

**School of Engineering and Technology**

**BCA**

**TY / 5th Semester**

**AR & VR Technologies**

**Moksha (22000254)**

**Course In-charge**

**Darshan Sir**

**Navrachana University, Vadodara**

**Autumn Semester 2023-24**

**VR Target Arena**

Table of Contents -

1. Introduction

2. Project Setup

3. Step-by-Step Process

* Setting Up the VR Environment
* Creating the Ground Plane
* Adding a Skybox
* Adding Environment Objects
* Configuring Lighting and Shadows
* Adding Audio
* Implementing Basic VR Interaction
* Writing the VR Interaction Script

4. Demo Application

5. Challenges Face

6. References

**INTRODUCTION :**

The Target Arena is an immersive VR experience designed to help users improve their shooting accuracy and precision in a virtual environment. With a range of customizable weapons and targets, it allows players to practice in a safe, controlled space that feels incredibly realistic. Whether you're a beginner looking to sharpen your skills or a seasoned shooter aiming for perfection, the Target Arena provides challenges suited for all levels. The game offers instant feedback, making it easy to track your progress and refine your technique. Overall, it's an engaging and interactive way to test your marksmanship.

**Project Setup :**

* Unity Version: 2020.3.25f1
* Plugins/Packages:
  + XR Interaction Toolkit - Installed via Unity Package Manager
  + XR Plugin Management - Installed via Unity Package Manager
  + Steam VR Plugin

