VIRTUAL ASSISTANT USING AUGMENTED REALITY AND SPEECH RECOGNITION

A Project Report submitted in the partial fulfilment of the requirements for the

Award of degree

Of

Bachelor of Technology

In Computer Science and Engineering

By

Vishwas Tamta — **D/17/CS/205**

Bikash Khanal — **D/17/CS/206**

Under the Guidance of

Mr. K. Legoh

Assistant Professor

Department of Computer Science and Engineering,

North Eastern Regional Institute of Science and Technology

Deemed to be University (Under MHRD, Govt. of India)

Nirjuli-791109, Arunachal Pradesh, India

June 2020

CERTIFICATE OF APPROVAL

This is to certify that the project work entitled "Virtual Assistant Using Augmented Reality and Speech Recognition" is hereby approved a bona fide work of study as an engineering subject, carried out jointly by students:

Vishwas Tamta (D/17/CS/205)

Bikash Khanal (D/17/CS/205)

And presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is hereby understood by this approval that the undersigned do not endorse or approve any statement made, opinion expressed or conclusions drawn therein, but approves the report only for the purpose for which it has been submitted.

Mr. K. Legoh
(Project Guide)

Mr. Pradeep Kamboj
(Project Coordinator)

Dr. (Mr.) M. Marjit Singh

(Head of Department)

Computer Science and Engineering

NERIST

CANDIDATE'S DECLARATION

We hereby declare that the project titled Virtual Assistant Using Augmented Reality And Speech Recognition, a prerequisite towards partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering submitted to the Department of Computer Science and Engineering, North Eastern Regional Institute of Science and Technology (Deemed University), Nirjuli, Arunachal Pradesh – 791109 is an accurate record of our work carried under the guidance of Mr. K. Legoh Def, Assistant Professor, Computer Science and Engineering Department.

VishwasTamta (D/17/CS/205)

Bikash Khanal (D/17/CS/206)

ACKNOWLEDGEMENT

The satisfaction that accompanies upon the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts.

We express our extreme gratitude and thanks to our project guide Mr. K. Legoh, Assistant Professor as well as project coordinator Mr. Pradeep Kamboj, Assistant Professor, Department of Computer Science and Engineering, NERIST for taking deep interest during project work and valuable guidance throughout our project work. Without his counsel, we would not have come up to this stage of completion.

We also owe our sincerest gratitude to Dr. (Mr.) M. Marjit Singh, HoD, Department of Computer Science and Engineering, NERIST for his kind support and cooperation. We are also thankful to our distinguished Faculty members and staffs of Computer Science and Engineering Department who played a vital role in bringing us to the level of engineering standards.

Vishwas Tamta (D/17/CS/205)

Bikash Khanal (D/17/CS/206)

ABSTRACT

This project involves the development of an android application which acts as a simple assistant to the user like for getting the information of maps and user location along with the nearby places to the augmented reality environment. Secondly user can watch YouTube videos and can feel the fun of augmented reality. Next it can be used for indoor navigation as it is possible sometimes that we forgot the ways inside a shopping complex etc.

This project can help in many ways as it does give the features for navigation, browsing YouTube videos and indoor navigation as it works as an HUD (head up display) for the better experience of the user and making all these abilities easier for user to have and access.

By the help of augmented reality we can achieve the ability of using the virtual objects and features in the real environment as a mixed reality which allows us to do even greater things to do such as in the field of navigation and in many other areas.

Table of Contents

List of Figures

Abbreviations and Nomenclature

POI: Point Of Interest.

AR: Augmented Reality.

VR: Virtual Reality.

AI: Artificial Intelligence.

SDK: Software Development Kit.

AR-Core: Used for building Augmented Reality Experiences for Android devices.

AR-Kit: Used for building Augmented Reality Experiences for IOS devices.

AR-Foundation: Augmented Reality development environment for both Android and IOS applications.

Raycast: Process of casting a ray with the help of device camera to detect a plane surface.

GameObject: These are used as a component in the Unity for containing various components.

Mapbox API: Open source application programmable interface for accessing and editing map data.

Wit.ai API: It is an application programmable interface for creating an AI bot.

JASON: JavaScript Object Notation.

PHP: Hypertext Preprocessor.

URL: Uniform Resource Locator.

VideoPlayer: The component that holds all the configuration for a video in the unity.

HUD: Head Up Display.

CHAPTER 1. INTRODUCTION

This project is an augmented reality project which will work if you give any basic commands using the voice commands then it will process that command and show the output on the screen, while the user can still see the real time environment through the same screen.

This project uses a head up display where the user can see some important notifications without any look away from their usual viewpoints. HUD is any transparent display that presents data without requiring users to look away from their usual viewpoints. The origin of the name stems from a pilot being able to view information with the head positioned “up” and looking forward, instead of angled down looking at lower instruments. A HUD also has the advantage that the pilot’s eyes do not need to refocus to view the outside after looking at the optically nearer instruments.

If you want the indoor navigation or if you want to have a virtual display ready to open a YouTube video then just by one voice command it will be executed and will be displayed on the screen.

Augmented Reality is the technology by which Digital Images and 3D Models can be viewed along with the real environment and thus creating a composite view for the user. Unlike virtual reality where whole environment is virtual. In Augmented Reality there are virtual objects in the real environment and thus creating an augmented reality which is a mixture of both real and virtual environment.

Speech recognition is the ability to recognize the speech of someone by the help of some computation. AI algorithms are used in this system for better understanding or recognition of the input speech.

1.1. Objective

This project helps the user to use the already existing features in easier way then before like the navigation system maps exists but the way we use them is by reading it in the app and then make the decision but with the help of augmented reality we can change that old way of seeing things as a plain sheet.

This project also helps getting the most of the user experience by browsing YouTube videos and getting indoor navigation in AR.

As it is a virtual assistant so the main aim of this project is to give the user a simple easy to use virtual assistant by the help of voice recognition for getting commands from the user.

The main objectives of this project are:

1. To implement the mapbox maps for the navigation system to work and thus create the AR environment map for the user to access.
2. To implement the indoor navigation system in AR by the help of creating and storing the paths in real-time for user.
3. To implement the feature of browsing YouTube videos in the AR environment.
4. Implementing the Voice Recognition system for getting the commands from the user to be executed with the help of wit.ai platform.

1.2. Scope

Augmented Reality technology demand is increasing rapidly in the field of computer science and technology as it offers us as programmers to develop applications which are able to provide us the props of mixed reality in the real-time environment. Therefore augmented reality has a very good scope in the upcoming future in many environment like learning, teaching, drawing, navigation, and in many more fields.

