

AUGMENTED REALITY INDOOR NAVIGATION SYSTEM

Mantripragada Sai Pavan Aditya^{*1}, K. Anvesh^{*2}, Kolluri Sai Madhav^{*3}

^{*1,2,3}Department Of Computer Science & Engineering, Matrusri Engineering College,
Hyderabad, Telangana, India.

ABSTRACT

Navigation systems help users to access unfamiliar environments. Current advances in technology enable consumers to encapsulate these systems into mobile devices, which expands navigation systems' popularity and the number of users effectively. In indoor conditions, navigation with orbiting satellites becomes more difficult relative to outdoor environments, due to the Global Positioning System (GPS) error rate of 3 meters.

This project intends to solve this by designing and building an indoor Navigation system, which I achieve with the assistance of a mobile device's camera and gyroscope, my approach would be to use AR Core to show the navigation path to the user on his mobile phone along with a 2D mini-map for guidance where I track the initial position of the user by making user scan a QR code at a specific start location. And another objective of this project is not to depend on any specific hardware or Internet access for navigation.

Keywords: Indoor Navigation, 2D Mini-Map, Navigation System, AR Core, QR-Code, Navmesh.

I. INTRODUCTION

Our inspiration for this project comes from the fact that people increasingly rely on their smartphones to solve some of their basic everyday problems. One such challenge that smartphones are not yet thoroughly tackled is indoor navigation. There is currently no low-cost scalable smartphone solution available on the market that effectively navigates the consumer from one location to another indoors. Such an app will help users who are unfamiliar with the area. Tourists, for example, would be able to maneuver confidently inside a tourist attraction without assistance. In places such as museums and art galleries, the application could be expanded to include the most optimal or 'common' paths.

Such a device may also be incorporated at airports to direct travelers to their gates of entry. Similarly, an indoor navigation system could also benefit local users who have previously visited the location but are still unaware of the whereabouts of some of the desired items. These include supermarkets, libraries, and shopping malls. This technology could also help customers who have deployed the system, by learning customer behavior and using it to promote ads at specific places or on specific routes and collect valuable data about the customer.

II. EXISTING SYSTEM

For indoor navigation system companies are majorly using technologies are Bluetooth beacons, Wi-Fi-Fingerprinting, and UWB. even those these technologies are widely used but there are some problems with these technologies they are:

- In Bluetooth technology, a huge number of beacons should be installed which are very costly. And it makes it difficult for detecting the devices when more connectivity requests are sent.
- In Wi-fi fingerprinting technology the major problem concerns the security of public Wi-fi. As the frequency varies as we move from one place to another it leads to connectivity problems which further leads to loss of tracking.
- In UWB technology the major problem is the signal strength released by the UWB device. This device releases very low signals which creates difficulties in signal catching and tracking the position of a person.

Apart from these technology-specific problems they also have more general disadvantages like Expensive Implementation, Need for hardware installation, Dependent on internet access .

III. PROPOSED SYSTEM

Our solution will be using AR Core to display the navigation path to the user on his mobile phone with the help of Anchors which are based on the nav path generated with the help of navmesh. Here the navmesh is created with the help of a floor plan of the area to be navigated.

To track the user's position, we use some simple calculations of distance and angle from the previous position to the current position. The user will also have a 2D mini-map for reference while trying to navigate.

To run AR Core Applications the device needs to be certified as AR Core compatible. Certification is given by Google where they decide which device to certify based on the quality of the camera, motion sensors, and the design architecture is expected to perform to a standard. Also, the device needs to have a powerful enough CPU that integrates with the hardware design to ensure good performance and effective real-time calculations.

IV. OBJECTIVE

The primary objective of the application is to eliminate all the dependencies like Wi-Fi Access points, Bluetooth beacons, Internet and other devices which require additional sources to realize Indoor navigation. We relied on new but a reckonable technology on Augmented reality to help us realize Indoor Navigation on the device itself. This requires the mobile handset to be ARCore certified from Google during manufacture(nowadays all the units come with this capability) and a QR Code preprogrammed with the location indoors.

V. SCOPE OF THE PROJECT

Having access to advanced handsets people tend to navigate large indoor spaces hassle-free as they might be puzzled looking at larger structures and multiple sign boards, they tend to use an easy way to access all the directions on their phones. This brings Indoor Navigation systems into play. Having an Augmented Reality based Indoor Navigation system at their disposal makes users navigate seamlessly and also aid vendors for advertising in malls etc.

