

AR APPLICATION ON MULTIPLE OBJECT
PLACEMENT AND MANIPULATION

DATE-18TH SEPT TO 29TH DEC

OINDRILA PAUL

BTECH VELLORE INSTITUTE OF TECHNOLOGY
(VIT-VELLORE)

COMPUTER SCIENCE CORE

MENTOR :SOUMYADIP MAITY

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to TCS Research and Innovation for providing me this opportunity to undertake my internship.

The experience I gathered here during my tenure is invaluable in shaping my professional and enhancing my skills.

I extend my heartfelt gratitude to my mentor Mr Soumyadip Maity for his guidance and mentorship throughout my internship period.

I also am grateful for the guidance I have received from Mr Abhijan Bhattacharyya .

I have learnt a lot from my interactions with everyone during my internship and enhanced my professional skills.

I am extremely thankful to my friends and family for their support throughout.

Thank you to everyone who played a role in making this internship possible and a great experience.

ABSTRACT

This Unity-based mixed reality application immerses users in an interactive experience, seamlessly integrating virtual objects into their physical environment. Employing a user-friendly interface, it enables dynamic manipulation, precise positioning, and real-time updates on object dimensions and distance from the camera. The application also in

KEYWORDS

Unity, mixed reality, AR environment, user interaction, virtual objects, real-time feedback, object manipulation, spatial understanding, AR enhancement, practical applications.

INTRODUCTION

This innovative mixed reality application, developed within Unity's AR environment, introduces an immersive experience allowing users to interact seamlessly with virtual objects in their physical space. At its core, the application harnesses a user-friendly interface featuring dropdown menus to store a diverse range of objects and buttons facilitating easy access to these items.

Users can dynamically manipulate, resize, and precisely position these objects using intuitive touch gestures within the augmented environment. Furthermore, the application provides real-time updates on the dimensions of the placed objects and their distance from the user's viewpoint, enhancing the user's spatial understanding and interaction with the virtual elements.

Through this comprehensive suite of functionalities, encompassing dropdown-based object selection, intuitive button controls, object manipulation, dimension and distance display, the application aims to deliver a deeply engaging mixed reality experience. By seamlessly integrating virtual objects into the physical world, this application facilitates an interactive environment that merges technology with user experience for both entertainment and potential practical applications across various industries.

AIM

Develop an immersive and user-friendly mixed reality application using Unity for Augmented Reality (AR) environments. The primary objective is to create an interactive experience that allows users to

seamlessly interact with virtual objects within their physical surroundings, providing functionalities for object manipulation, information display, and user-controlled actions.

OBJECTIVES

1. User Interaction:

- Enable users to intuitively interact with the AR application through touch gestures, including tapping, dragging, and pinch-to-scale for object manipulation.

2. Dynamic Object Placement:

- Implement a system to allow users to place predefined 3D objects onto real-world surfaces detected by AR technology.

3. Object Manipulation:

- Provide users with the ability to resize placed objects using pinch gestures or other intuitive methods.

4. Real-time Information Display:

- Display accurate dimensions and distance information of placed objects from the user's perspective in real-time.

5. User Interface Enhancement:

- Develop a user-friendly interface, including interactive buttons, dropdown menus, or other UI elements for seamless navigation and control.

6. Spatial Understanding and Interaction:

- Utilize raycasting and spatial awareness to enable realistic interactions with virtual objects based on their placement and orientation in the physical environment.

7. AR Environment Enhancement:

- Enhance the AR experience by providing features like object highlighting upon interaction, visual feedback, or spatial audio cues to improve user engagement.

8. Educational/Practical Use Case:

- Implement the application with a specific use case in mind, such as education, design visualization, or practical applications to cater to a particular audience or industry.

METHODOLOGY

Various steps which have been followed to reach the final project.

Plane Detection:

Detects flat surfaces in the real world, crucial for accurately placing digital objects. Utilizes Unity 3D packages like AR Foundation and XR Plugin Manager, configures the AR Session Origin, and employs the AR Plane Manager for surface detection.



Tap to Place & Resize Single Object:

Allows precise object placement and resizing within the AR environment. Involves importing necessary packages, configuring AR Session Origin, implementing a Raycast Script for object placement and resizing, and displaying object dimensions.