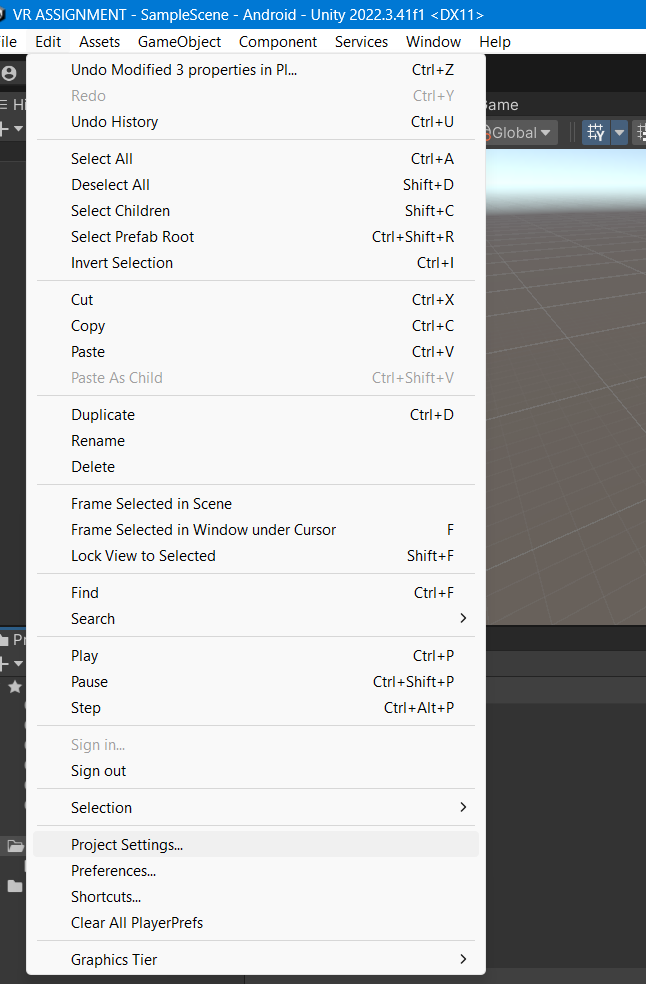
**Step-by-Step Process :**

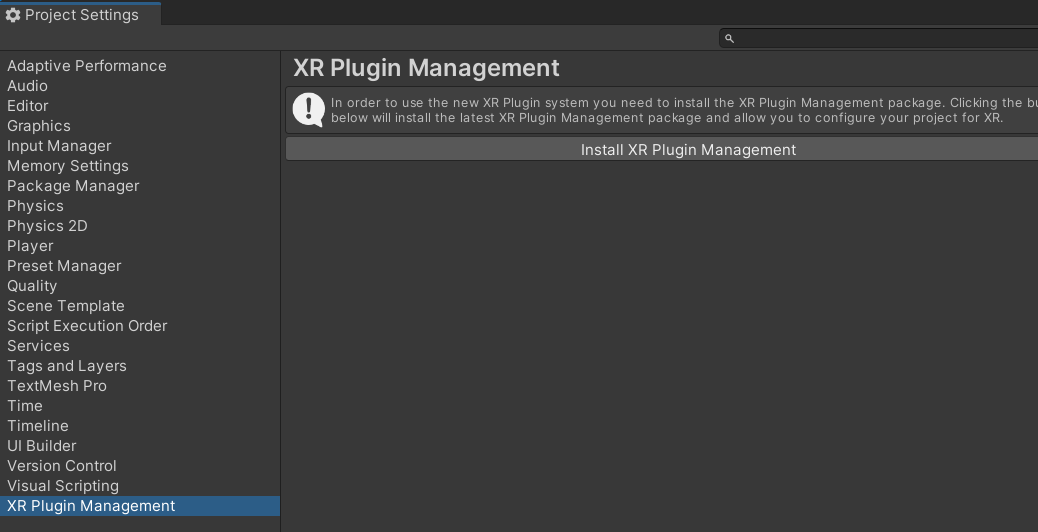
* Task 1: Setting Up the VR Environment

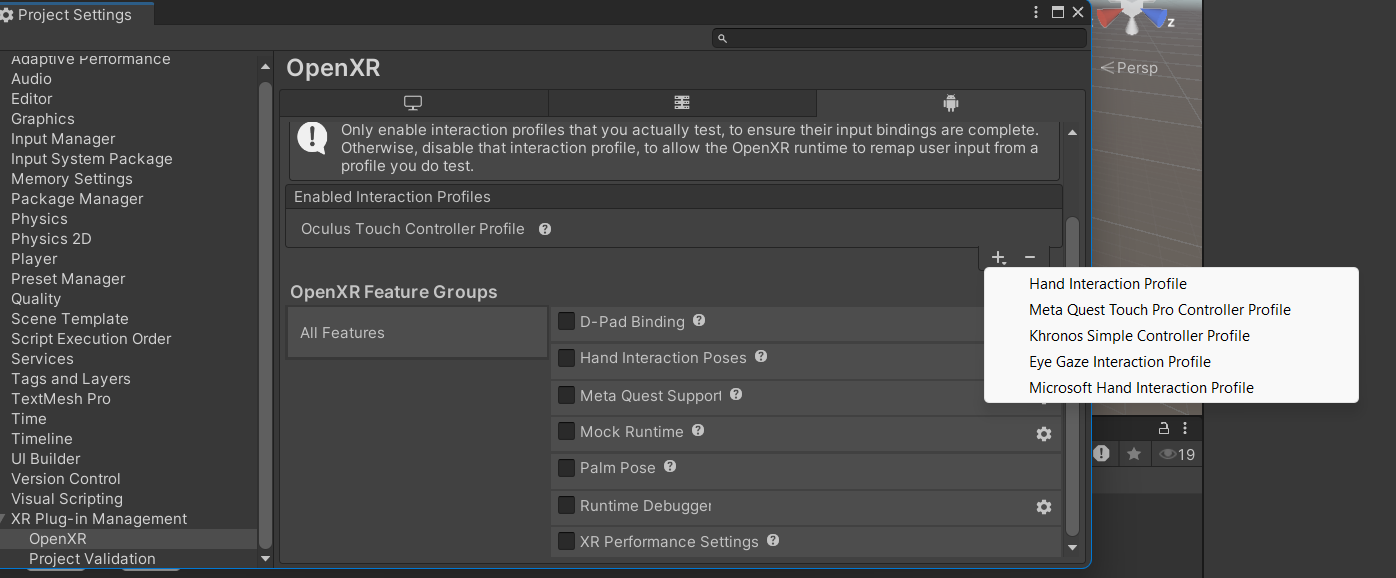
- Created a new Unity project and installed the XR Interaction Toolkit.

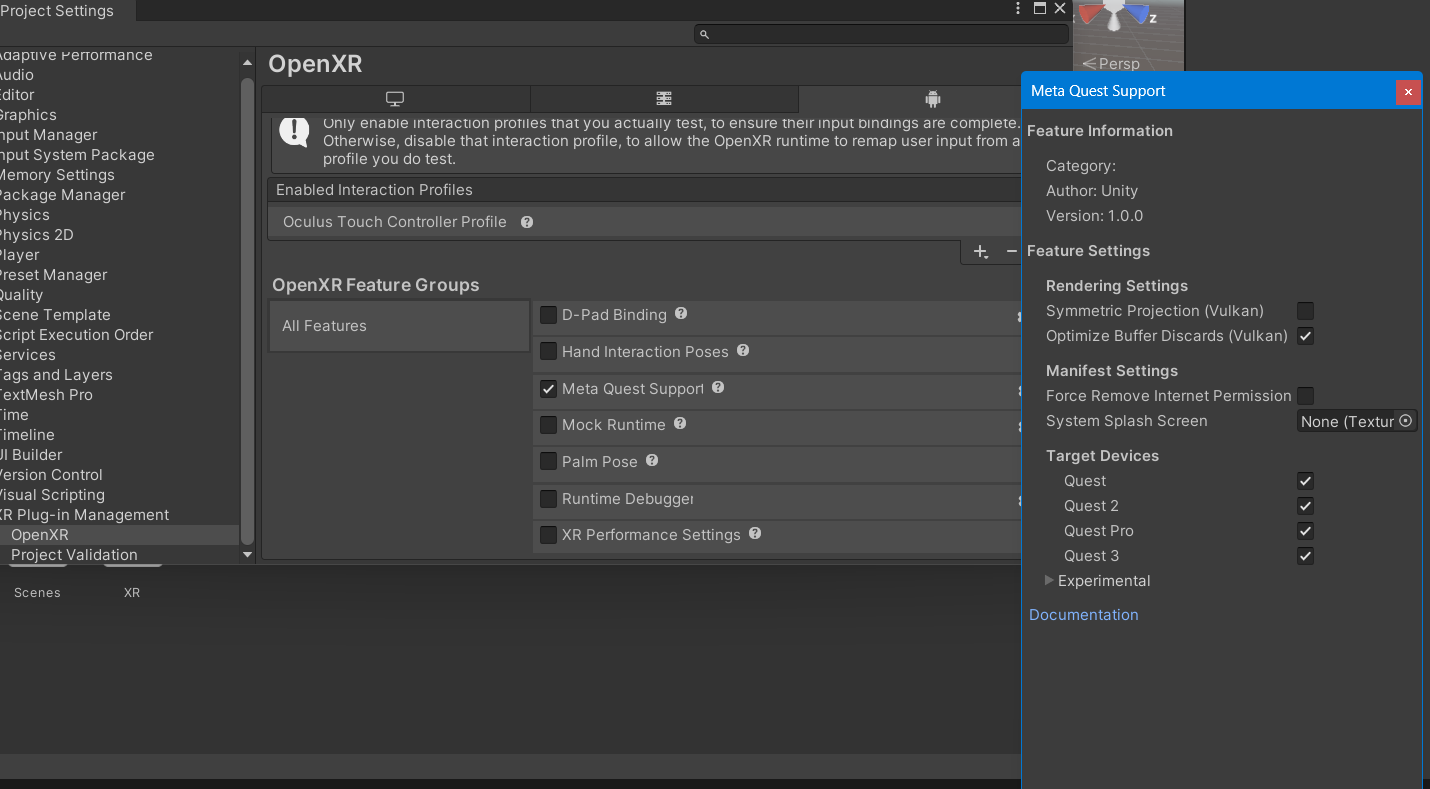
- Added the XR Rig to the scene to facilitate VR interactions.

