International Journal of Engineering, Management and Humanities (IJEMH)
Volume 4, Issue 6, Nov.-Dec., 2023 pp: 159-162 www.ijemh.com

Object Navigation Car-bot

Tejas D R, Srujan A S
Tejas D R, PES University, Bangalore, Karnataka.
Srujan A S, PES University, Bangalore, Karnataka.
Corresponding Author: Tejas D R

Date of Submission: 22-11-2023 Date of Acceptance: 06-12-2023

ABSTRACT:

The project begins with designing a mechanical structure, selecting appropriate sensors for mapping, and choosing an Arduino board. Next, the physical robot was constructed and the modules were electrically connected. The penultimate stage involved writing software to generate a room map. The final objective was to create a map of a small area.

KEYWORDS:

VUFORIA AUGMENTED REALITY SDKs , SLAM ARKIT, XCODE , API'S

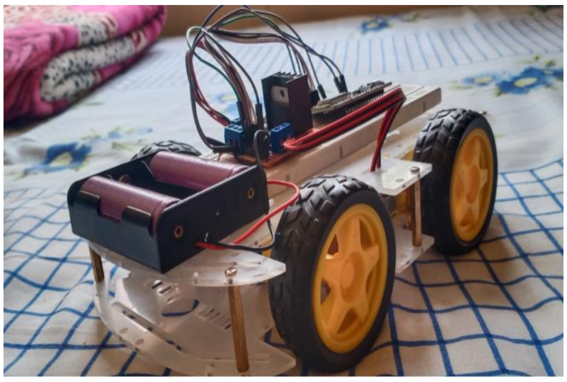
I. INTRODUCTION:

This project focuses on development of a robot that can autonomously navigate. The entire system consists of a robot and an application that is used for monitoring and mapping. The robot is a standalone system and the autonomous navigation is done by the on-board controller that is an Arduino. This project uses an Arduino robot running a ESP-8266 Node MCU that communicates (and drives around) with a mobile phone. The phone is running an app made in the video game engine Unity 3D that does 3 things

1) The first scene allows you to drive around the robot with a camera feed going back to your computer. You can use the arrow keys to drive it around in any direction and the video feed allows you to keep driving even when the robot is out of sight.

2) The second scene allows the robot to track anything you put in front of it. You can click the screen to initialise the tracker and then the robot will follow around that object.

3) The third scene allows you to drive the robot with your computer using the arrow keys. The app uses an augmented reality SDK to find the walls and ceiling which it sends back to your laptop giving you a digital representation of your environment.



II. BACKGROUND WORK:

Vuforia Augmented reality SDK :

Vuforia is an augmented reality software development kit (SDK) for mobile devices that enables the creation of augmented reality applications. It uses computer vision technology to recognize and track planar images and 3D objects in real time. This image registration capability enables developers to position and orient virtual objects, such as 3D models and other media, in relation to real world objects when they are viewed through the camera of a mobile device.

The virtual object then tracks the position and orientation of the image in real-time real time situations so that the viewer's perspective on the object corresponds with the perspective on the target. It thus appears that the virtual object is a part of the real-world scene.

The Vuforia SDK supports a variety of 2D and 3D target types including 'markerless' Image Targets, 3D Model Target, and a form of addressable Fiducial Marker, known as a VuMark. Additional features of the SDK include 6 degrees of freedom device localization in space, localised Occlusion Detection using 'Virtual Buttons', runtime image target selection, and the ability to create and reconfigure target sets programmatically at runtime.

Unity 3D : Unity gives users the ability to create games and experiences in both 2D and 3D, and the engine offers a primary scripting API in C#, for both the Unity editor in the form of plugins, and games themselves, as well as drag and drop functionality. Prior to C# Being the primary programming language used for the engine, it previously supported Boo, which was removed with the release of Unity and a version designed using JavaScript called *UnityScript*, Within 2D games, Unity allows importation of sprites and an advanced 2D world render. For 3D games, Unity allows specification of texture compression, mipmaps, and resolution settings for each platform that the game engine supports, and provides support for bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to- texture and full-screen post-processing effects.