CHAPTER 2. LITERATURE SURVEY

Well we got the idea of doing this project by the help of experiencing different kinds of virtual assistants currently available they all have an AI algorithm working within it for interacting with the users and when the user demands something within the capabilities of virtual assistant it will fulfil the demand. But then we thought what we can do to improve the facilities already provided by the virtual assistant so we came up an idea of a virtual assistant with augmented reality embedded on it so that the user can interact with the virtual assistant easily and can have a good output through the virtual assistant in a mixed reality.

VIRTUAL ASSISTANT

As name suggests it works as an assistant for the user which provides a number of things or works which an assistant can do, there are many other assistants in the web now a days like Google Assistant, Apple Siri, Amazon Alexa and many more coming to assist the user in some small tasks but among all of these this project is all about assisting the user in some tasks because all the tasks are pre made by different people around the world like navigation, indoor navigation, browsing videos etc. but this project helps the user to get most from these tasks by having an virtual assistant that can interact with the user not by the speech but by the augmented reality output given by it.

HEAD UP DISPLAY (HUD)

As the name suggest itself head up display is a kind of display where, while user is using the HUD as a display he/she does not have to look away from their usual view points and thus getting some notifications etc. on the HUD.

This technology helps us a lot in this project because we want the user to have a Virtual Assistant which can not only interact with the user but it can show the output directly to the users while not disturbing the usual viewpoints of the user.

MAPBOX API

Mapbox is an API which allow us to access the maps data of any place in the earth and with the help of its API we can create our own map layout and design some places and objects inside.

It is an open source API which does not charge any cost until the user wants to make it public.

It also provides Augmented Reality features to spawn digital objects at any location in the map.

As we are only developing this project for the learning purpose for now so we prefer implementing it on our project.

UNITY

Unity is a Development Platform for 3D, 2D, VR, AR game development, animation etc.

It is an open source development platform which gives us a user friendly environment to work with and may more important assets. We used it as our editor because it is easy to use and has very impressive and powerful tools provided for developers.

WIT.AI API

It is an open source Natural Language Processing (NLP) Interface provided to users for developing an Artificial Intelligence (AI) Bot which can interact with user's inputs.

We used this API so that our virtual assistant can interact with the user with the help of this API and AI technology which makes our assistant intelligent and smarter to give the correct responses back to the user as assumed.

C#

C# is a programming language which is supported in the Unity for the development and it is easier to get a grip using it in our project.

It is an object oriented programming language developed by Microsoft which runs on .NET Framework.

It can be used to develop web application, games, windows applications, mobile application and many more.

2.1 Methodology

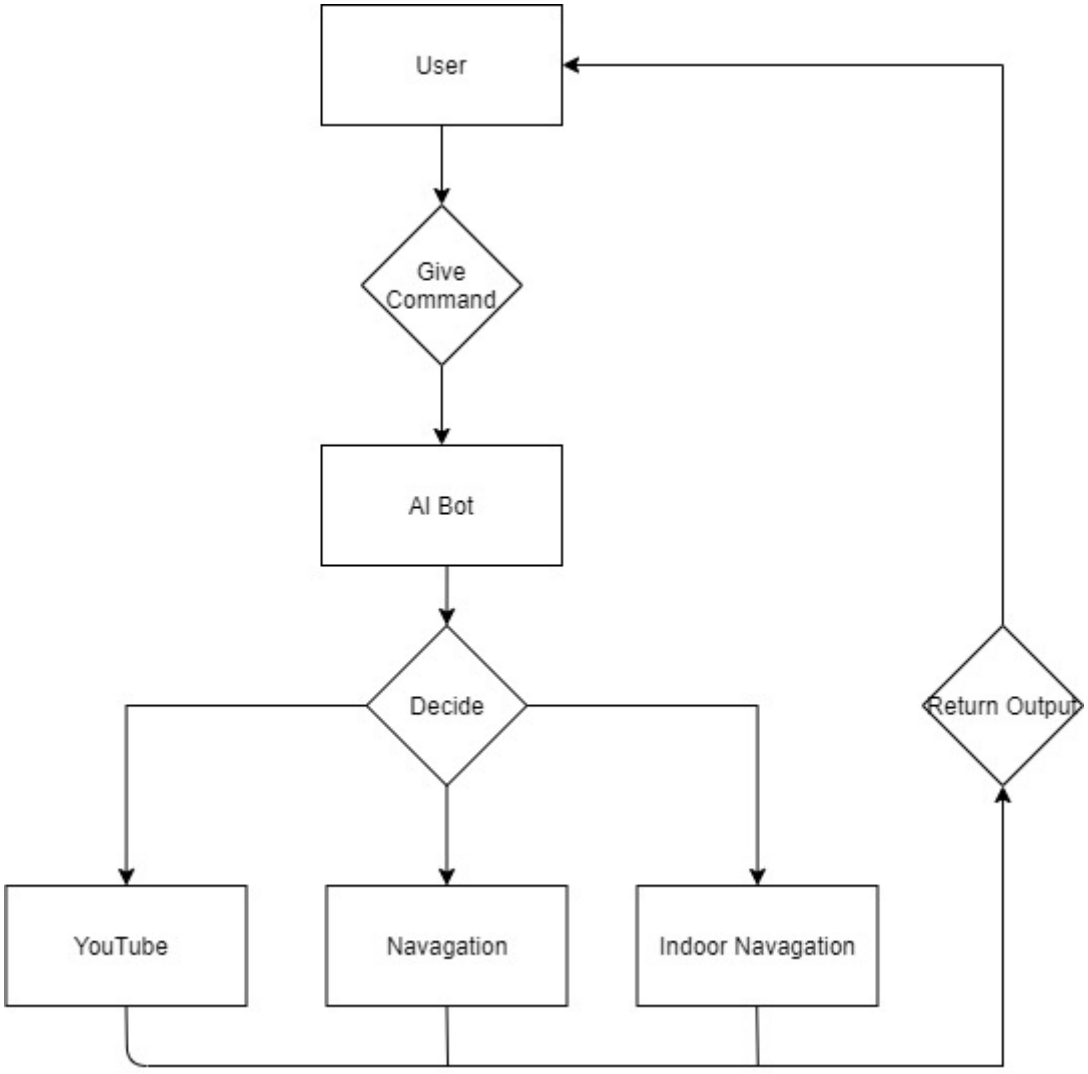
VISUAL STUDIO CODE

It is a free code editor which supports many features like code refactoring, intelligent code completion, debugging and many more features for the developers.