VI. SYSTEM DESIGN AND ARCHITECTURE

The User Interface is the primary way through which the user interacts with the Android application where the whole backend is being performed. In the background, both the Augmented Reality Core (ARCore) and the Android SDK are connected to Unity Engine. The data from QR-Code is retrieved by using ZXing which is a library we use in some of the scripts.

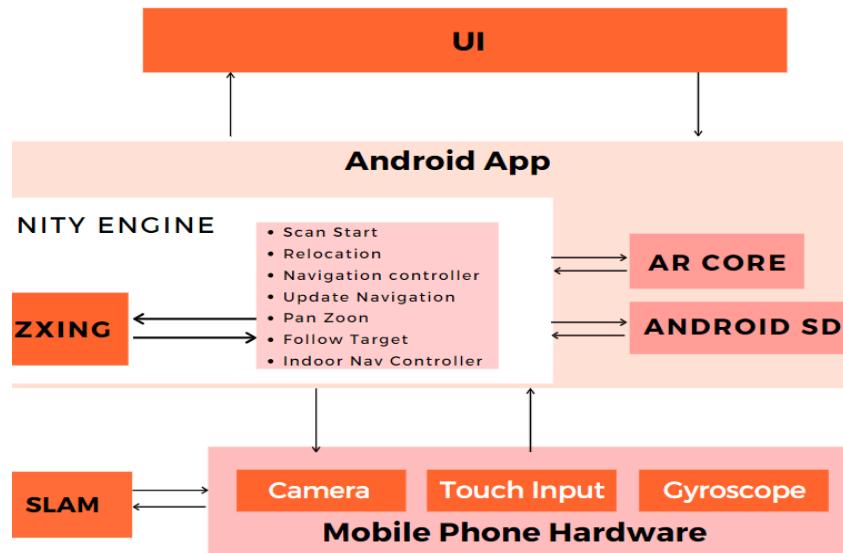


Fig 1: System Architecture

The User Interface is the primary way through which the user interacts with the Android application where the whole backend is being performed. In the background, both the Augmented Reality Core (ARCore) and the Android SDK are connected to Unity Engine. The data from QR-Code is retrieved by using ZXing which is a library we use in some of the scripts.

Apart from the Software application, the physical requirements (Mobile phone hardware) i.e., The position of the Camera and the gyroscope are used by ARCore along with SLAM to process any visual being displayed to the user.

VII. IMPLEMENTATION AND WORKING

7.1 Working

The first step in the navigation process is to scan a QR code. As the user opens the application, the user gets an interface for scanning QR-code, once the user points his device camera towards a QR code the application starts operating. The QR-codes are placed at key locations where the users are most likely to pass from. As the user scans the QR-code, the user is presented with a drop-down box where they can select the destination and a clear view button to clear the previous destination if selected.

To aid the user with smooth navigation, there is also a mini-map of the floor, locating his current position and also showing a path to his destination which the user can zoom in and pan if needed.

To start navigation the user needs to select his desired destination from the drop-down box, thus setting the path from the starting position and the destination, then the user is shown an AR floating arrow (AR-based anchor) which navigates the user towards the destination in the real environment. As the user reaches his/her destination an AR floating PIN is placed at the destination which indicates that the user has reached the destination.