Supported platforms

Unity is a cross-platform engine.The Unity editor is supported on Windows, macOS, and the Linux platform, while the engine itself currently supports building games for more than 25 different platforms, including mobile, desktop, consoles, and virtual reality. Which supports different Platforms like iOS, Android,Tizen, Windows,Universal WindowsPlatform,Mac, Linux,WebGL, PlayStation 4 PlayStation Vita, Xbox One, Oculus Rift, Google Cardboard, Steam VR, PlayStation VR, Gear VR,Apple's ARKit, Google's ARCore, Vuforia,and Magic Leap.

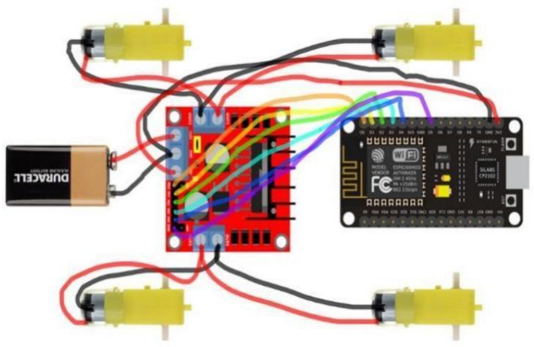
III. IMPLEMENTATION:

Parts used here are ESP-8266 Node MCU, l298N Motor Driver, Assorted Arduino Wires, Robot Chassis, wheels, battery and Unity App. After assembling the parts, make the connections.

To connect the motors and Arduino board we will use an l298N dual H bridge, this among other things allows us to spin the motors in different directions so our robot can turn in any direction.

This thing can take in up to 12 volts and has an on-board voltage regulator which outputs 5V, the perfect amount for powering our Arduino. Connect everything according to the diagram above. Make sure the motors are facing the same direction.

Essentially, the motors on each side are getting wired together so they can being controlled as one motor. This is because the motor controller only allows for a maximum of 3 motors. Even so with two motors we can still get this thing to turn in any direction by making both sides go in opposite directions.edit the code in the to use your network name and password. We do this so your Node MCU can connect to your WIFI network and send packets of information to your phone and computer.



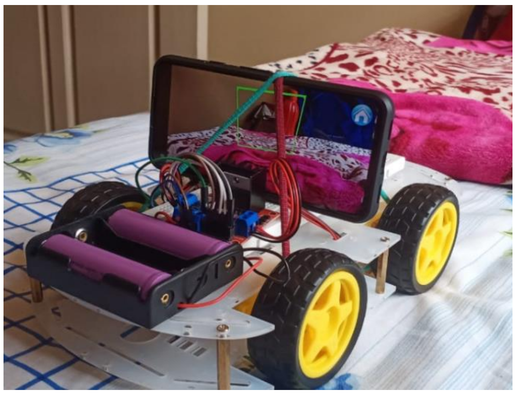
IV. WORKING:

All of the communication here is done via WIFI so phone and computer should be connected to the same local WIFI network. With a robot plugin in via USB, open up the serial monitor in the Arduino IDE to 115200 BAUD. Hit reset on the Node MCU and wait for the board to connect to your network. It will print out its IP address.

First Scene is the camera scene. AR Camera the server script is what takes the video feed from Vuforia and streams it to the computer via TCP.

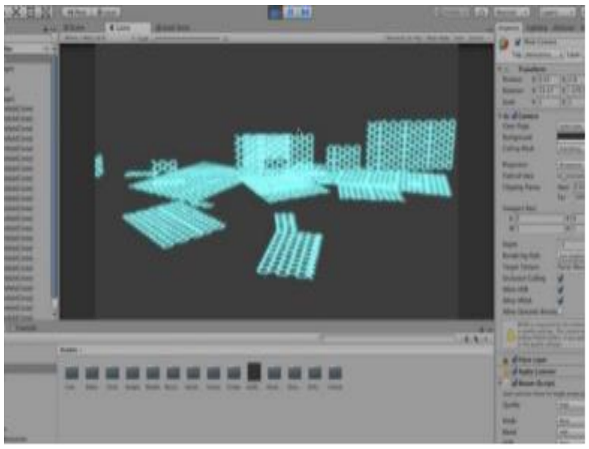
The controller scene is what receives the video stream and also allows the user to use the arrow keys to control the robot. The phones ip address should be included into videoclient.game and the node ip address should be included in onkeybord.game object.