Placing Multiple Objects by Tapping:

Enables users to tap and place multiple digital objects on a detected plane in AR. Involves integrating the AR Default Plane, incorporating ARRaycast Place Component, crafting 3D object prefabs, and activating touch input for object placement.

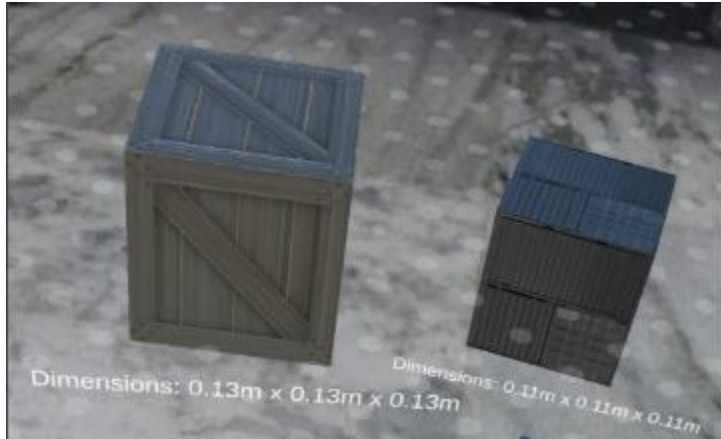


Precise Object Placement & Displaying Dimensions:

Facilitates tapping to place a single object precisely, resizing it via pinching, and displaying its dimensions for accurate placement. Utilizes AR Default Plane Detection, Raycast Script implementation for dimension calculation and displaying dimensions in a text box.

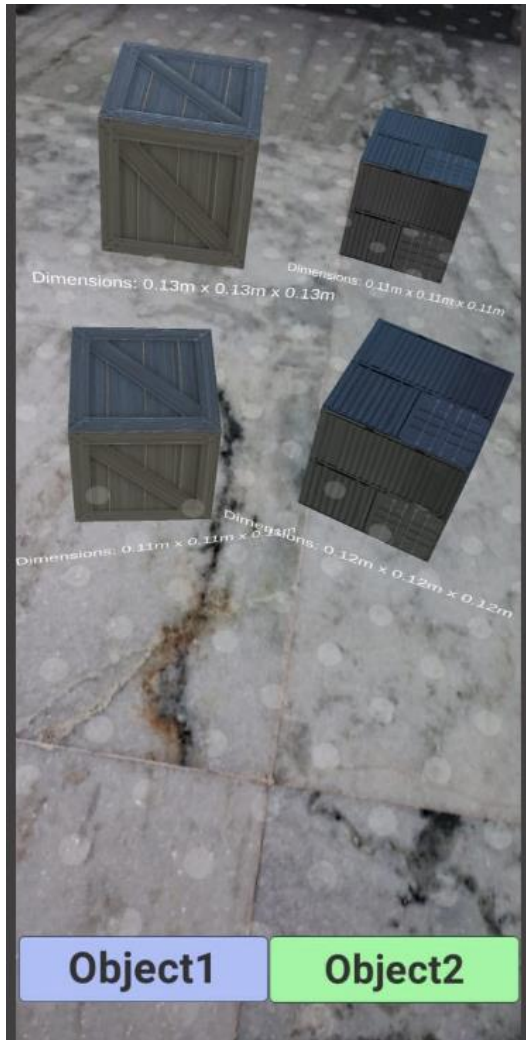
Placing & Resizing Multiple Objects Individually:

Enables users to place and resize multiple objects one by one. Involves AR Default Plane setup, script integration for individual resizing and placement, setting timers for object placement, and implementing pinch gestures for resizing.