7.2 Scanning

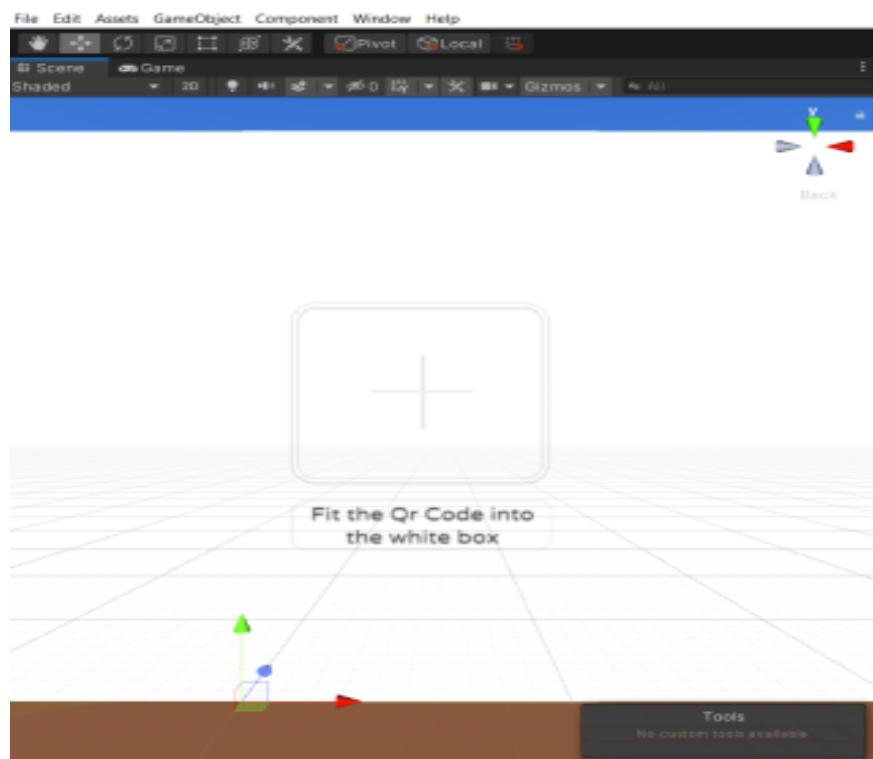


Fig 2: View of QR code Scanning

In Scanning (Scan.cs), immediately after the application has been opened by a user, it checks for availability of rear-view camera then updates a variable which is used to track the availability of camera, which in turn tells that there's a camera available for scanning QR code. Then it scans the QR-code, which is nothing but capturing a still image which is converted to a stream of bytes which is sent to ZXing, (here ZXing function resides in another script called image recognition.cs) once After ZXing gets the data, it parses it and gives the resultant string back to the called function as a string is compared to the list of start positions, we coded in beforehand and if there is match the result. (result is sent back to scan.cs).

After retrieving the data from the QR-code the system starts checking all the starting positions in the map provided by the map in Unity engine, and if any one of the start positions in the respective map is matched with the data which is retrieved from the embedded QR-code. Then, that position would be set as the user's starting position and the system gets notified of the position of the user.

7.3 Path Finding and Navigation

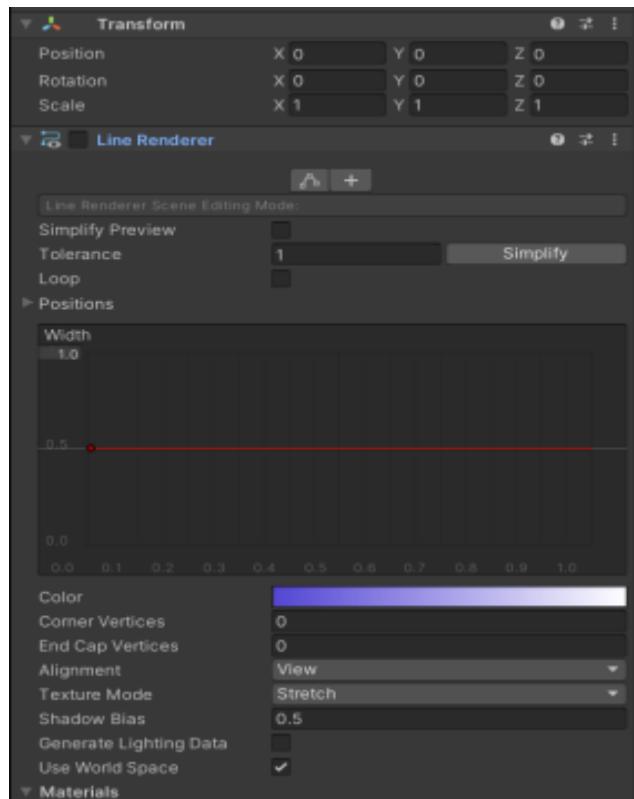


Fig 3: line render to draw a path on min-map

Once the user selects the destination from the drop-down box provided in the Interface, the system finds an optimal path using the Indoor navigation controller class where the class uses a function called Mesh.CalculatePath where it is calculated based on the navmesh we have been created beforehand thus setting the path between the start position and the destination and here the path is updated repeatedly as the user starts moving.

