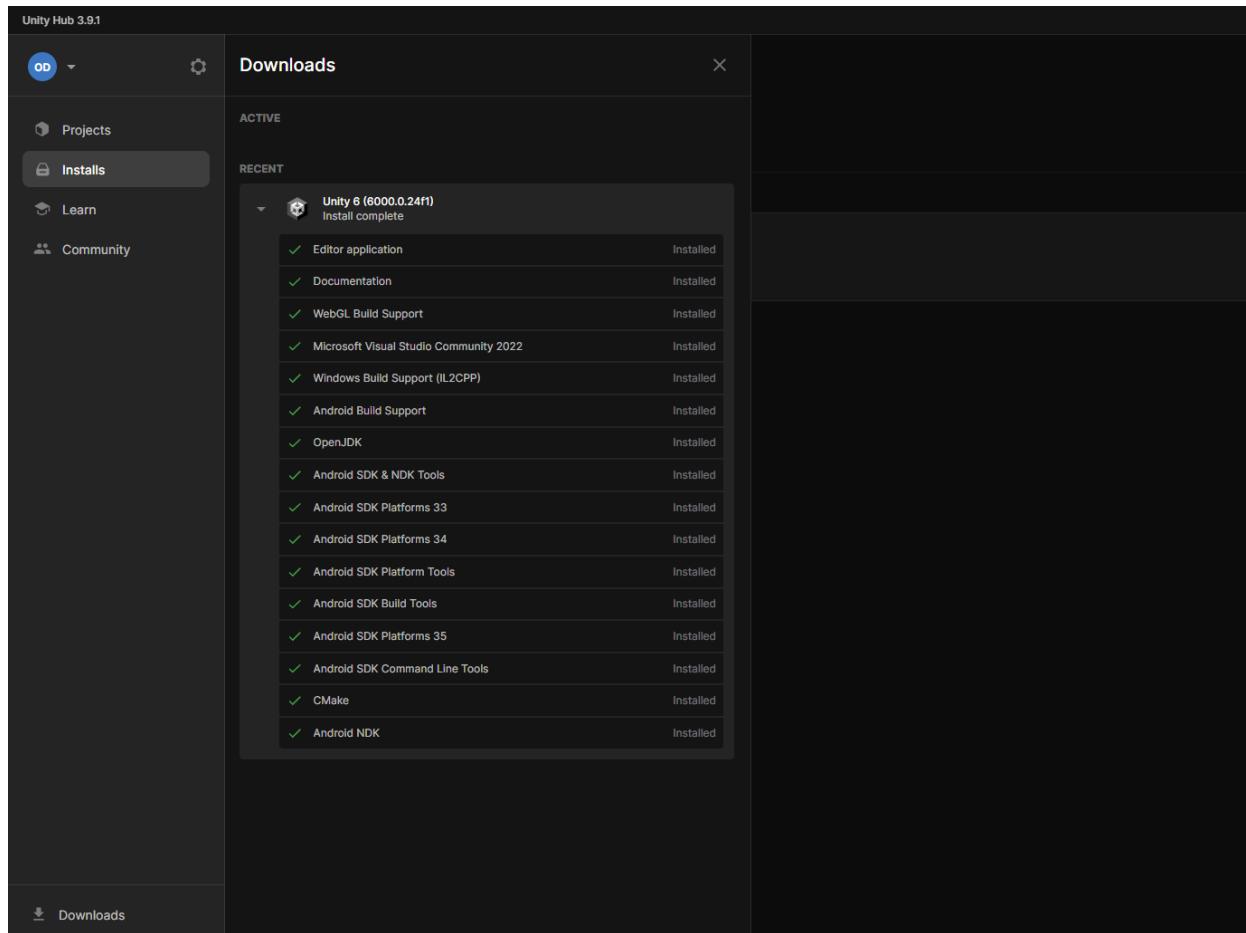


GAME PROGRAMMING PROJECT

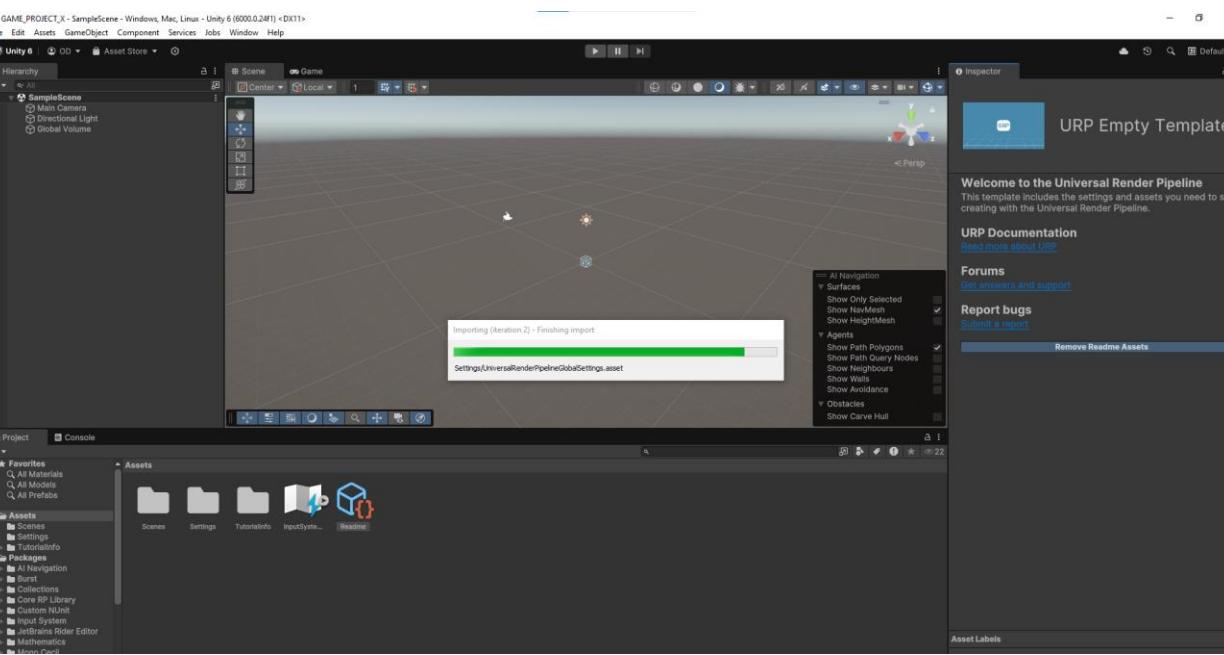
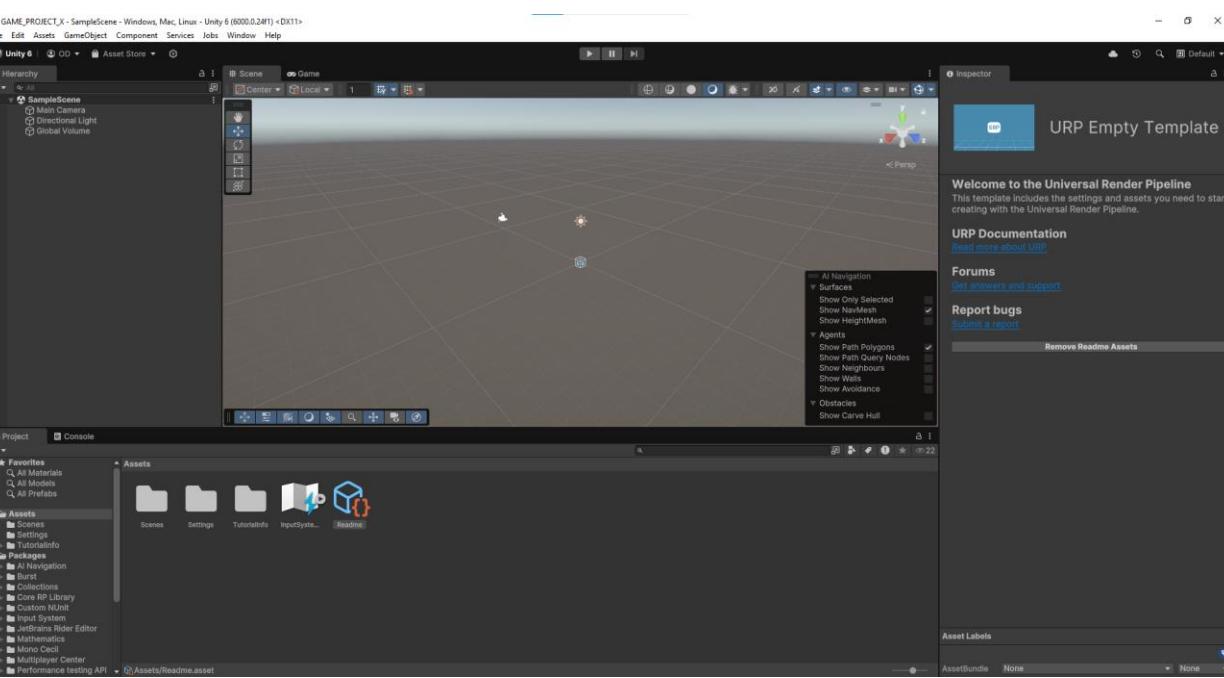