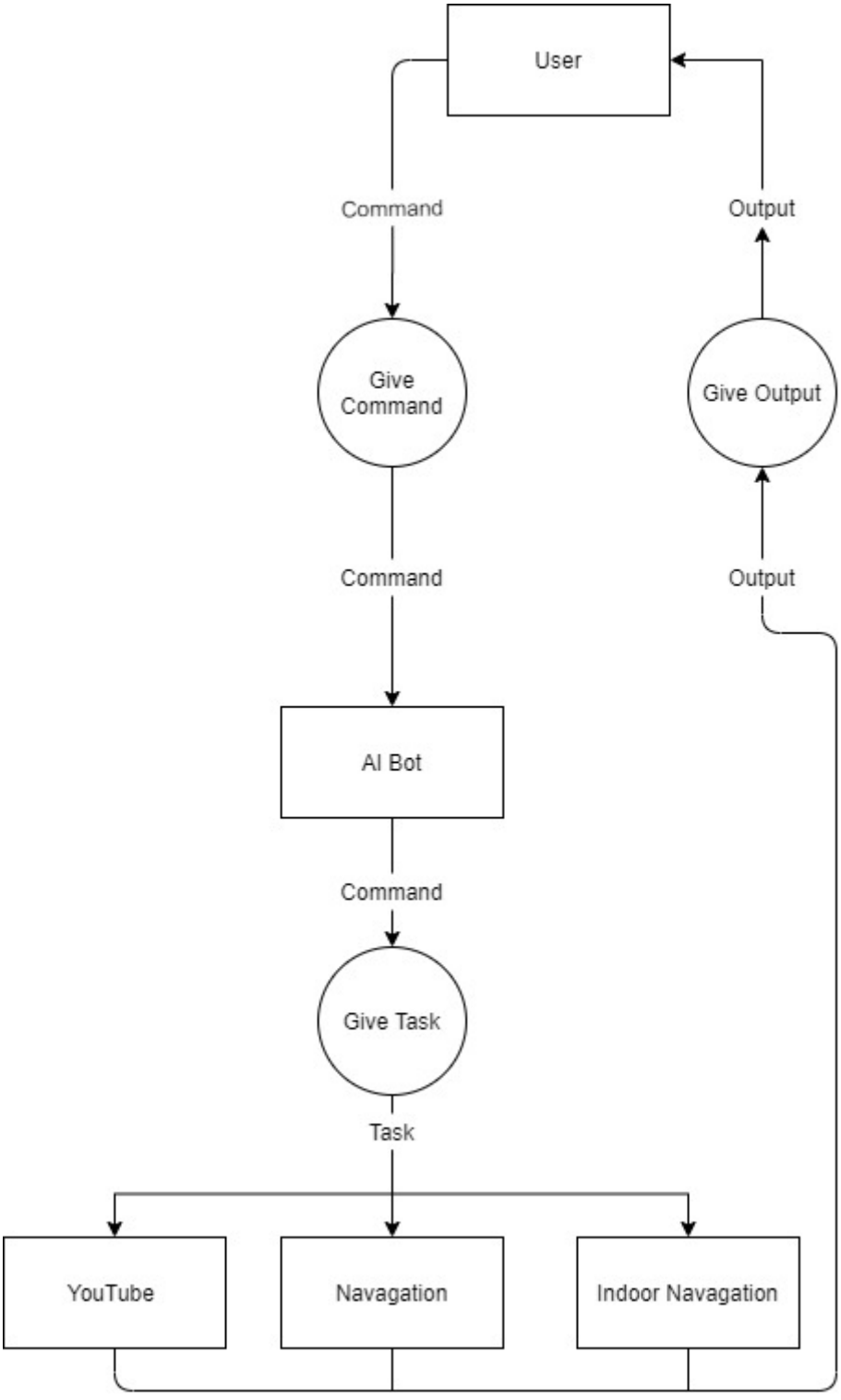
It supports C#, HTML, XML and many other languages for the developers.

As for unity it comes along with the unity as visual studio community version which is free version of visual studio therefore we used visual studio community for development of our project.

2.2 ER DIAGRAM



2.3. DFD


User
Command
Output
Give Command
Give Output
Command
Output
AI Bot
Command
Give Task
Task
YouTube
Navagation
Indoor Navagation

CHAPTER 3. Present Work

3.1 Objectives

The objective of our project is to give the user a enhance way of using a virtual assistant for which we have choose to add Augmented Reality as an head up display and it also provides a good interaction to the application. The main objectives of this project are:

- Making a world scale AR map where the user can view the navigation routes directly in the streets or in the roads by the help of head up display.
- Making a way to implement YouTube search and be able to view YouTube videos in the Augmented Reality environment.
- Implementing a way for Indoor Navigation system for Augmented Reality environment.
- Making an AI Bot for getting user responses and take actions accordingly.

3.2 Making an AI BOT

We have implemented an AI Bot with the help of wit.ai API which allows us to create and train our own AI Bot by the help of reinforcement learning algorithm.

First we train our AI Bot in wit.ai interface and accordingly the Bot gets more and more intelligent and accurate to give the correct response back to our application.

Second we fetch the output of the AI Bot we just created by the help of C# script, AI Bot returns the output in the form of a node.js format so we can easily read it by the help of c# get function.

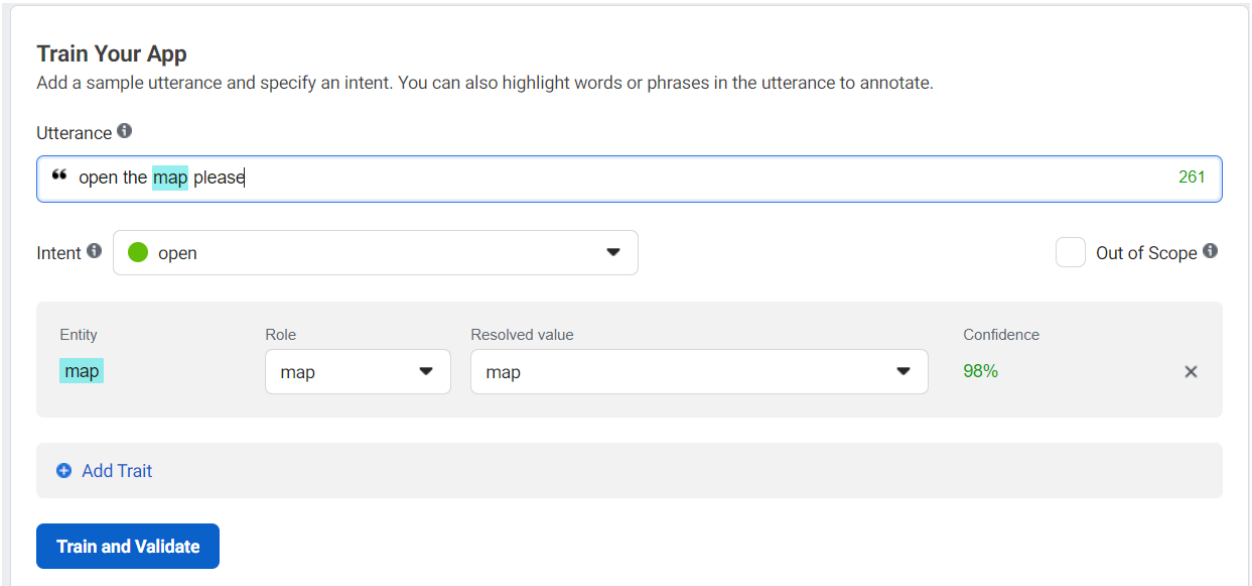


Fig 3.2.1. Training the AI Bot.