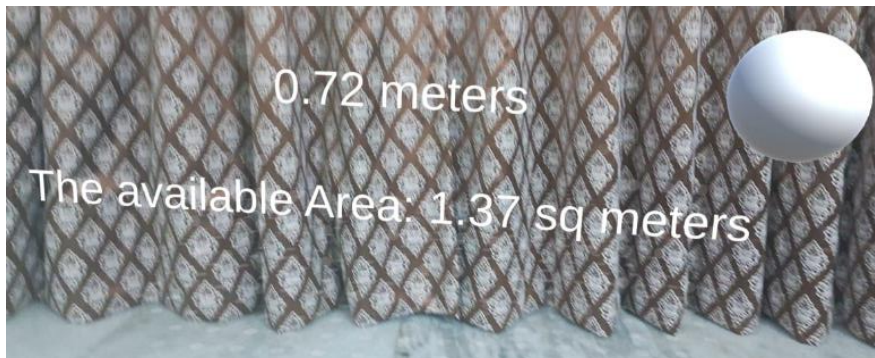
Selecting & Placing Different Objects with Buttons(UI):

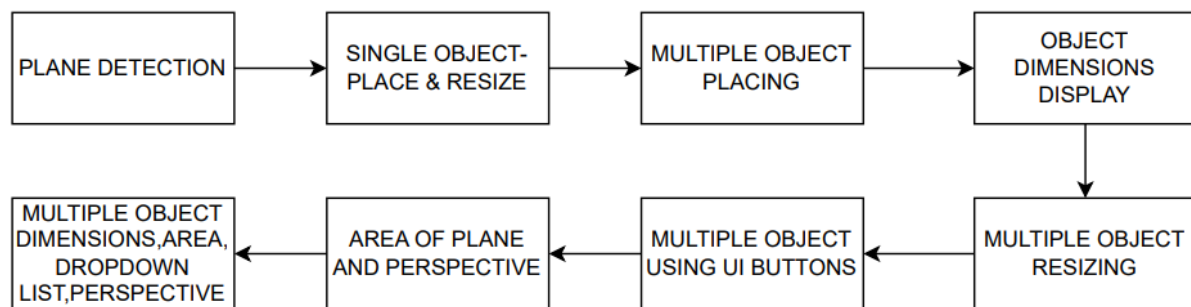
Allows users to select different objects using buttons, place them with a tap, and customize their sizes. Involves AR Default Plane integration, setting up button prefabs, creating a Button Manager Script, and assigning object prefabs to buttons for placement.



Area of plane detected and distance from camera (perspective)

Users can get an idea of the available plane area and the distance of a point from the camera and understand the perspective from the camera.





Final Project

For the final project AR Session Origin and 3D Object (Dimension manager) is used. The dropdown list is constructed out of the existing 3D objects like cubes.

Boxes, objects and material for the objects has been downloaded from the Unity asset store and imported into the application.

The following scripts have been used.

ButtonManager: This script manages button interactions in a Unity project. It's responsible for instantiating objects upon button clicks with a delay, as well as adjusting the button sizes when hovered over. It's linked to another script that handles the placement of objects.

CubeButton: Attached to buttons, this script handles the delayed instantiation of objects upon clicking. It also manages the button's visual behaviour by adjusting its size on hover and maintaining its original scale after being clicked.

DimensionManager: In a Unity scene, this script manages the display of an object's dimensions and distance from the camera. It calculates these values on mouse clicks and updates accordingly.

DropdownScript: This script creates a dropdown-like behavior in Unity. It toggles the visibility of options based on mouse clicks, effectively creating a dropdown list of options.

PinchScaleObject: This script allows users to scale objects in the scene using a pinch gesture on touch devices. It dynamically updates the dimensions of the scaled object and its distance from the camera.

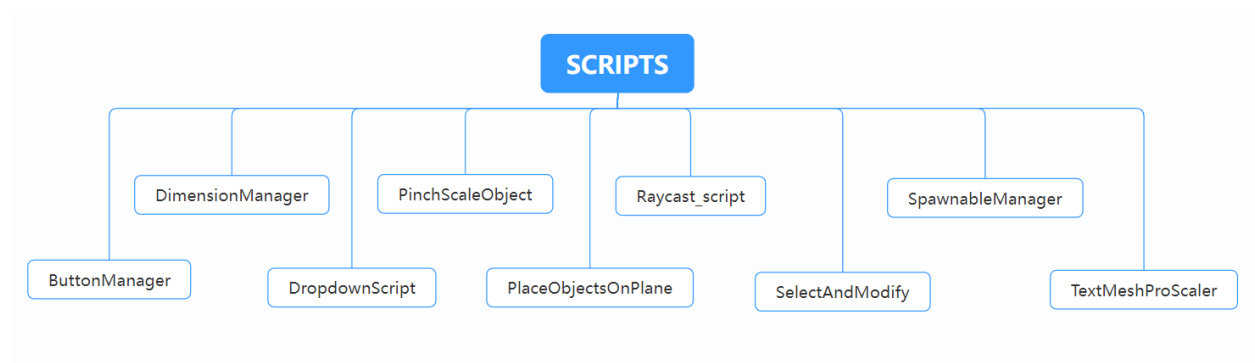
PlaceObjectsOnPlane: This script is responsible for placing objects onto an AR plane in Unity. It handles the selection of object types and delays the placement of objects.