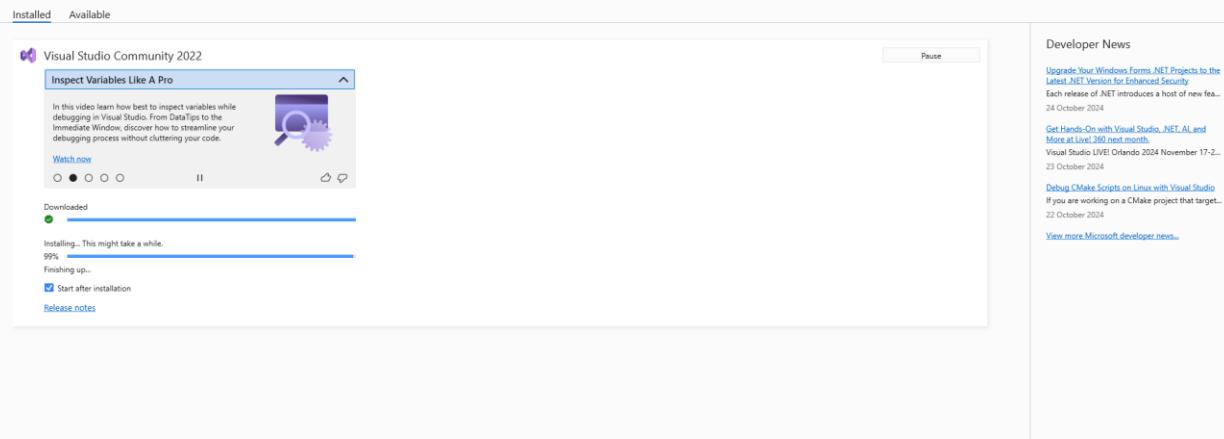
AR CAMPUS NAVIGATION

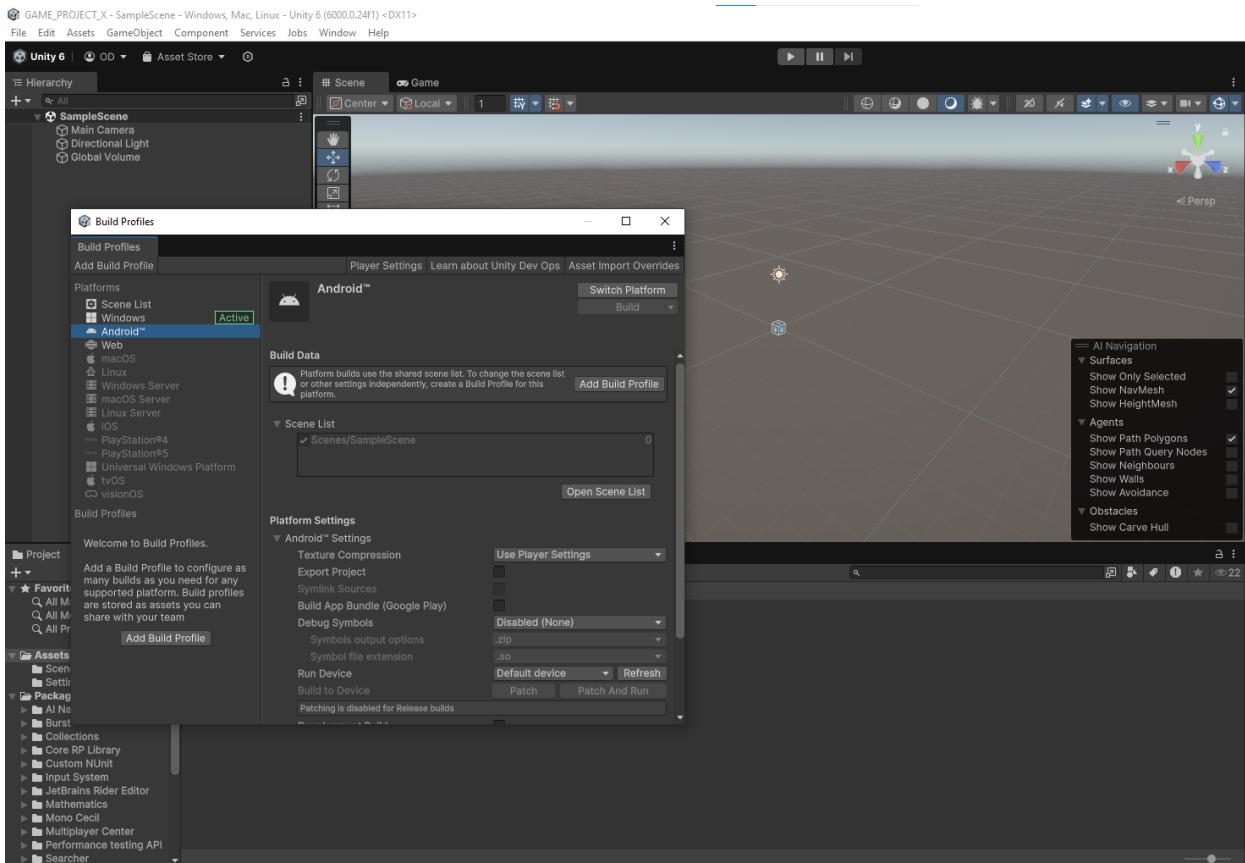
