long description and image URLs. We then simply populate the prefab with the information provided, and instantiate the panel in the world canvas, relative to the AR camera position. The Pose driver then ensures that the panel stays fixed in the in-game position, whilst varying the AR camera position to match real-world movements of the device.

3.4 Other AR Techniques

A couple of other Augmented Reality techniques were explored during the development of this app. These techniques were implemented and worked really

well as stand-alone, yet when combining the features, these standalone techniques were omitted, as they would require infrastructural changes to the way that landmarks are stored and communicated, which I believe is out of the scope for this project. They are still left available in the unity project in case these would be implemented in the future yet as is, there is no way of accessing them through the App.

3.4.1 Plane detection

Although raw plane detection was not used in the final version, under the hood google uses it to keep the floating information panel in place. Through goggle's AR Core it is made possible to detect vertical and horizontal planes, to which other game objects can be anchored to!

In an example, plane detection was used to find a stable surface, and when the user clicks on a plane, a 3D Game model is spawned in place and anchored to the plane. The user is also able to walk around in the room, whilst the objects stay anchored to the plane!

3.4.2 Image Recognition & Augmented Images

Image Recognition was also a really interesting feature to use. In the case of this project, a quick database manager was created in which a list of images could be inserted. These images also need to have specific features, to be distinguishable from other places (for example some image of plain grey gravel is not very easily distinguishable, but an image of the earth is). And actions would be taken according to the image detected!

In an example, an image of the earth was used as a key, and when this image is detected, a 3D spinning globe would be overlayed on it, where the user can go around the globe and see all sides! This may have been able to be implemented in the app, yet as the landmark menu and the ARMode switching works, It did not have much room to be used. (As in the near landmark menu, there is no access to the camera), and the user may only use AR Mode when near a landmark. However, this feature is also fully working, and may easily be implemented if a better use in the context of outdoor AR is identified.

4 Evaluation & Analysis

Due to the domain of this project and the technologies adopted, it was not as straightforward to test the raw performance of the system. As there is no performance metric or such that can be analysed and compares to other instances [?].

4.1 Testing Performed

4.1.1 Landmarks AR Information Panel

After finalizing the system (application and server), testing took place in the real locations of the landmarks. Through this on-site testing, an idea of how the application performs was obtained.

Since the floating information panel stays in place using plane detection and sensory data, care was given to push the limits of these technologies. Plane detection struggles when there are no close discernable attributes (the floor has a regular pattern for example), which the tests have also shown to affect the stability of the position of the panel.

4.1.2 Augmented Images

As augmented images were implemented some basic testing was also involved, which turned out to be quite successful. The image was recognized from different angles and light settings whilst tracking also was really responsive to even moving the image.

Further testing is shown in the video, as it's the easiest way to show off the experience.

4.2 Potential Testing

Three key points mentioned in [?] have great potential in giving a more formal understanding of the performance of such systems.

4.2.1 Independent Variables

The variation of independent variables can be used as a performance metric and help analyse the experience of the user, these variables include things that are not varied by the user during the test. Examples of such variables include the device size and the field of view of a device's camera.

4.2.2 Dependant Variables

These variables give a robust metric of how the users react to the application presented, such as the number of attempts taken to carry out an action. Yet this is very application-specific and does not allow for comparison between other systems, as tasks in an application usually are specific to it.

4.2.3 Questionnaires

Questionnaires were mentioned as another performance metric, which allows a subjective metric. In applications such as the tourism domain, this is of utmost importance, as it follows that from the user-centric nature, the ultimate goal is the users' experience.

5 Conclusion

In this project, I managed to meet the intended aims and objectives and during the process also obtained hands-on experience of using the latest technologies to apply the theories learnt into practice. Augmented Reality and computer vision as a domain also turned out to be very rewarding, as the effort done is immediately reflected into tangible progress.

5.1 Future Work

During development, it was made sure to keep the system as open as possible. Through the centralised API, unity engine and other technologies used, the system is meant to be dynamic and expandable.

With minimal effort, it can be easily be adapted to different uses such as an AR game, using google APIs for international standardized locations or even showing 3D models in AR mode.