Fig 1.1 shows us how we have trained the AI Bot, First we have to write a sentence in the utterance block assuming the user can say that to the Bot then we have to give it an intent in this case which is “open” an intent is what the user wants to be done like in this case we want to open the map and an entity is what is to be done with respect to the intent like here we want to open the map so map is an entity and resolved value is what is to be returned from the AI Bot as an output to our application then at last confidence is how much the AI Bot has the confidence to say that this particular action is to be done or returned if this is the utterance and that is the intent. Which in this case it is showing 98% which is good it can be increased more and more by making more and more utterances and train our Bot more.

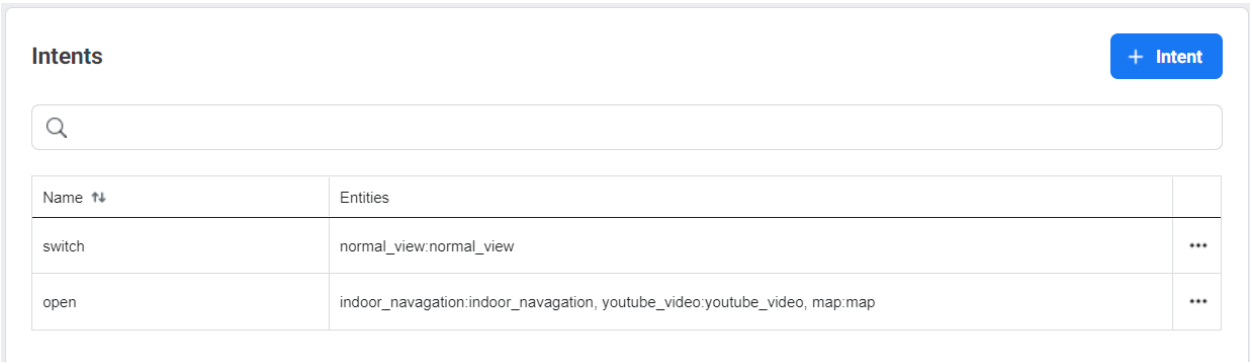


Fig 3.2.2. Intents.

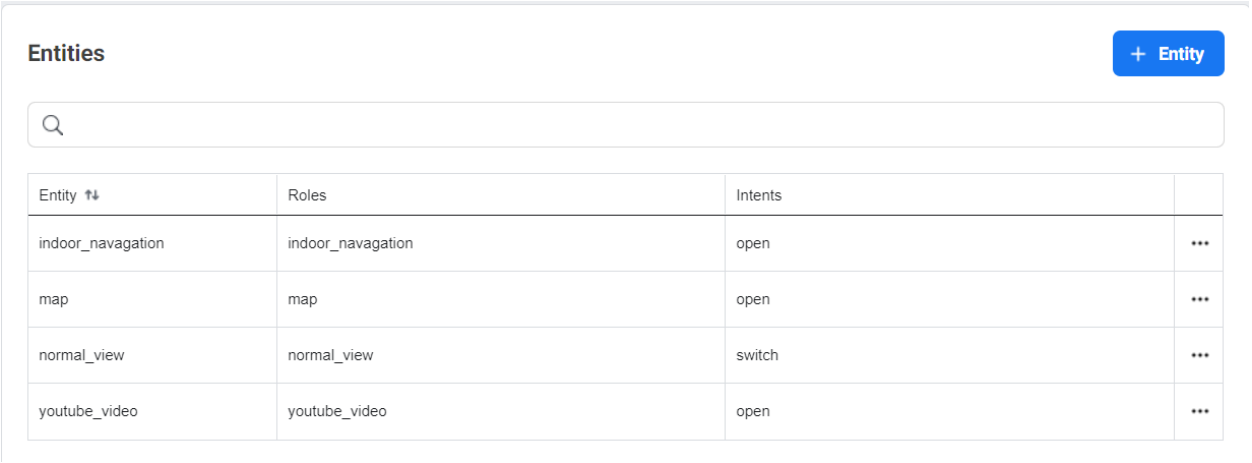


Fig 3.2.3. Entities.