Once the user has set the destination and the path is generated, an anchor is instantiated based on the current position of the user and the path, here the anchor we use is a floating arrow, which also so shows the direction to follow, here we calculate the angle of the arrow based on the current and previous position as following code.

```
float angle = Mathf.Rad2Deg * (Mathf.Atan2(personHelp.y - personPos.y, personHelp.x - personPos.x) - Mathf.Atan2(node2D.y - personPos.y, node2D.x - personPos.x));
```

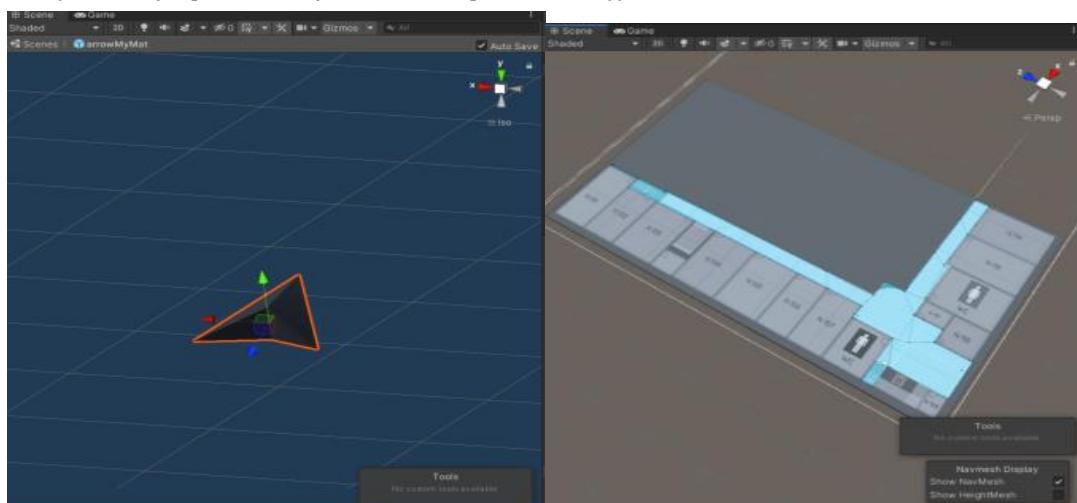


Fig 4: Navigation Anchor

Fig 5: NavMesh

Here, person Help is an invisible cube that is placed in front of the user which is used to calculate tangent, and persons are the position of the user. By using these values and MathF.Atan2 (Single, Single) Method which is used to return the angle whose tangent is the quotient of two specified numbers, we calculate the angle the arrow is to be rotated.

As the user starts following the first anchor the next new anchor is placed on the path, this is continued till the user is reached to the destination as the user reaches the destination a PIN is placed at the destination point to show the user that he has reached the destination. If the user gets out of the path of navigation the application starts alerting the user by a message on the screen. So, that the user can reach the correct destination.

VIII. TESTING

The testing phase was done as a proof of concept that this application works just fine. It was carried out in two parallel phases amid COVID-19 restrictions in a limited space of our Department of Computer Science & Engineering only, confining to our theoretical application works in reality.

The parallel phase of testing is done for tweaking and adjusting the Navmesh, Navigator icons and 2D mini-map tracking of the current location of the user in real-time. The results were astonishingly promising that it works for multi-level structures and large public places effectively.

The application offers a list of destinations for the user to choose from a dropdown and navigates across the confinement space seamlessly.