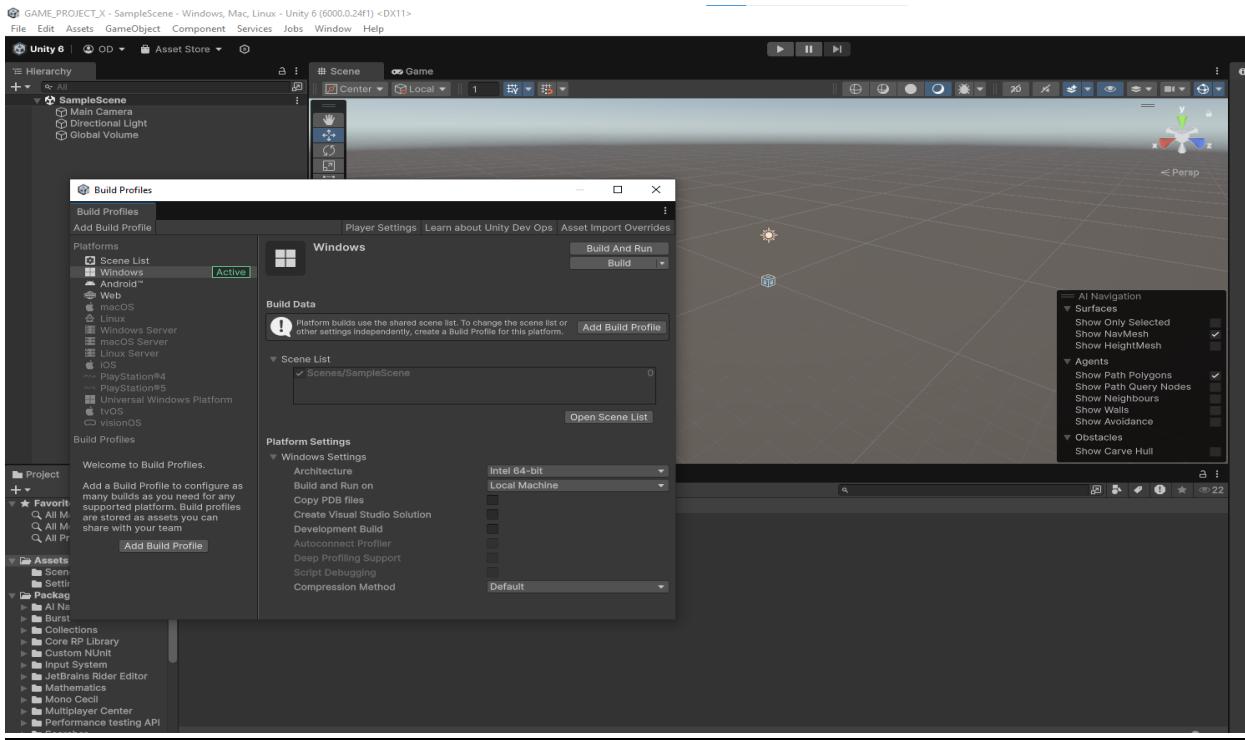
Overview

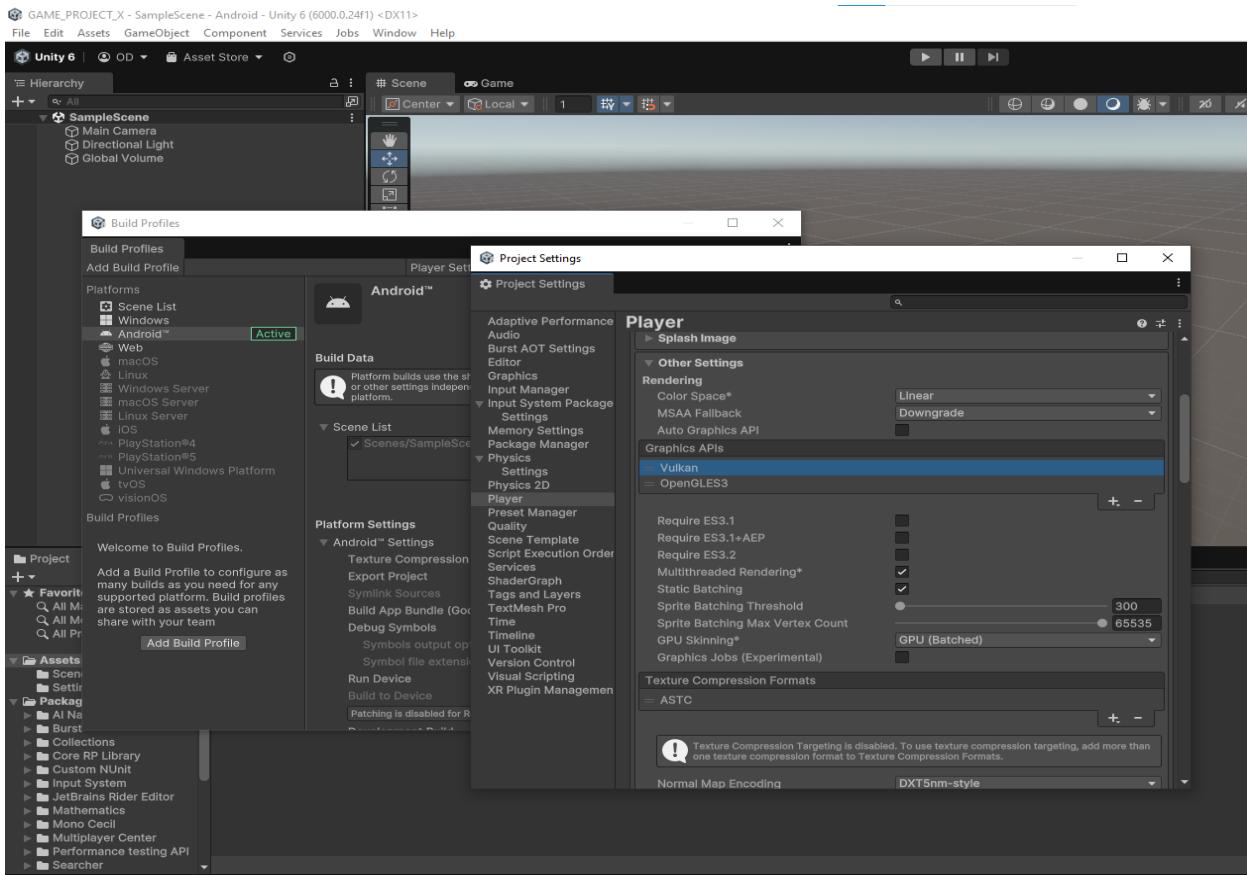
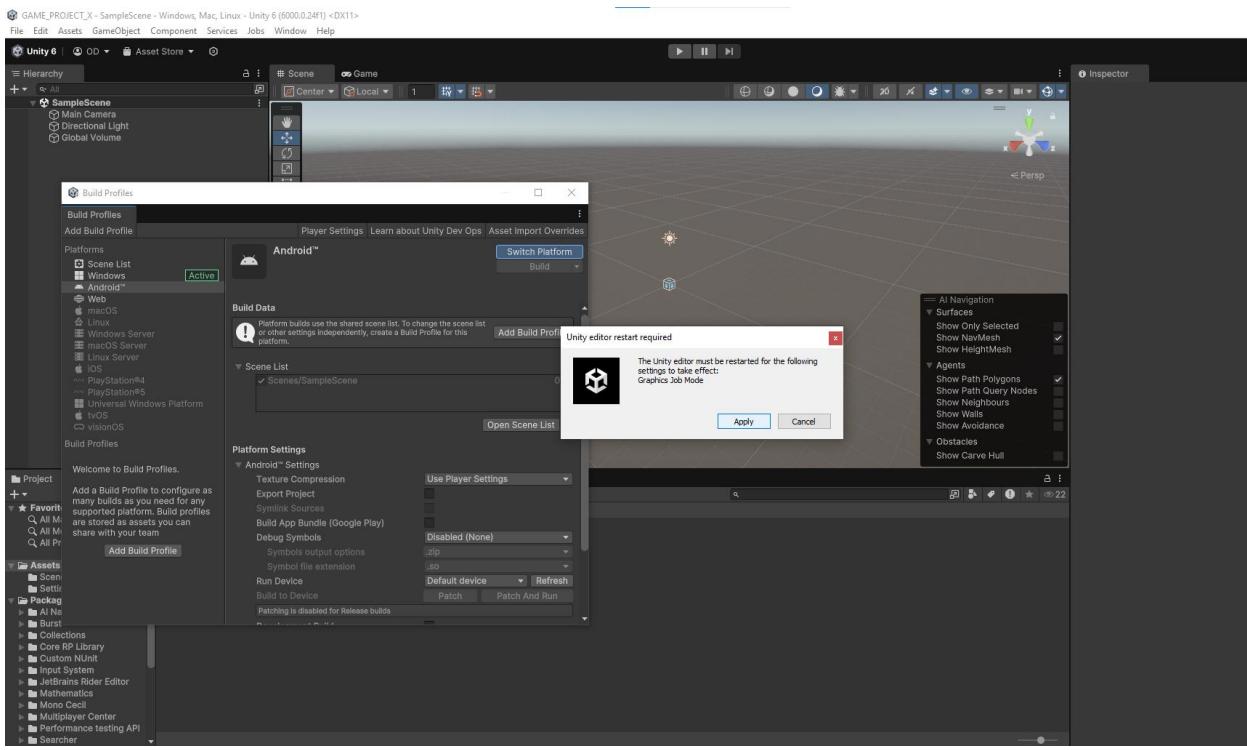
The AR Campus Navigation project leverages augmented reality to provide an interactive navigation experience on campus, guiding users through various landmarks, buildings, and facilities. This project focuses on using Unity's AR Foundation and other key plugins to set up AR-based directional guidance, which overlays navigation paths in the real-world environment as seen through the camera on a mobile device.