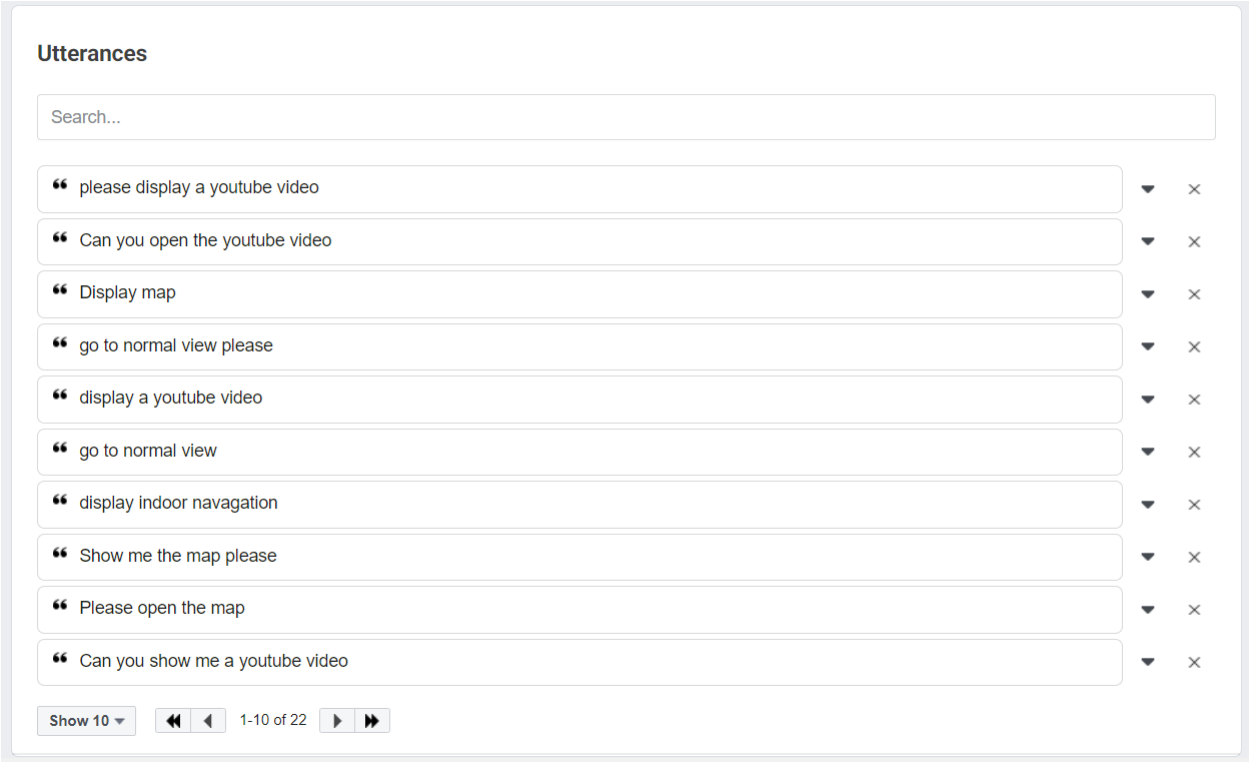


Fig 3.2.4. Utterances for AI Bot.

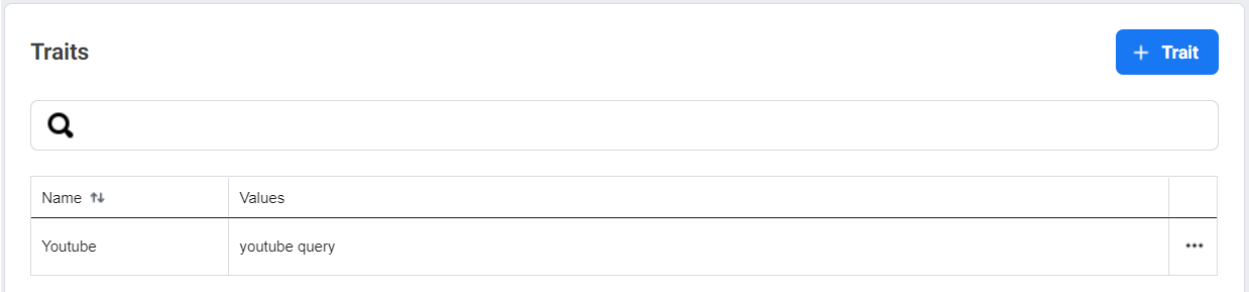


Fig 3.2.5. Traits of AI Bot.

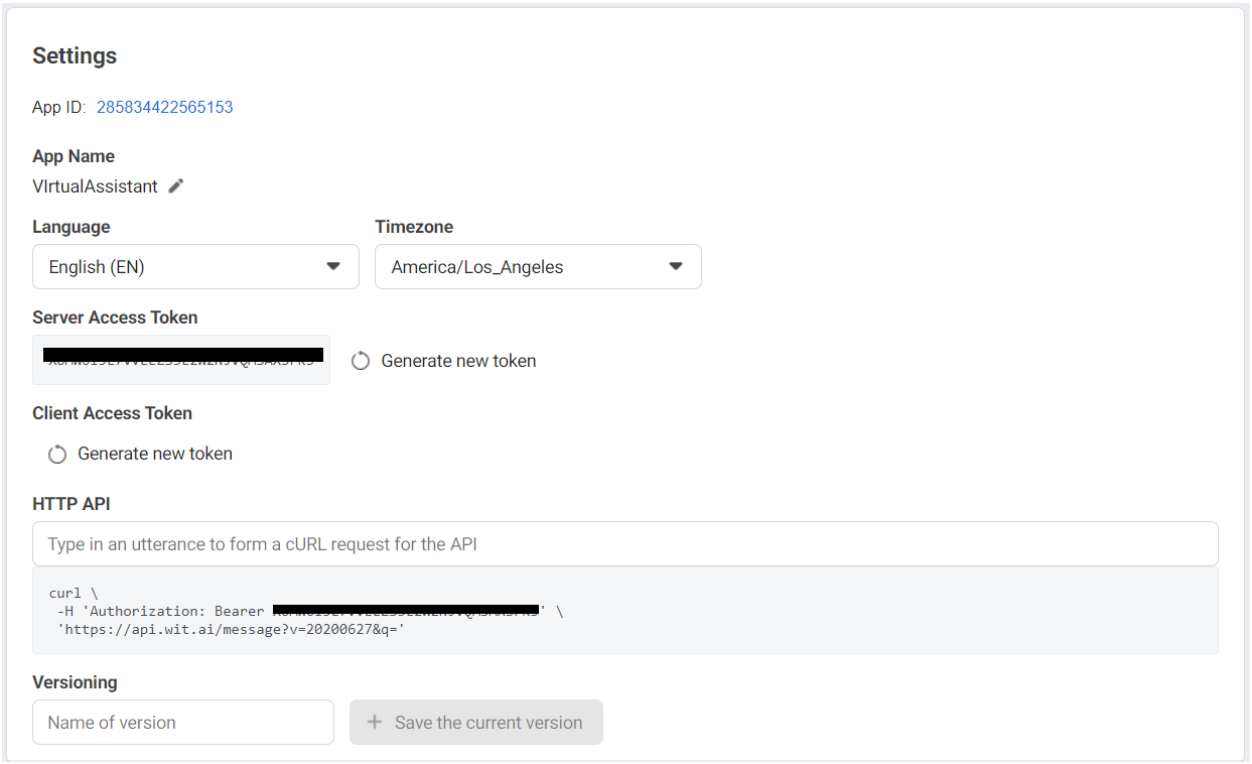


Fig 3.2.6. Settings of AI Bot.

3.3 Implementing the MapBox SDK in Unity

For this, first we have to create an account in the mapbox official website then we get a public access token which is needed later, then we have to download the SDK for unity and then we have to import the SDK in unity, after importing the SDK in unity we have to provide our access token which we got after logging in the mapbox.

3.4 Creating a world scale AR map

First we have to open the mapbox AR world alignment demo example and then make duplicate it for our custom use it just provide us the world size map which we can manually align at the run time with the help of roads and buildings alignment.