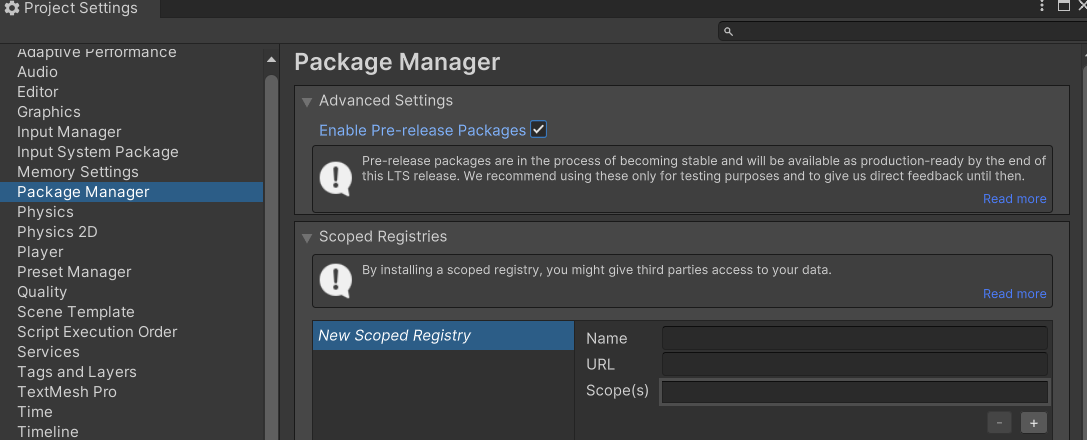
**Images:**

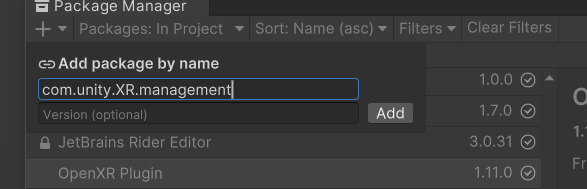


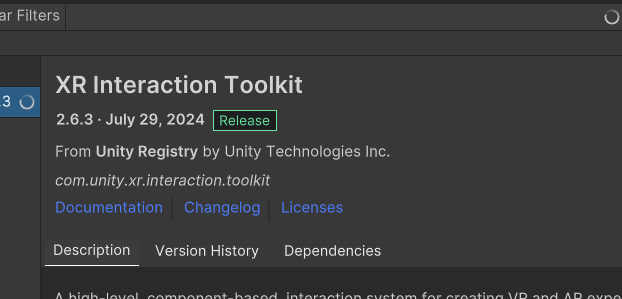


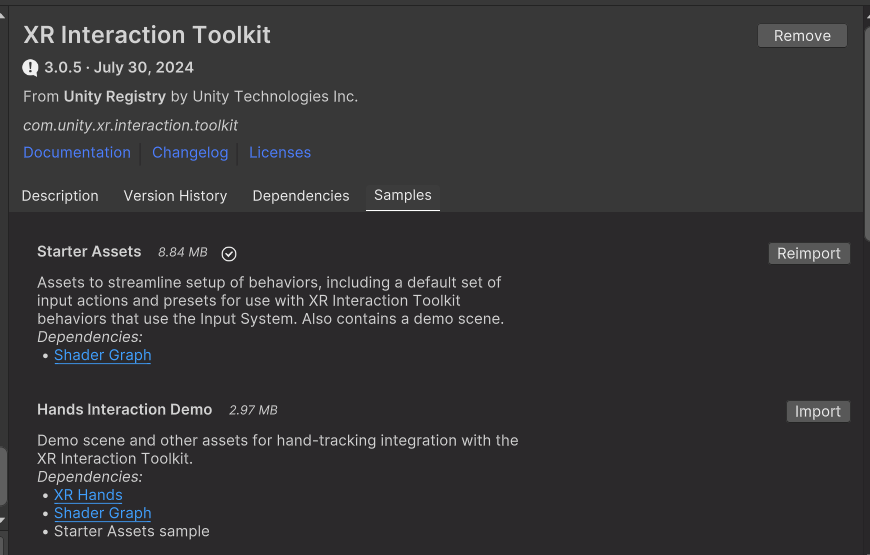










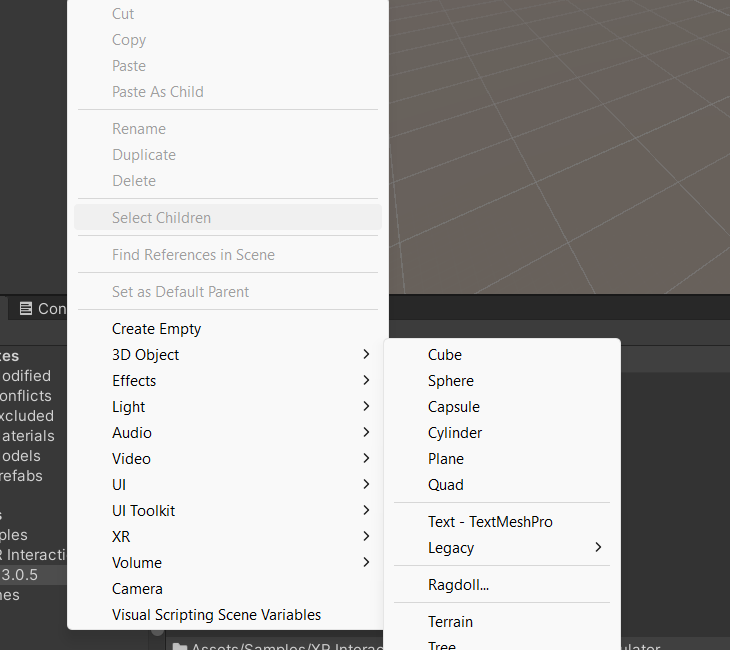


* Task 2: Creating a Ground Plane

- Used GameObject > 3D Object > Terrain to create the ground plane.

- Adjusted the terrain size to allow the player to move around.