Raycast_script: This script continuously updates object dimensions and the distance of the object from the camera within the Unity scene.

SelectAndModify: Facilitating object scaling using a two-finger touch gesture, this script updates the object's dimensions and its distance from the camera.

SpawnableManager: Using AR raycasting, this script spawns objects on a plane, calculates the area of the plane, and updates a TextMeshPro textbox accordingly.

TextMeshProScaler: A script adjusting the TextMeshPro text size based on the scale of the parent object within a Unity scene. It dynamically scales the text to match the object's size.



The scripts that have been used are assigned to each object as needed.

The 3d objects(which are the ones that are being placed) have been made to have two textmeshpro objects to display the necessary data.

There has been a textmeshpro added to the scene permanantly to display the area of the plane detected.

The two textboxes attached with the objects will also get scaled and changed in size according to the object.

The area detected by the plane will always get displayed in the permanent textbox and be updated as according to the area of the plane detected.

DISCUSSION

1. Create Scene & Import Packages:

- Created a new scene in Unity.
- Imported necessary packages, such as TextMeshPro, AR Foundation, AR Core and platform switched to Android/IOS as according to the device we are using.

2. Scripts Added:

- Added various scripts to the Unity project
- ButtonManager, CubeButton, DimensionManager, DropdownScript, PinchScaleObject, PlaceObjectsOnPlane, Raycast_script, SelectAndModify, SpawnableManager, TextMeshProScaler.

3. Dropdown Created:

- Implemented a dropdown-like behavior using the DropdownScript
- the dropdown has been constructed from the inbuilt 3d objects like cubes
- Linked the dropdown to specific objects (the cubes which have been modified in shape).

4. Dimension Manager Added:

- an empty 3D object has been created which is named as DimensionManager
- add the DimensionManager script to the DimensionManager inspector
- Implemented functionality to calculate and display object dimensions and distance from the camera on mouse clicks.

5. TextMeshPro Elements Added:

- Utilized TextMeshPro elements within the scene.
- One TextMeshPro added to the scene to display the area
- two TextMeshPro added to the objects to be placed on touch
- Implemented TextMeshProScaler to dynamically scale text based on the size of parent objects.