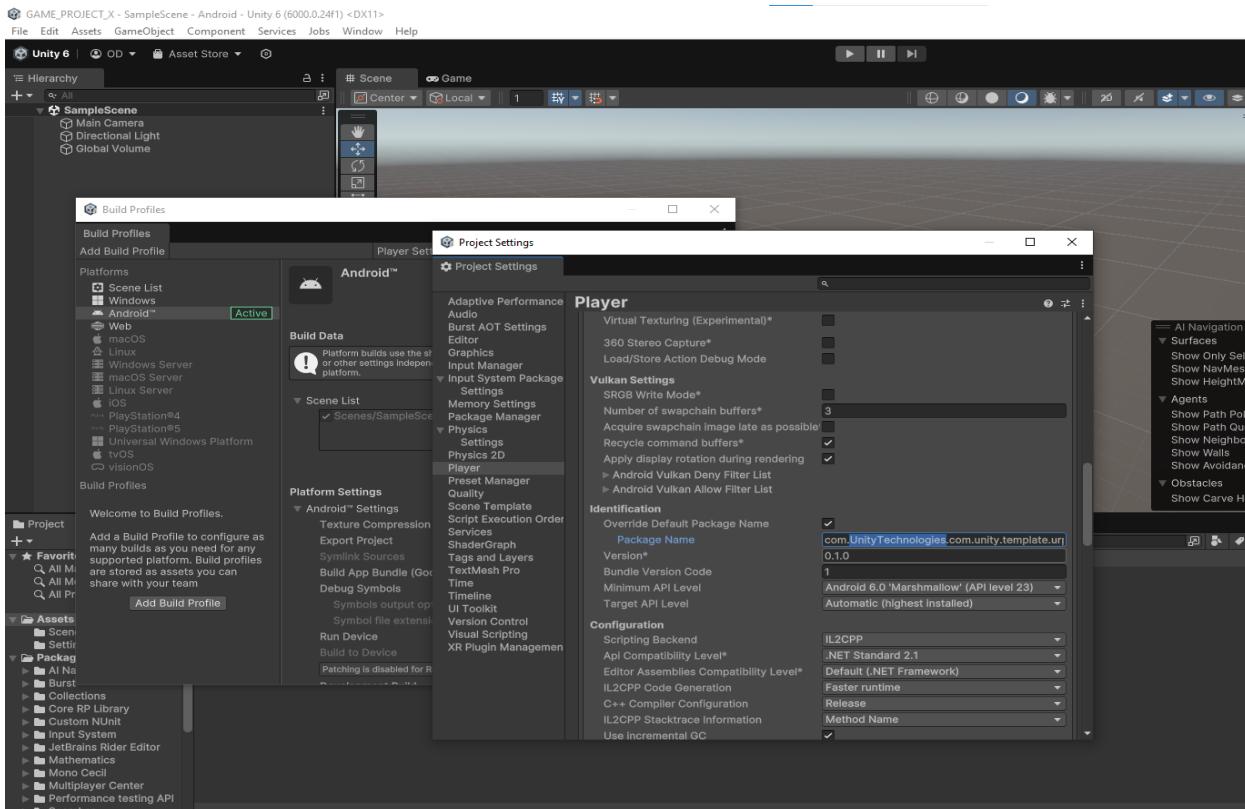