**Images:**

****

* Task 4: Adding Environment Objects

Adding the Target Arena from Unity Asset Store

Step 1: Open the Unity Asset Store

In Unity, open the Asset Store by navigating to Window > Asset Store or by directly visiting the Unity Asset Store in your web browser.

Step 2: Search for "Target Arena"

In the search bar of the Asset Store, type "Target Arena" to find the asset.

Look through the results and select the Target Arena asset that best fits the project.

Step 3: Download and Import the "Target Arena" Asset

Once you’ve located the Target Arena asset, click on Download (if it’s free) or Purchase (if it’s a paid asset).

After the download is complete, click Import to add the asset to your project.

Unity will show a list of files to import. Make sure to import all necessary assets like models, materials, prefabs, and textures by leaving everything selected.

Step 4: Add the "Target Arena" to Your Scene

Once the asset is imported, go to the Project window and navigate to the folder where the Target Arena asset is stored (usually under Assets/Target Arena or a similar folder name).

Look for a prefab or scene related to the Target Arena.

Drag the Target Arena prefab into your Hierarchy window or directly into your active scene view.

Step 5: Adjust the "Target Arena" Environment

Positioning: Adjust the position, rotation, or scale of the Target Arena to fit your scene layout using the Transform tool.

Ensure that the arena is properly aligned with the Main Camera or the XR Rig so the player can interact with the targets.

Lighting Adjustments: If the lighting in the scene does not complement the Target Arena, you may need to adjust the lighting.

You can add a Directional Light or modify the existing light sources by going to GameObject > Light in the Unity menu.

Step 6: Test the Scene with "Target Arena"

Press the Play button in Unity to test how the Target Arena looks and functions in the VR environment.

Make sure to check the interaction between the player and the targets. Ensure the environment allows enough space for shooting and navigation.

Step 7: Save the Scene

Once you’re happy with the layout and functionality of the Target Arena, save the scene:

Go to File > Save Scene and ensure all your changes are saved.

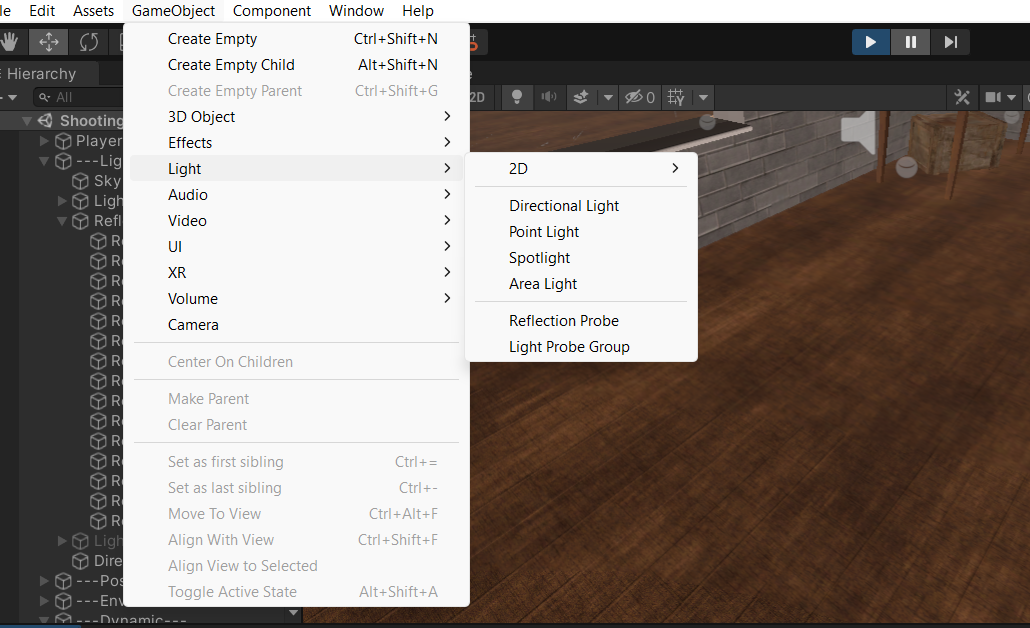
* Task 6: Configuring Lighting and Shadows

Adding Lighting and Shadows to the Target Arena

Step 1: Add a Directional Light

Go to GameObject > Light > Directional Light to simulate sunlight.

Adjust the light’s rotation to cast shadows in the desired direction.



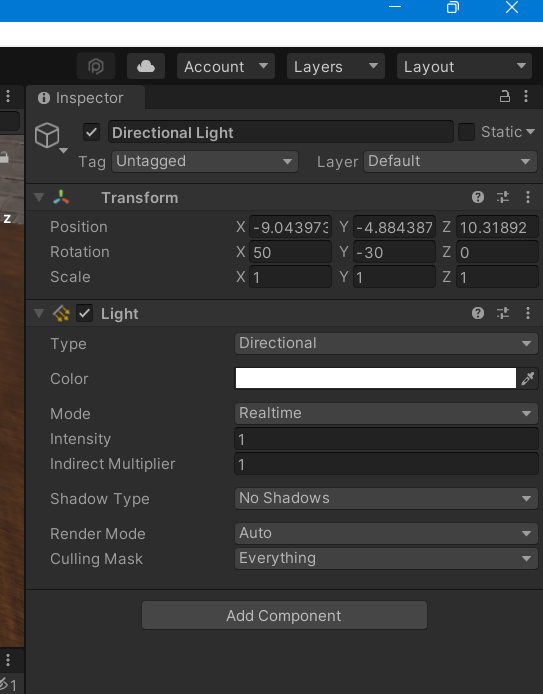
Step 2: Configure Light Settings

Select the Directional Light and adjust the following in the Inspector:

Color: Choose a suitable color for your scene.

Intensity: Set between 0.5 to 1.5.

Shadows: Set Shadow Type to Soft Shadows and adjust Shadow Strength (0.8 to 1.0).



Step 3: Enable Shadows on Objects

Select key objects in the Target Arena.

In the Inspector, under Mesh Renderer, set Cast Shadows to On.