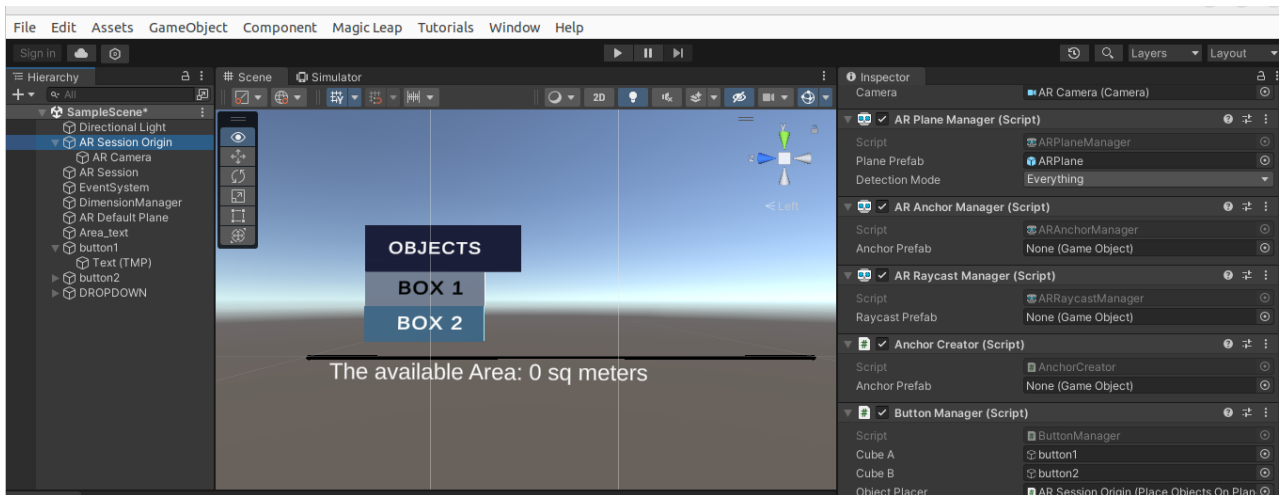
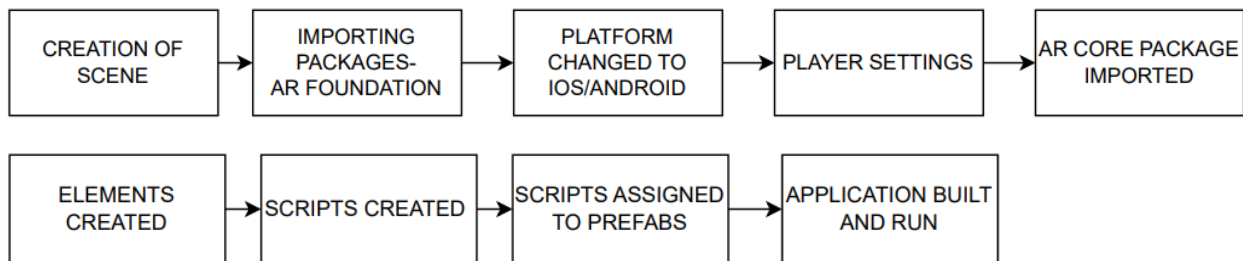
6. All Scripts Assigned:

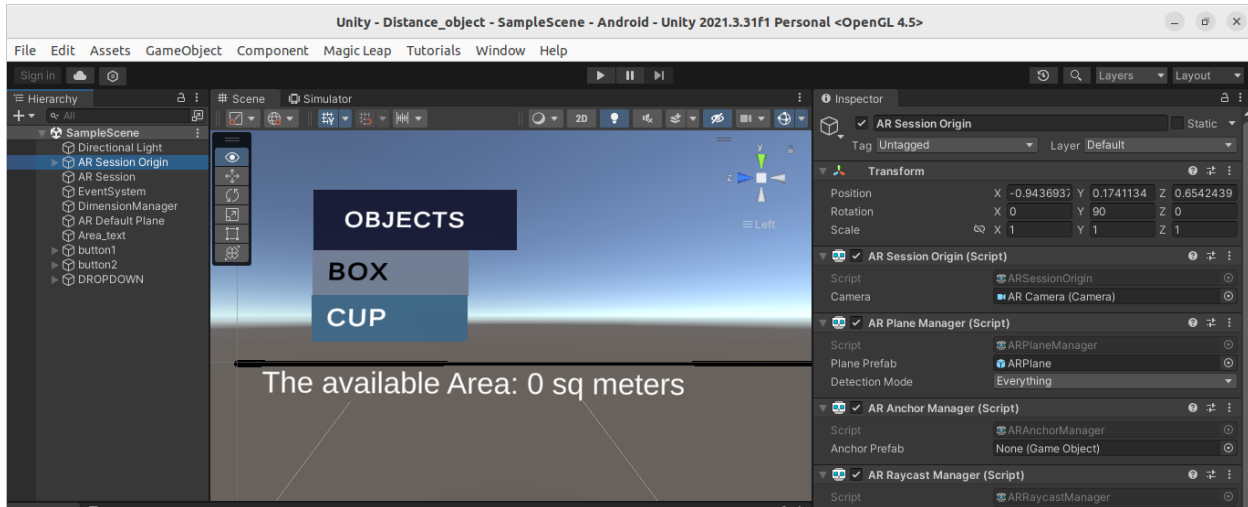
- Linked and assigned relevant scripts to GameObjects and UI elements within the Unity scene.