But because the Mapbox world scale AR example scene make use of old depreciated package of AR Interface and since unity have already updated its contents from AR interface to new package AR Foundation we were unable to do this part of our project for now, So we just created a small 3D map which displays the user current position and nearby stores info etc. And then we used AR Foundation for projecting it in the AR environment as a mixed reality. Previously there was AR-kit and AR-Core for the development of AR applications where AR-kit supports the IOS devices

and AR-Core supports the Android Devices but then AR interface comes by which a developer can directly intricate and can use either AR-kit or AR-Core but now AR Foundation was launched which can automatically decides what to choose among AR-kit and AR-Core according to the

```
List<ARRaycastHit> raycast_hits = new List<ARRaycastHit>();
public GameObject locationBasedARPrefab;
```

device that is running the application.

```
void Update()
{
    if(Input.touchCount > 0)
    {
        Touch touch = Input.GetTouch(0);
        if (m_ARRaycastManager.Raycast(
            touch.position,
            raycast_hits,
            UnityEngine.XR.ARSubsystems.TrackableType.PlaneWithinPolygon))
        {
            Pose pose = raycast_hits[0].pose;
            locationBasedARPrefab.transform.position = pose.position;
        }
    }
}
```

Fig 3.4.1. Detecting and Placing the 3D Object.

In Fig 3.4.1 first we check the user input as a touch to the screen, then if the screen is touched then do a raycast with that touch position, raycast_hits for storing the output into and PlaneWithinPolygon for detecting planes as a polygon search and matches it further. Then after if the raycast is successful then just placing the 3D object to the position of the raycast_hits. This shows how we have used a raycast method to raycast our 3d map GameObject into the scene.

```
void Map_OnInitialized()
{
    LocationProviderFactory.Instance.mapManager.OnInitialized -= Map_OnInitialized;
    _mapInitialized = true;
    _map = LocationProviderFactory.Instance.mapManager;
}

1 reference
Vector2d LatitudeLongitude
{
    get
    {
        if (_mapInitialized)
        {
            var startingLatLong = Conversions.StringToLatLon(_latitudeLongitude);
            var position = Conversions.GeoToWorldPosition(
                startingLatLong,
                _map.CenterMercator,
                _map.WorldRelativeScale
            ).ToVector3xz();
            position += _targetTransform.position;
            return position.GetGeoPosition(_map.CenterMercator, _map.WorldRelativeScale);
        }

        return Conversions.StringToLatLon(_latitudeLongitude);
    }
}

6 references
protected override void SetLocation()
{
    _currentLocation.UserHeading = _targetTransform.eulerAngles.y;
    _currentLocation.LatitudeLongitude = LatitudeLongitude;
    _currentLocation.Accuracy = _accuracy;
    _currentLocation.Timestamp = UnixTimestampUtils.To(DateTime.UtcNow);
    _currentLocation.IsLocationUpdated = true;
    _currentLocation.IsUserHeadingUpdated = true;
    _currentLocation.IsLocationServiceEnabled = true;
}
```

Fig 3.4.2. Editor Location provider.

In Fig 3.4.2 we find the device location and then project the map of that area only with the help of this script which automatically finds out the current location of the device using the device GPS and then sets displays the map of that location from the mapbox server to our scene.

Then finally we display the 3D map with the help of raycasting it at any detected plane surface

3.5 A way to access YouTube videos

At first we thought of implementing YouTube API directly but then it was costly in the unity's Asset Store, after some research we found a php script which takes the video URL and gives back

the video information like different quality, size, download URL but some time it fails to fetch and after some time it we not working completely so I found one more way, it was an open source project which gives us the same things about the video but it was using the C# script and that was good because my whole project is on C# so we implemented it in our project and then get the videos from the URL and Display on the Augmented Reality Environment by creating a 3D plane and playing video in its mesh renderer.

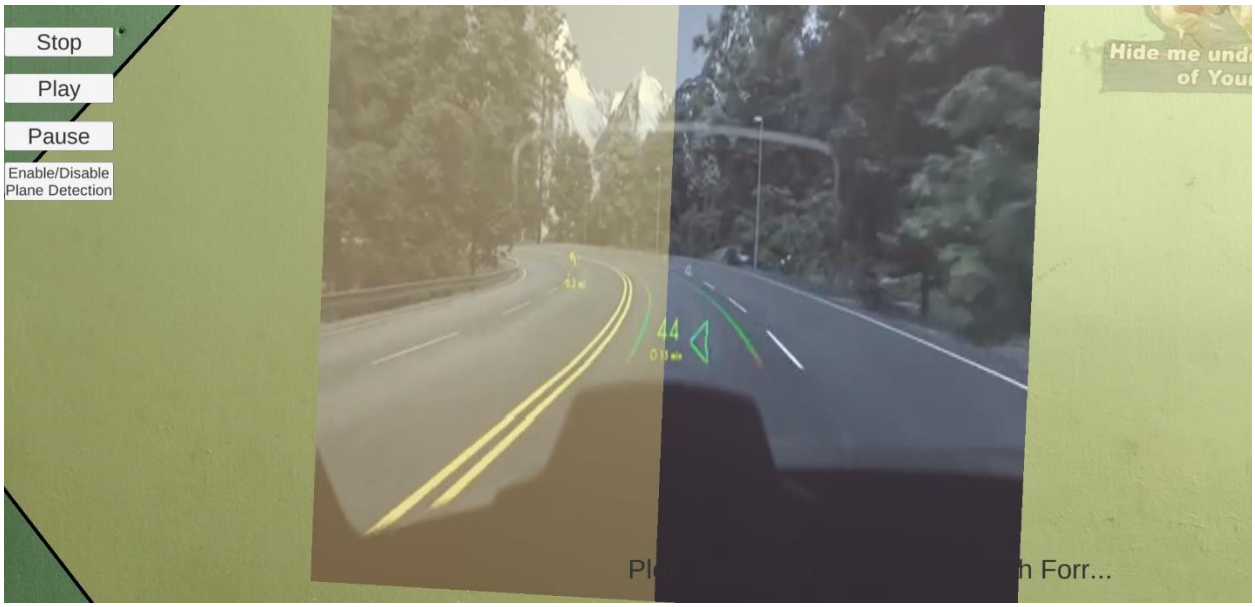


Fig 3.5.1. Playing video by raycasting.

Fig 3.5.1 shows us how we can play video in any plane surface by first scanning the surface with AR Foundation Plane Detection and then placing the 3D object wherever we touch on the detected planes. Then the VideoPlayer renders it to the 3D object and hence the video is played in 3D object.

Fig 3.5.2. Playing video in fixed position.

Fig 3.2 and Fig 3.3 shows us that the video can also be played in a fixed portion of the screen inside the canvas therefore it is playing in the top right corner of the screen which gives the user to view the rest of the real view properly along with the video.

Enable Plane Detection: This button toggles between enabling and disabling the plane detection.

Pause: This button will pause the video.

Play: This button will first ask the user for doing a voice search which records the audio for at least 5 seconds then when the results comes back from the query it plays the first result which was returned from that query.

Stop: This button will Stop the video and only display the original background with buttons visible.