Step 4: Test the Scene

Press the Play button to see how the lighting and shadows enhance the environment

* Task 7: Adding Audio

Adding Audio to the Target Arena

Step 1: Import Audio Files

Drag and drop your audio files (.wav or .mp3) into the Assets folder in Unity.

Step 2: Add an Audio Source

Select the Target Arena or main camera in the Hierarchy.

Go to Component > Audio > Audio Source.

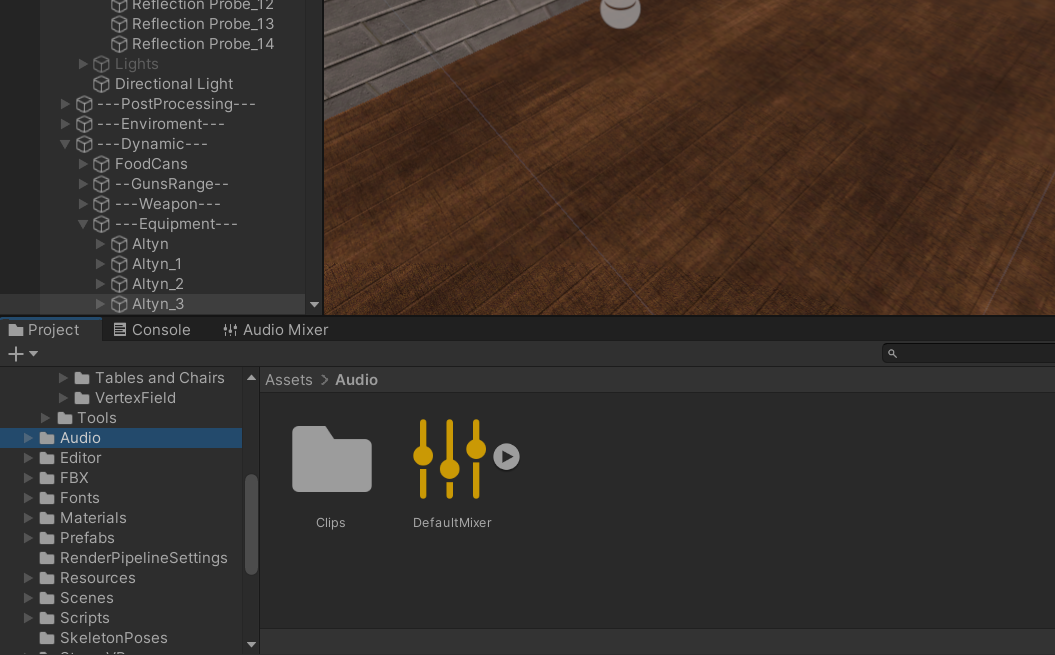
Step 3: Configure the Audio Source

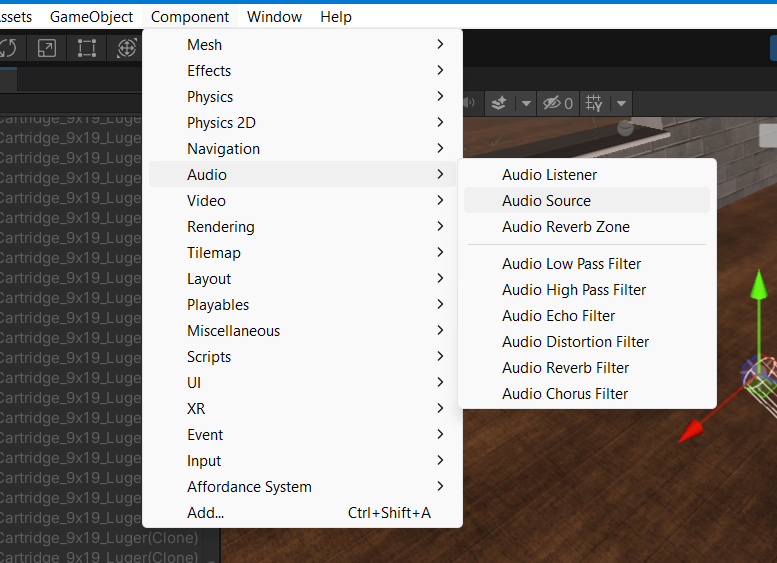
In the Inspector, set the Audio Clip to your imported audio file.

Check Play On Awake and Loop (if needed).

Step 4: Test the Audio

Press Play to check if the audio plays correctly in the VR environment.