- Interconnected functionalities across various scripts, facilitating object placement, manipulation, UI interactions, and dimensional calculations.
- The scripts should be loaded and assigned to the scene and the prefabs for all the elements to be working properly

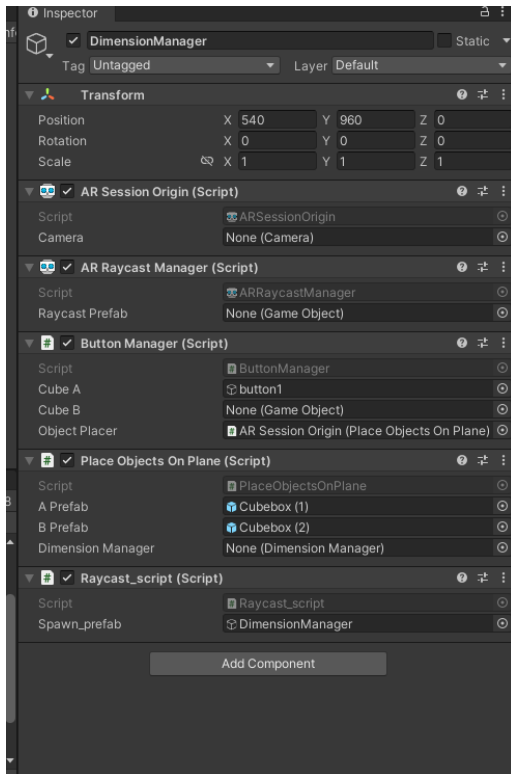
7. Application Build and Run:

- save the scene and then we have to change the platform to Android/iOS depending on the device we are using
- then after going to player settings under XR plugins we have to import the ARCore package (this ensures that our camera is used in the device)
- then we have to compile and build the Unity application.
- Executed the application, allowing for interaction and functionality based on the implemented scripts and configurations.