3.6 Implementing indoor navigation

For this first we detect the plane using the device camera and by the help of AR Plane Manager which comes under AR Foundation, it detects plane once per frame so we can visualize the detected plane constantly by doing so we give the user the ability to see if the plane is scanned properly or not visualization is done with the help of AR Plane Mesh Visualizer script, After the user completes the scanning of the plane then he/she can create a path by clicking on the plane and then if he/she press the save path button to save the current path he/she just scanned using the plain detection it is done with the help of plane manager because plane manager directly instantiate a GameObject which can be null initially but while detecting the planes it gets the information. Then After saving the path if the user clicks the follow button then a rabbit shows and follows the path which was said to be followed.

```
foreach (var curPlane in _newPlanes)
{
    var planeObject = Instantiate(TrackedPlanePrefab, Vector3.zero, Quaternion.identity,transform);
    planeObject.GetComponent<DetectedPlaneVisualizer>().Initialize(curPlane);

    planeObject.GetComponent<Renderer>().material.SetColor(
        "_GridColor",
        new Color(Random.Range(0.0f, 1.0f),
        Random.Range(0.0f, 1.0f), Random.Range(0.0f, 1.0f)));
    planeObject.GetComponent<Renderer>().material.SetFloat("_UvRotation", Random.Range(0.0f, 360.0f));
}
```

Fig 3.6.1. Plane Visualizer.

Fig 3.6.1 shows how we have instantiate the trackedPlanePrefab into the scene and set the colour of it as random between 0.0f to 360.0f.

```
if(Input.GetMouseButtonDown(0) && followPathBool == 0)
{
    var ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    RaycastHit hitInfo;

    if(Physics.Raycast(ray , out hitInfo))
    {
        hitArr[i] = hitInfo.point;
        var go  = GameObject.Instantiate(arrowGameObject , hitInfo.point , Quaternion.identity);
        go.GetComponent<MeshRenderer>().material.color = Random.ColorHSV();

        gameArray[i] = go;
        i++;
    }
}
```

Fig 3.6.2. Raycasting for user defined arrows.

Fig 3.6.2 tell us about how we are instantiating the arrow game object whenever the user click anywhere in the visualized path. So that gives user the ability to make a path by instantiating multiple arrow game object to make a user defined path.

```
public void followPath1() {

    if(rabitVisible == true) {
        Destroy(travellingSalesman); // current rabit Destroy
    }
    rabitVisible = true;

    for(int x = 0; x < path1Len; x++) {
        hitArr[x] = path1Array[x];
    }
    i = path1Len; // i = n

    drawPath(path1Array, path1Len); // redraw, gameArray store
    follow();
}
```

Fig 3.6.3. Follow selected path.

The above figure shows us how we move the helping rabbit to move in the selected path with the help of MoveTowards method we can make its position to the new position i.e. the next arrow object in the path.

```
public void reset() {
    for(int x = 0; x < i; x++) {
        Destroy(gameArray[x]);
    }
    initialize(); // i = 0, len = 0

    if(rabitVisible) { // if rabit moving
        followPathBool = 0;
        rabitVisible = false;
        Destroy(travellingSalesman);

        rabitVisible = false;
    }
}
```

Fig 3.6.4. Resetting the rabbit to its initial position.

Above Fig helps to reset the position of the rabbit if it is being following the path this is called when the reset button is pressed by the user.

```
public void path1Save() {

    if(rabitVisible) return; // if rabit moving

    rabitVisible = true;
    path1Len = i; // n
    for(int x = 0; x < path1Len; x++) {
        path1Array[x] = hitArr[x];
    }
    reset();
    initialize();
}
```

Fig 3.6.5. Saving the Path.

Above fig explains us how the paths are saved in the arrays like in this case path one is saved in the path1Array[] array. This will be done only when the user click the save path one/two button.

CHAPTER 4. RESULTS AND DISCUSSION

The outcome of this application is beneficial to the user in many ways likely it have the features for entertainment like watching YouTube videos while walking on the street. It gives user an indoor navigation system with the help of augmented reality thought of a big mall of shopping complex if you can't find your way to a certain shop or someplace then you can easily navigate to their using this app. Finally the main part of this project was the outdoor world scale navigation which could be a big help for outdoor navigation system because then the driver does not have to look away from their usual view to see the map but until mapbox does not updates there unity SDK it may not be helped so instead of world scale AR navigation we have added a simple AR map with many POI's available to view nearby locations and own location.

Fig 4.1. AR 3D map.

4.1. Results of module 1:

Fig 4.1 shows that how a user can see the 3D map in any place by raycasting it to any place where the user's location is shown by the help of a yellow "I" mark and the map not only shows the streets and roads but also other POI's Like the Arts, medical, Food etc. and the buildings as 3d models in the map according to the size of the map.