IX. RESULTS



Fig 6: Scanning QR Code

Fig 7: Retrieve user position

Fig 8: Selecting Destination

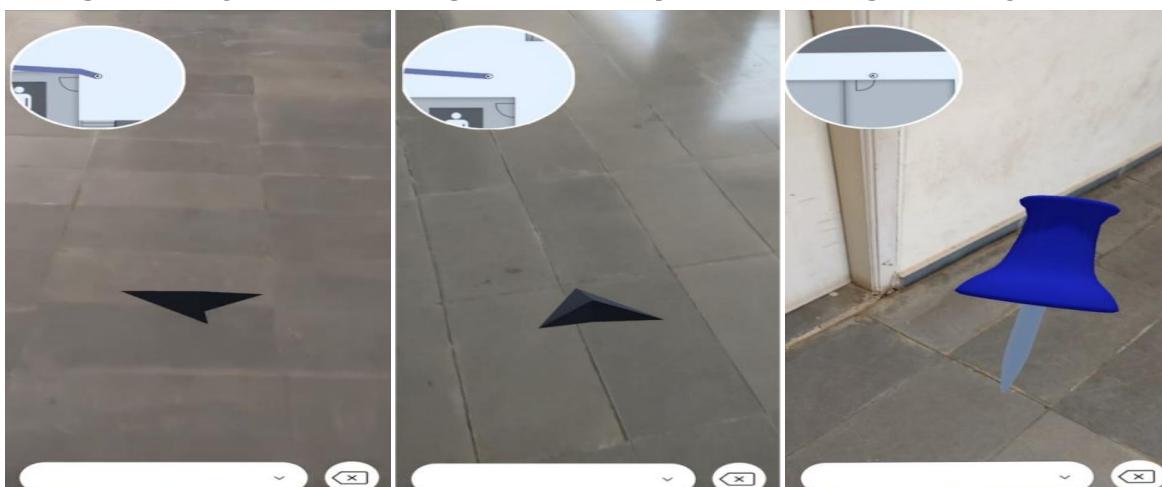


Fig 9: Navigation in corners

Fig 10: Navigation in walkway

Fig 11: Final Destination

X. CONCLUSION AND FUTURE ENHANCEMENTS

Our project can successfully help navigate a user in a given space without any additional resources like external hardware or even the Internet. In this project, the space we choose to build is our department floor and the approach we have used can be adapted to any other place.

Even with these advantages, it should be kept in mind that there is a need of making a NavMesh of the location be navigated, but its great trade-off compared to buying expensive hardware or installation of such hardware. Moreover, NavMesh can easily be generated by an existing floor plan.

Lastly, even though we tried to make our project as perfect as possible there are many things which we could have done better or would have added if he had more time, some of these things are listed in the future enhancements section below.

Future Enhancements

The application has a lot of room for improvement and advancement and some of the ideas are:

- Making a plugin to automatically create a NavMesh from a floor plan that follows certain rules.
- Making an augmented reality line to follow instead of reappearing arrows.
- Have a better way to get the starting position of the user other than QR Code.
- Make the application more generalized, such that users have an option to customize destinations.

XI. REFERENCES

- [1] Akshay Sawant, Abhijeet Patil, Yash Shiwalkar, Dr. Sunil Chavan, Indoor Navigation Assistance System For Visually Impaired. (IRJET), DEC(2020), 1389-1390.
- [2] IRJET-V7I12244.pdf
- [3] Fallah N, Apostolopoulos I, Bekris K, Folmer E (2013) Indoor human navigation systems: a survey. *Interact Comput* 25(1):21–33.
- [4] <https://doi.org/10.1093/iwc/iws010>
- [5] He S, Chan S-G (2016) Wi-fi fingerprint-based indoor positioning: recent advances and comparisons. *IEEE Commun Surv Tutor* 18(1):466–490.
- [6] <https://doi.org/10.1109/COMST.2015.2464084>
- [7] Li Y, Zhuang Y, Lan H, Zhou Q, Niu X, El-Sheimy N (2016) A hybrid wifi/magnetic matching/pdr approach for indoor navigation with smartphone sensors. *IEEE Commun Lett* 20(1):169–172.
- [8] <https://doi.org/10.1109/LCOMM.2015.2496940>
- [9] Hilsenbeck S, Bobkov D, Schroth G, Huitl R, Steinbach E (2014) Graph-based data fusion of pedometer and wifi measurements for mobile indoor positioning. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing. pp 147–158.
- [10] <https://doi.org/10.1145/2632048.2636079>
- [11] simultaneous localization and mapping (SLAM)
- [12] Simultaneous localization and mapping - Wikipedia Working with anchors
- [13] <https://developers.google.com/ar/develop/developer-guides/anchors>
- [14] Esther Vaati, What is Android SDK, and How to start using it. July(2020).
- [15] <https://code.tutsplus.com/tutorials/the-android-sdk-tutorial--cms-34623>
- [16] ARCore Overview
- [17] ARCore overview | Google Developers
- [18] ZXing Project
- [19] <https://github.com/zxing/zxing>
- [20] Unity Documentation
- [21] <https://docs.unity3d.com/2021.1/Documentation/Manual/system-requirements.html#player>