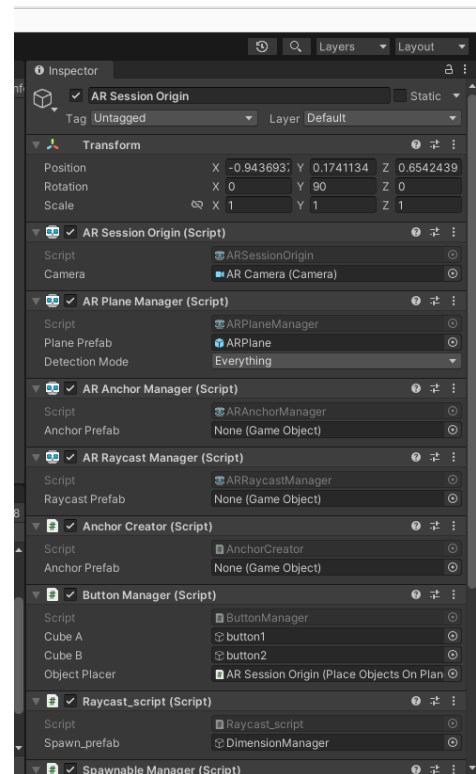




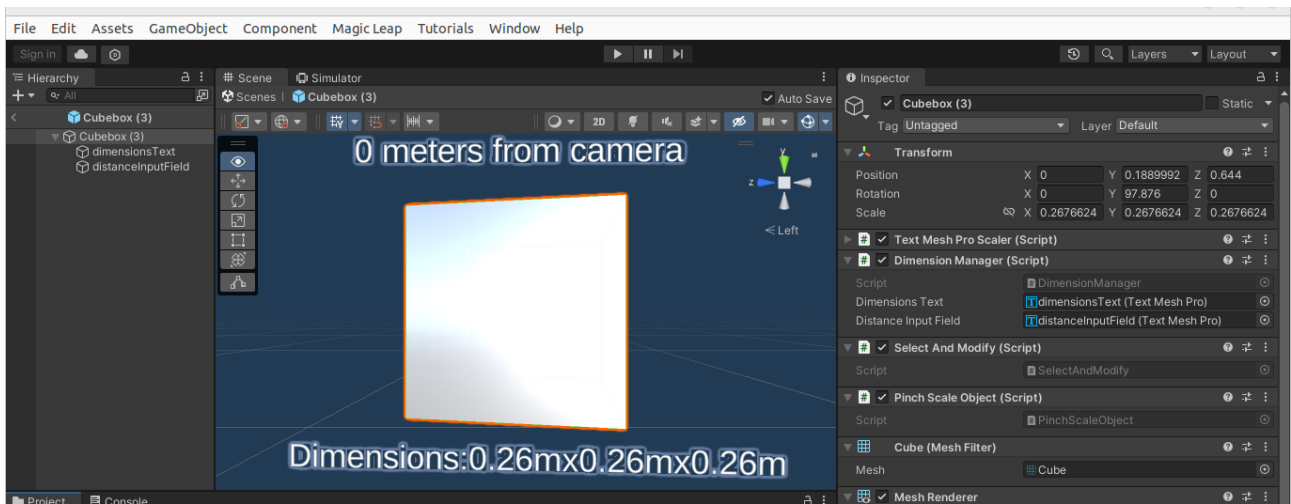
SAMPLE SCENE



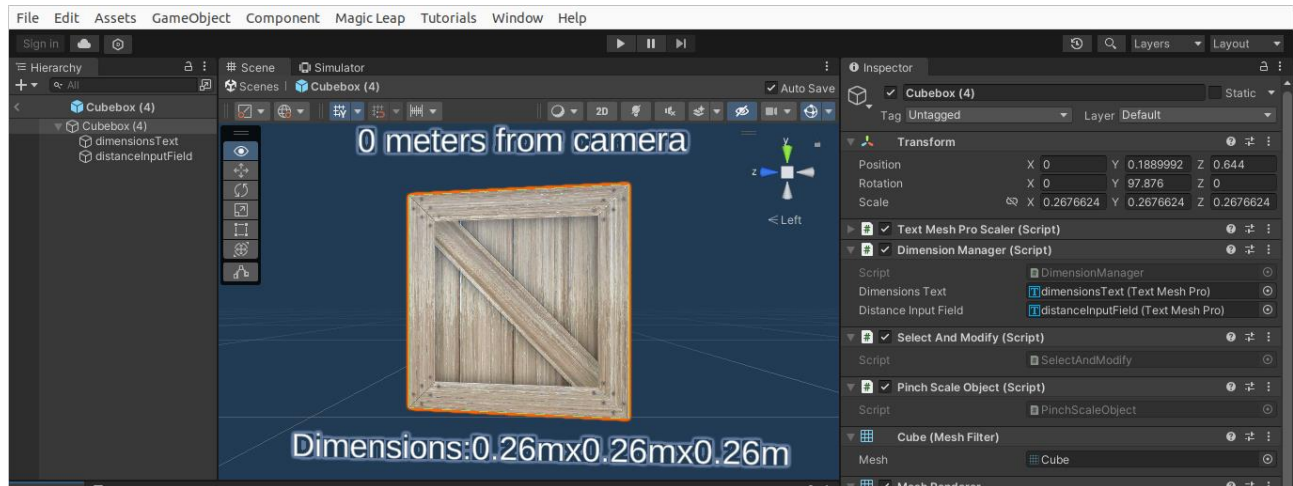
SCRIPTS
ASSIGNED



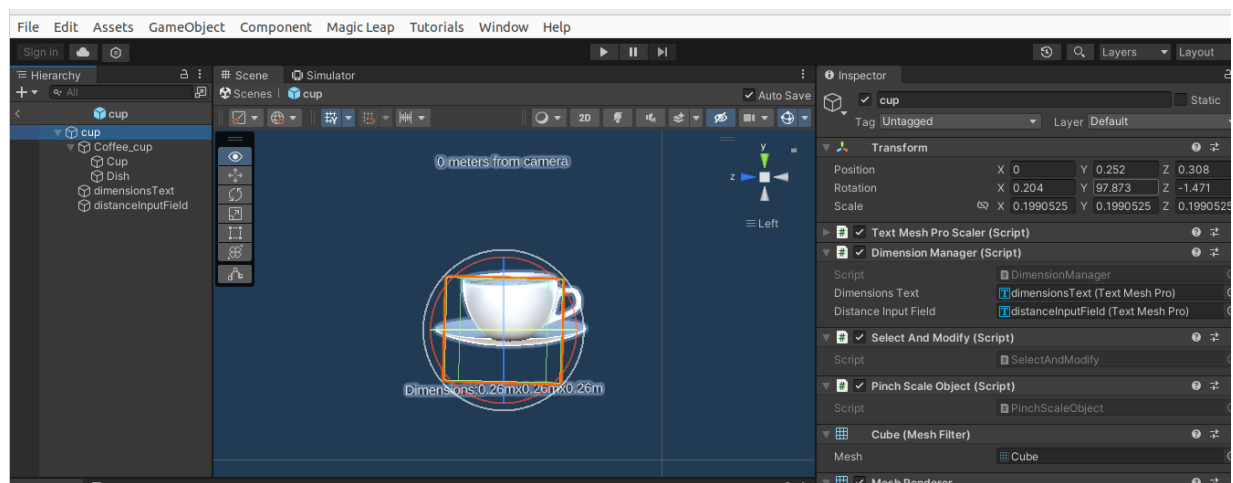
OBJECT 1



OBJECT 2



OBJECT 3





RESOURCES USED

The sites like Unity Discussions, Unity

Forum, GamedevelopmentStack, Medium, StackOverflow have been used to refer articles, code and steps.

USE CASES

1. Education:

- **Interactive Learning:** The application can be used in educational settings to create interactive learning experiences. For example, students studying anatomy could place 3D models of human organs in their physical space for a more immersive and hands-on learning experience.

2. Design and Architecture:

- **Spatial Visualization:** Architects and designers can use the application to visualize and manipulate 3D models of buildings or interior designs in real-world settings. This can aid in better spatial understanding and decision-making during the design process.

3. Manufacturing and Prototyping:

- **Product Prototyping:** Engineers and product designers can use the AR application for prototyping and visualizing products in a real-world environment before physical prototypes are created. This can streamline the design iteration process.

4. Medical Training:

- Surgical Planning: Surgeons and medical professionals can utilize the application for surgical planning, allowing them to place virtual medical equipment or visualize anatomical structures in a 3D space.

5. Entertainment and Gaming:

- Immersive Gaming: The application can enhance gaming experiences by allowing users to interact with virtual objects in their physical space. Gamers could, for example, place virtual characters or objects within their living room for a more engaging gameplay experience

EXAMPLE

Amazon online shopping offers a feature to check whether the item we are buying is going to fit in our expected place.