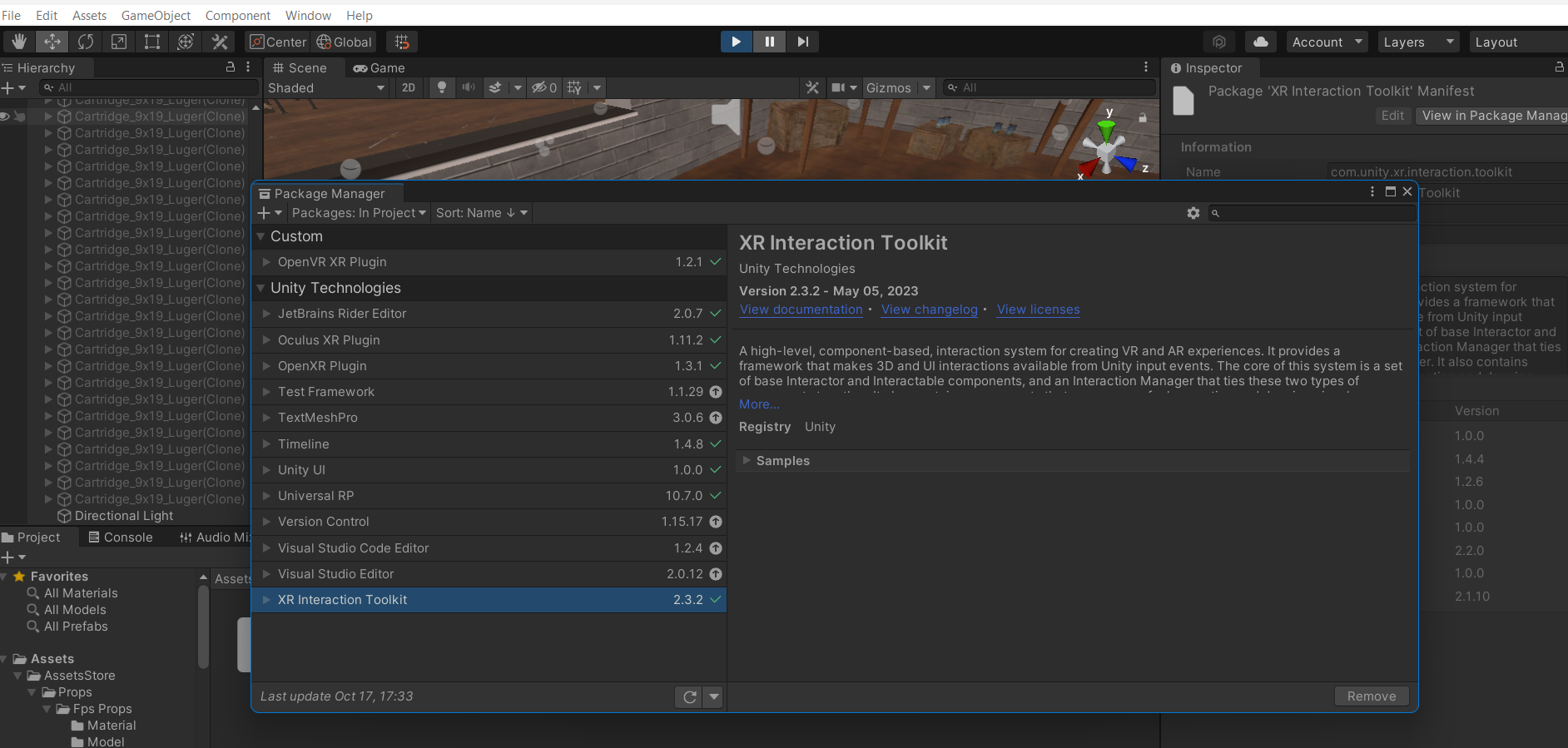


* Task 8: Implementing Basic VR Interaction

Implementing XR Grab Interactable and XR Ray Interactor

Step 1: Install XR Interaction Toolkit

Ensure the XR Interaction Toolkit is installed via Window > Package Manager.



Step 2: Set Up XR Ray Interactor

Select the XR Rig in your scene.

Add XR Ray Interactor to the controller (e.g., left or right):

Add Component > XR > XR Ray Interactor.

Step 3: Add XR Grab Interactable

Select the object you want to grab (e.g., weapon or target).

Add XR Grab Interactable:

Add Component > XR > XR Grab Interactable.

* Task 9: Writing the VR Interaction Script

Code:

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

public class VRGrabInteraction : MonoBehaviour

{

public XRGrabInteractable grabObject;

private void Start()

{

grabObject.onActivate.AddListener(Grab);

grabObject.onDeactivate.AddListener(Release);

}

private void Grab(XRBaseInteractor interactor)

{

// Code for grabbing the object

Debug.Log("Object grabbed");

}

private void Release(XRBaseInteractor interactor)

{

// Code for releasing the object

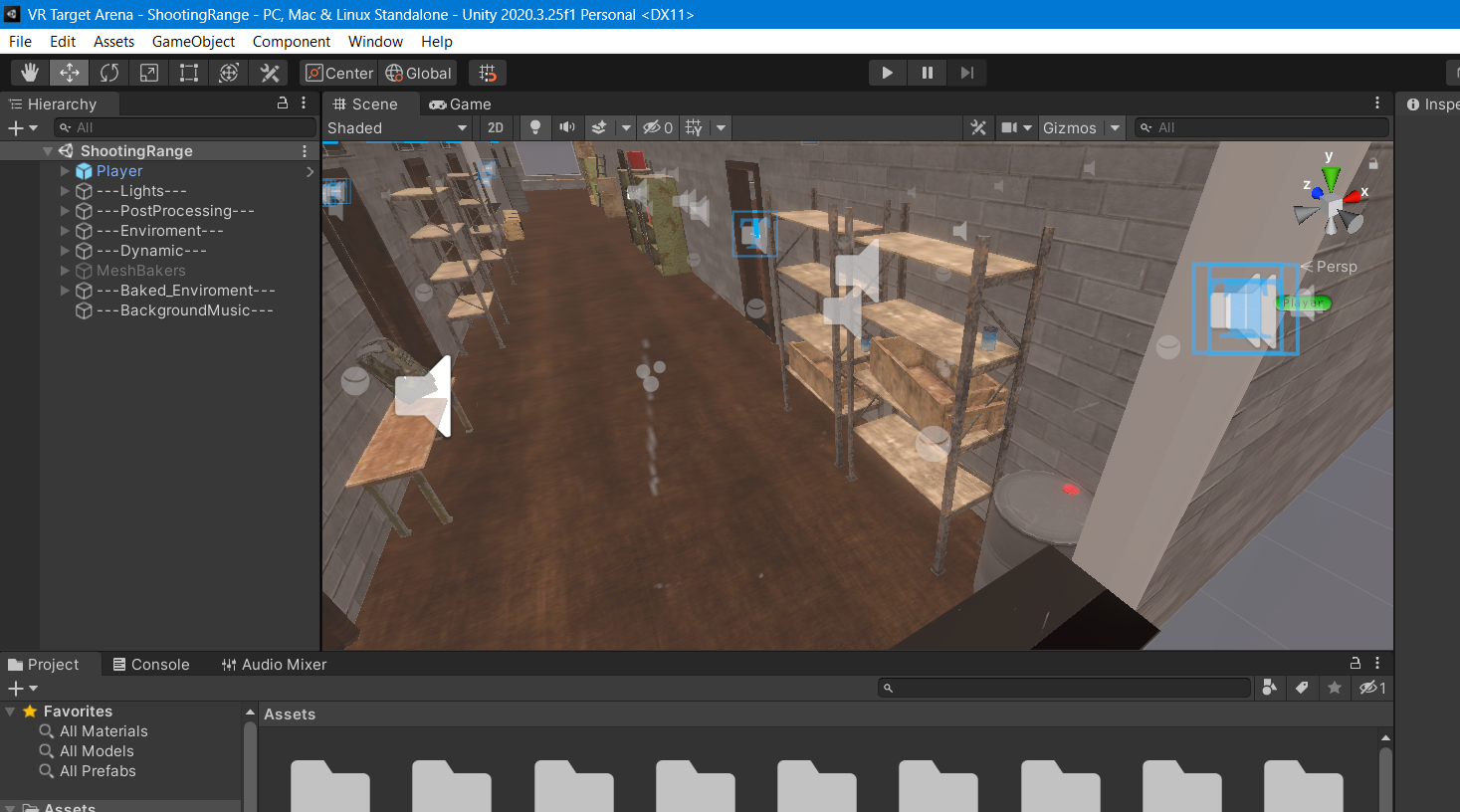
Debug.Log("Object released");