The next scene is the "follow" scene. This scene uses Vuforia's user defined targets so We can create a trackable object at runtime as long as it has enough feature points. On the robot "follow manager" must be made sure to put in the IP address of Node MCU in the send message script. If the image has enough feature points it will say image quality is high and tracking will begin. As the battery power diminishes this tracking behaviour slightly changes . The robot follow behaviour will see the script and checks if the current image target is within a certain set of bounds and if it isn't it sends a message to the robot to move for one frame before stopping.



The mapping scene is made sure to include the computer's IP address on the SendMessageBehavior.cs script. This script sends the name, position, rotation, and scale of the generated planes back to your computer as ARkit instantiates them. This gets received from the MapController scene where all the planes will be displayed on the computer.

Mapping scene



Follow scene

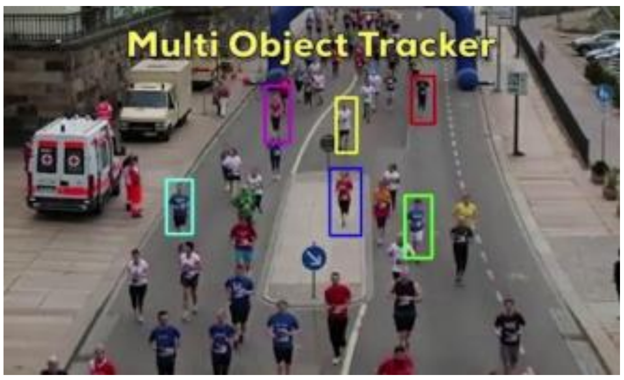
Now the last scene uses Apple's ARkit which detects vertical and horizontal planes (which Vuforia does support) so it will only work on IOS.

V. CONCLUSION :

The implementation of augmented reality in consumer products requires considering the design of the applications and the related constraints of the technology platform. Since AR systems rely heavily on the immersion of the user and the interaction between the user and the system, design can facilitate the adoption of virtuality. For most augmented reality systems, a similar design guideline can be followed. We have introduced a system which included mobile robots and a host computer.This project uses an Arduino robot running a ESP-8266 Node MCU that communicates (and drives around) with a mobile phone. The phone is running an app made in the video game engine Unity 3D

Scope :

- The Follow scene is used in automated vehicles as an adaptive cruise controlling feature.
- The Follow scene is used in real time tracking of objects like balls in cricket and tennis broadcasting coverage.
- The Follow scene can be used in track trajectories and can be enhanced using machine learning and deep learning algorithms .
- The Mapping scene can be used in industrial automated equipment carriers
- The Mapping scene can be used in obstruction avoidance and advanced 3D mapping.



FUTURE WORK:

The development features in brief:

- Room mapping.
- Multi Object tracker.

REFERENCES:

[1]. N. Birkbeck and M. Jagersand, **"Visual tracking using active appearance models,"** First Canadian Conference on Computer and Robot Vision, 2009. Proceedings., London, ON, Canada, 2004, pp. 2-9, doi: 10.1109/CCCRV.2004.1301414.

[2]. Z. He, M. Shi and C. Li, **"Research and application of path-finding algorithm based on unity 3D,"** 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, 2016, pp. 1-4, doi: 10.1109/ICIS.2016.7550934.

[3]. S. Patil, A. Wagh, M. Sawant, S. Panda and A. Bhopale, "**Design and implementation of advanced auto calibrating line following sensor for coloured surfaces with a white line,**" 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, 2016, pp. 1-6, doi: 10.1109/ICPEICES.2016.7853535.

[4]. Y. Kuang and X. Bai, **"The Research of Virtual Reality Scene Modelling Based on Unity 3D,"** 2018 13th International Conference on Computer Science & Education (ICCSE), Colombo, 2018, pp. 1-3, doi: 10.1109/ICCSE.2018.8468687.

[5]. N. Mir-Nasiri, **"Camera-based 3D Object Tracking and Following Mobile Robot,"** 2016 IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, 2006, pp. 1-6, doi: 10.1109/RAMECH.2006.252655.