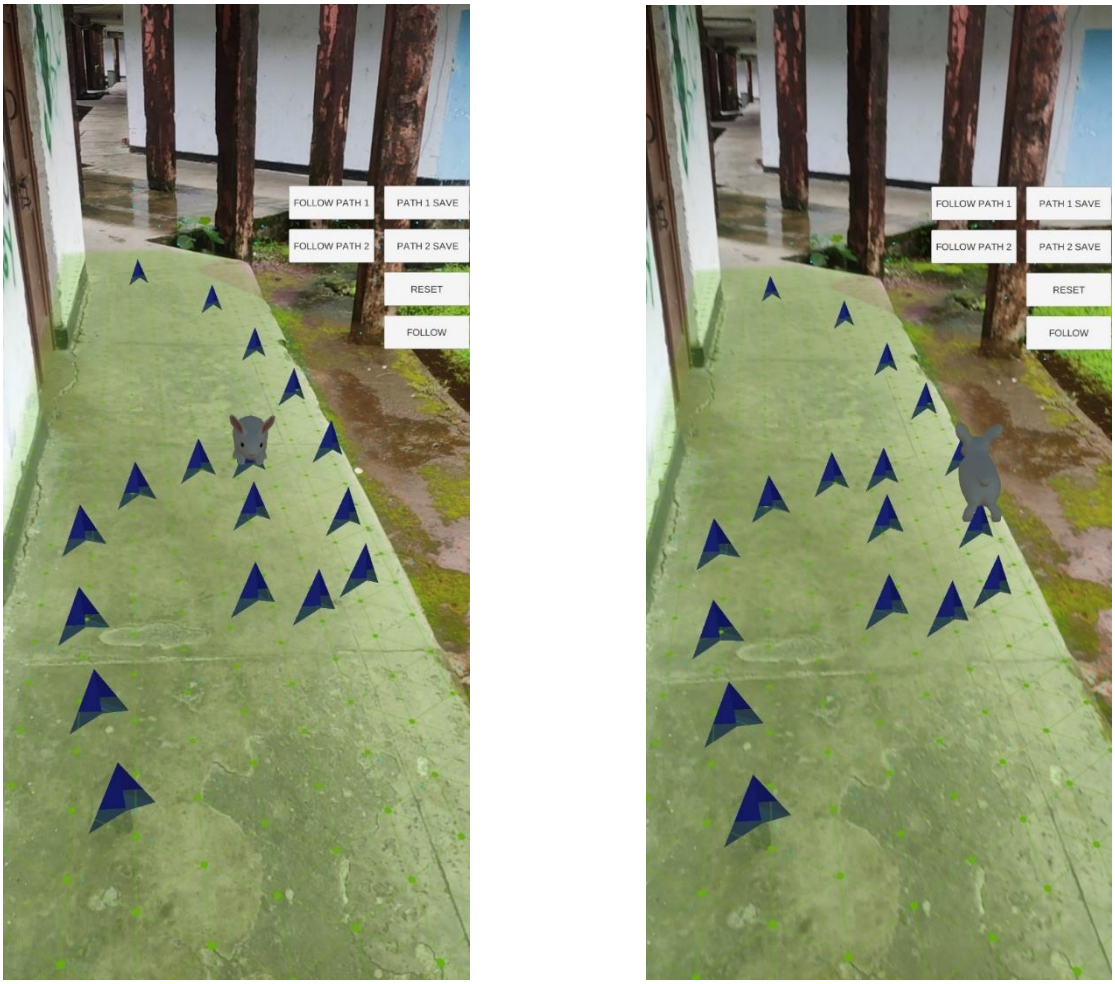


Fig 4.2. Indoor navigation.

4.2. Results of module 2:

Fig 4.2 shows the outcome of the indoor navigation which is implemented in this project.

It shows the visualized plane as green and the blue arrow objects placed by the user to define a path and a helping rabbit which assist the user to follow the path correctly by following the path by himself. If the user want then he/she can save two different paths for later use in the application and can easily recreate a new path or delete the previous path which is currently saved by creating a new path. And finally the user can select any path which is already created and can follow the same path. In Fig 4.2 the rabbit is following the “path1” path.

Fig 4.3. Youtube video in fixed position.

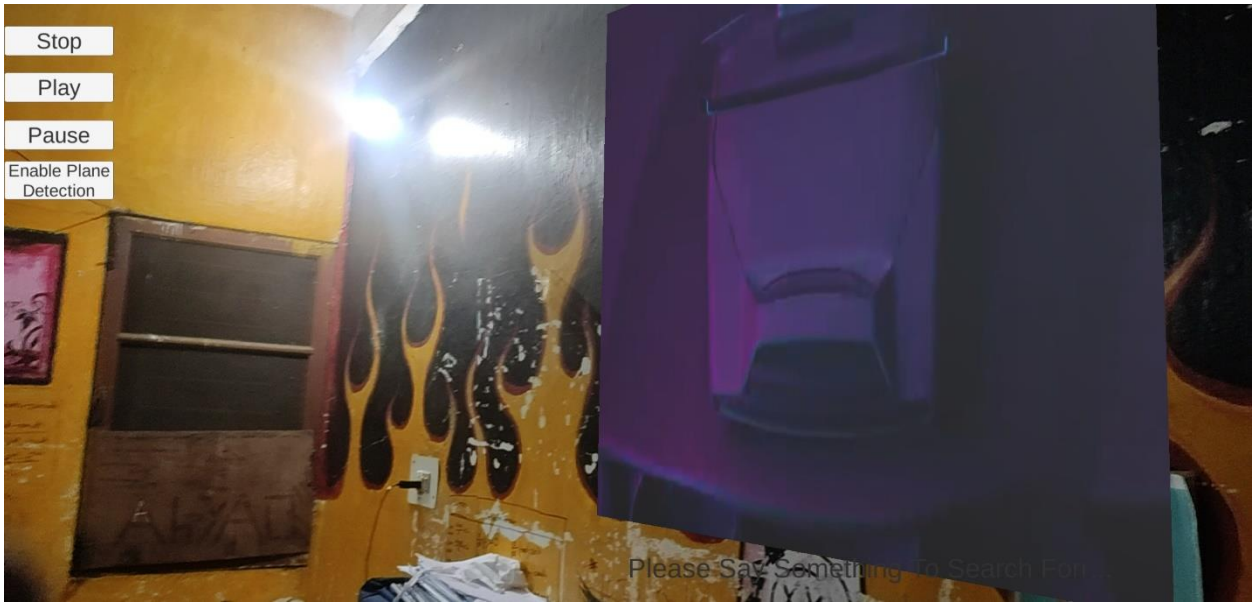


Fig 4.4. Youtube video in user defined position.

4.3. Results of module 3:

The fig 4.3 shows that how the user can view a video in a fixed position of the display.

Fig 4.4 show that the user can scan do the raycasting by the enable plane detection button on the screen and then when the plane is detected he/she can then touch anywhere in the detected plane and spawn the video there to played in that particular fix position.

Fig 4.5. Video playing in a horizontal surface.

Fig 4.5 show us that the video can be placed and played in the ground surface by the help of enable plane detection button which will start detecting the horizontal plane and then the user can spawn the video in the horizontal plane by clicking anywhere in the detected plane.

CHAPTER 5. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

This may not have been a fully satisfied project but while doing this we have learned a lot from it and about some advance technologies like Augmented Reality, Virtual Reality etc. But the YouTube module and the indoor navigation module is working fine along with the AR map view module.

5.2 Future Scope

- To change the AR map view model to World Scale AR model where the POI's can directly be shown in the World Scale AR view.
- Implementing the AR browser in the YouTube model so that the user can not only see the videos but also can browse the web in the AR view.
- In indoor navigation module it can be modified as well for saving the paths in the device secondary memory for getting access to the saved paths and visualize them even after the app is restarted.

This project can be more modified in future for the better and exact responses from the application about the augmented reality objects positioning and the virtual assistant may interact with the user and can do some more tasks for the user. The augmented reality technology is starting to become productive in many areas and if it will be there for a kind of a virtual assistant for the users so then the user can take decisions correctly, faster and easily.

References

Links

1. https://www.youtube.com/watch?v=IKzqNpWC9WI&t=350s
2. https://www.youtube.com/watch?v=1golaHxl4wM&t=204s
3. https://www.youtube.com/watch?v=lMLSACaaOjU
4. https://www.youtube.com/watch?v=VOMysKbDNxk&t=451s
5. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.1/manual/plane-manager.html
6. https://www.youtube.com/channel/UCIHl9sd2brgvjBlSETKYDcg
7. https://www.udemy.com/course/build-locationbased-ar-apps-with-mapbox-unity-sdkandarcore/learn/lecture/13866886?start=75#content
8. https://github.com/ashishgopalhattimare/PathMarkAR/blob/master/README.md