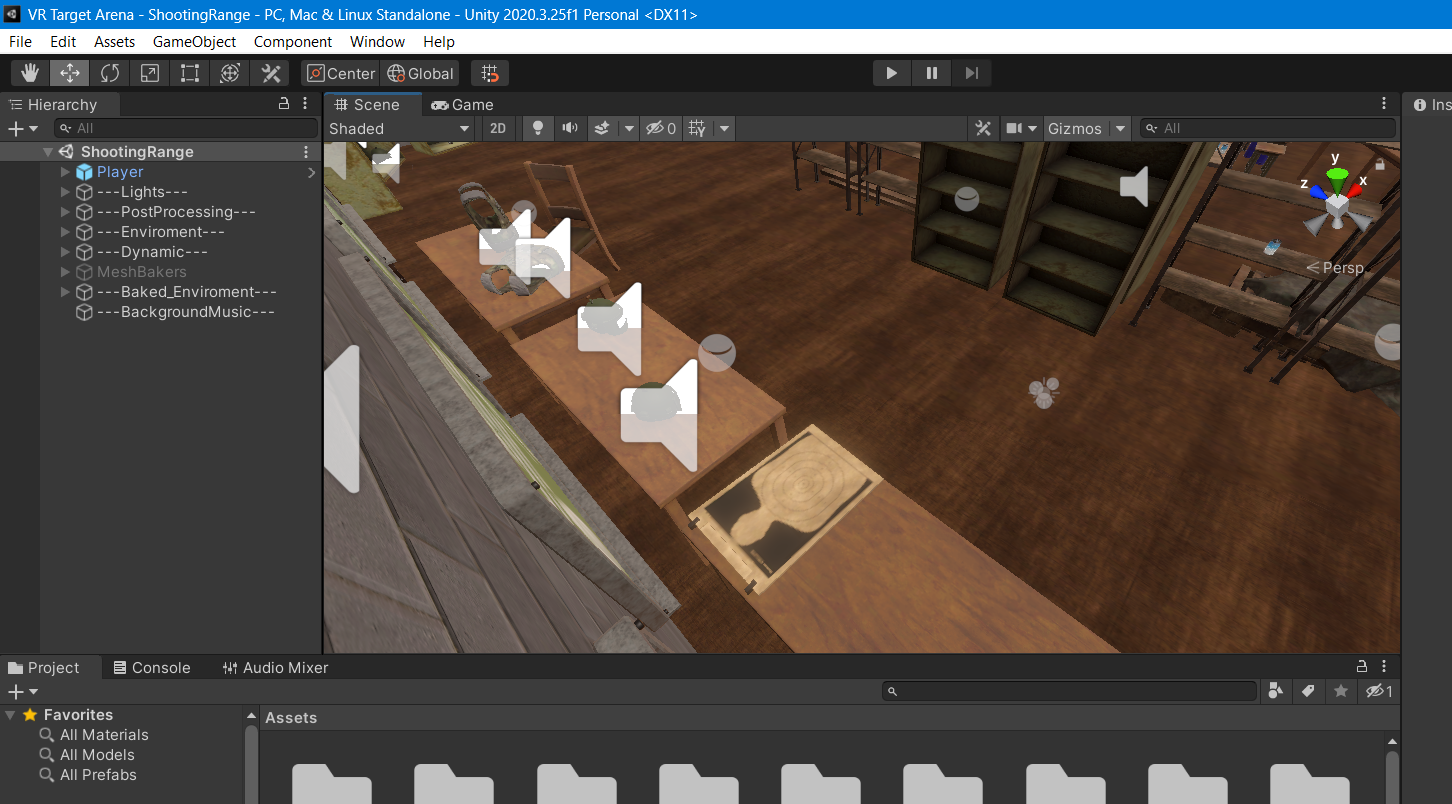
}

}

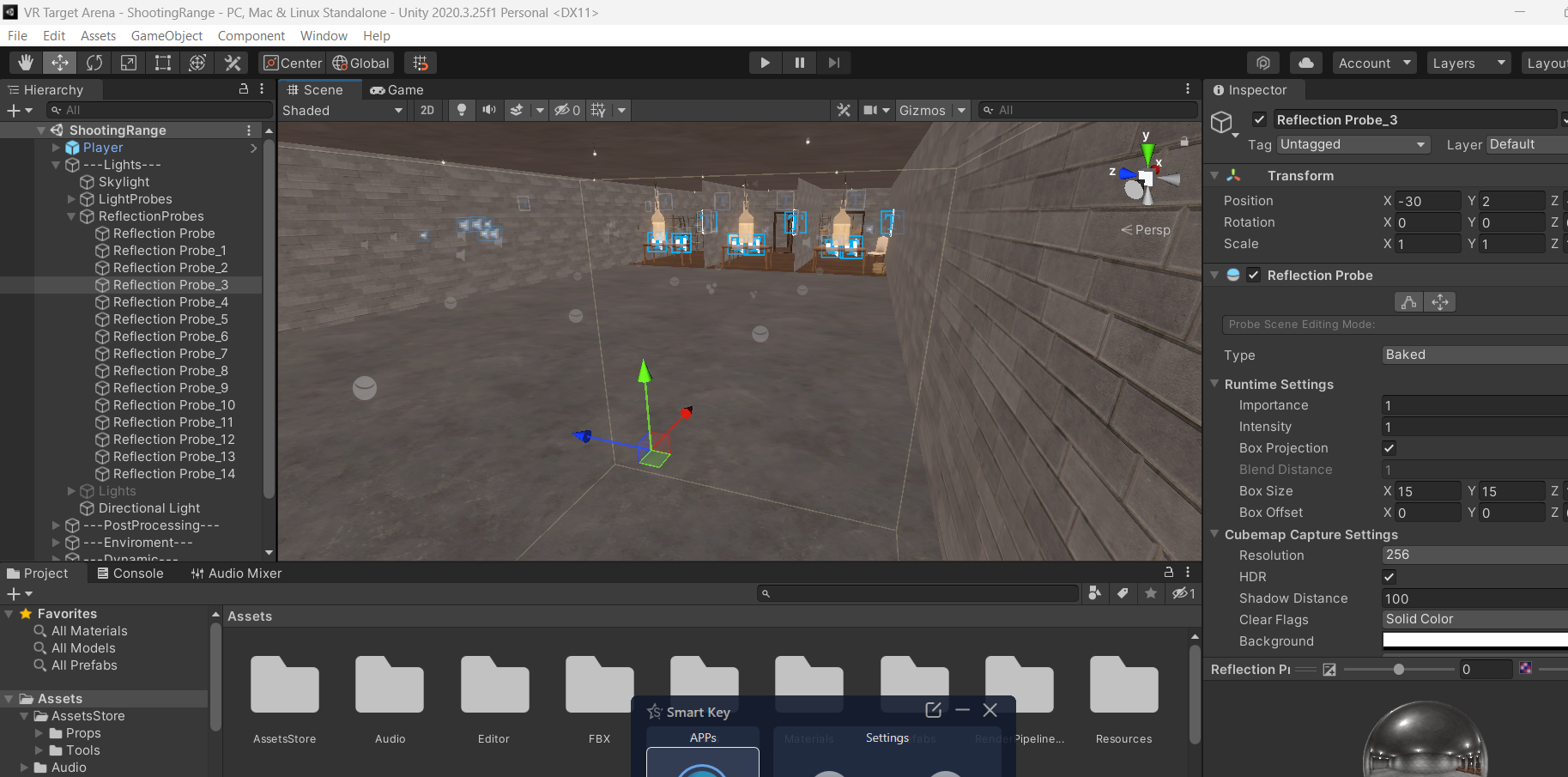
* Task 10: Demo Application

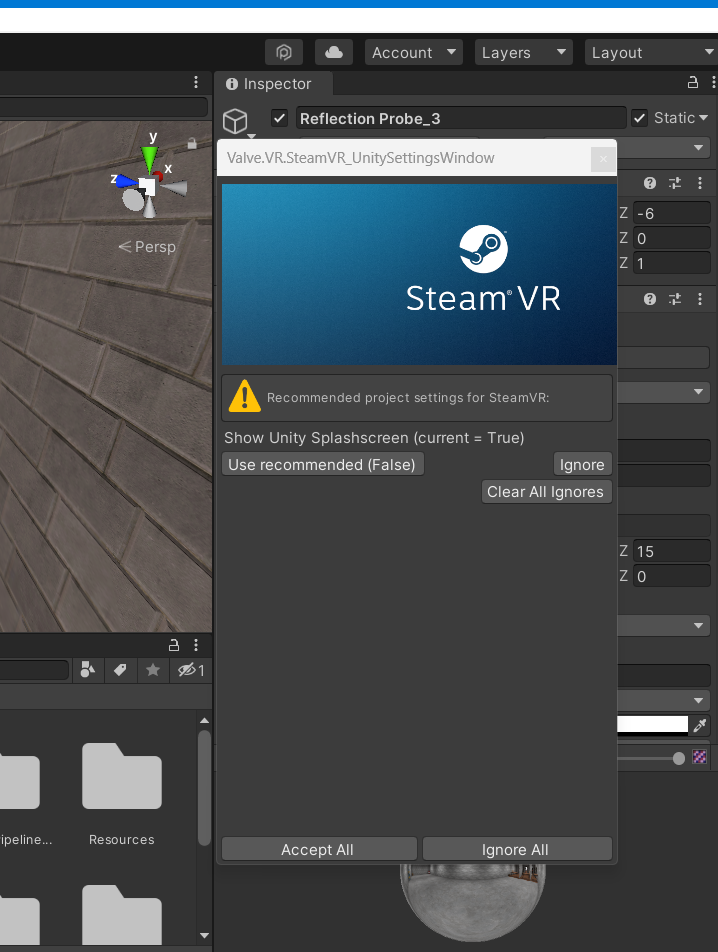
- A Screenshot & Screen Recording of a Demo:









****

**Challenges Faced**

1. Understanding XR Interactions

Learning and integrating the XR Interaction Toolkit to properly implement XR Ray Interactor and XR Grab Interactable components was challenging, especially ensuring smooth object interaction in VR.

2. Scene Setup and Alignment

Positioning the Target Arena environment to fit the VR space and ensuring proper alignment with the player’s viewpoint and interactable objects took careful adjustments.

3. Lighting and Shadows

Adding realistic lighting and shadows was tricky, as it required balancing visual quality with performance. Adjusting shadow settings without causing performance issues was particularly challenging.

4. Audio Integration

Integrating audio files and ensuring they worked well with 3D spatial sound in VR required understanding Unity’s audio system. Setting appropriate triggers for sound effects was time-consuming.

5. Asset Management

Handling large assets from the Unity Asset Store, such as the Target Arena, and managing their impact on project size and performance was a recurring challenge.

**References**

* Youtube
* Chatgpt
* GitHub
* Blackbox