If we are buying a sofa we can use the AR application feature introduced to check whether the sofa is going to fit or not.

Even a 75" tv we can check whether it will fit on our wall or not.

CONCLUSION

The creation of this mixed reality application within Unity's AR environment represents a significant leap forward in blending virtual elements with real-world surroundings. The comprehensive suite of functionalities, including dropdown-based object selection, dynamic placement, and spatial interactions, contributes to a deeply engaging mixed reality encounter.

The project's ultimate goal was to create an immersive, user-friendly experience, seamlessly merging virtual objects into the physical world. The incorporation of features like precise object placement, manipulation through touch gestures, and real-time dimensional updates enhanced user spatial understanding and interaction. Moreover, the application's potential spans across various industries, offering avenues for educational, entertainment, and practical applications in design visualization or other specialized fields.

Through meticulous methodology encompassing plane detection, multiple object handling, and robust script utilization, the project achieved its objectives of delivering a compelling mixed reality experience. By strategically implementing scripts, configuring AR functionalities, and ensuring seamless integration of TextMeshPro elements, the project set the stage for a refined user journey.

This culmination of efforts underscores the potential of mixed reality applications, showcasing how technology can seamlessly fuse with real-world environments. The project stands as a testament to the possibilities within the realm of mixed reality, offering a glimpse into a future where virtual elements seamlessly coexist and enhance our physical surroundings for diverse applications across industries and user experiences.