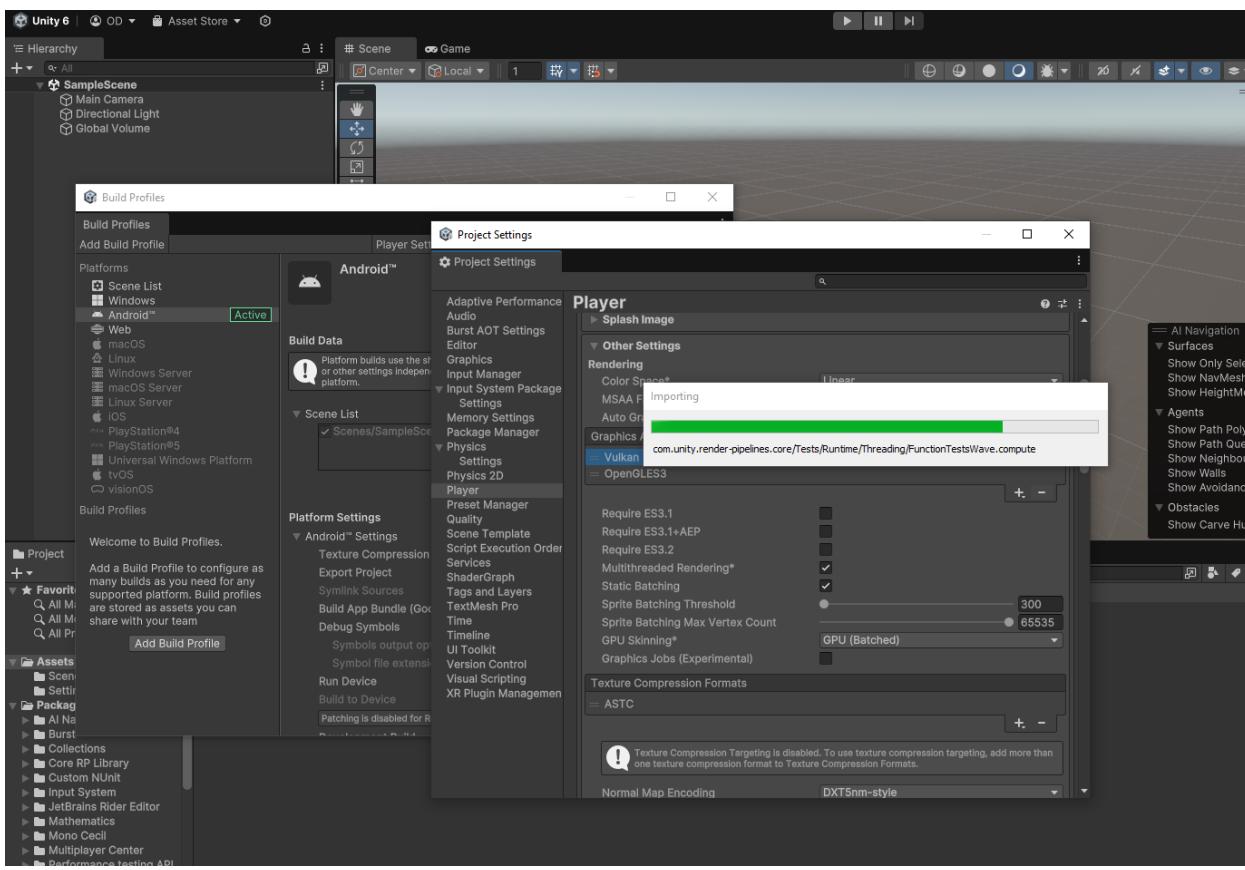


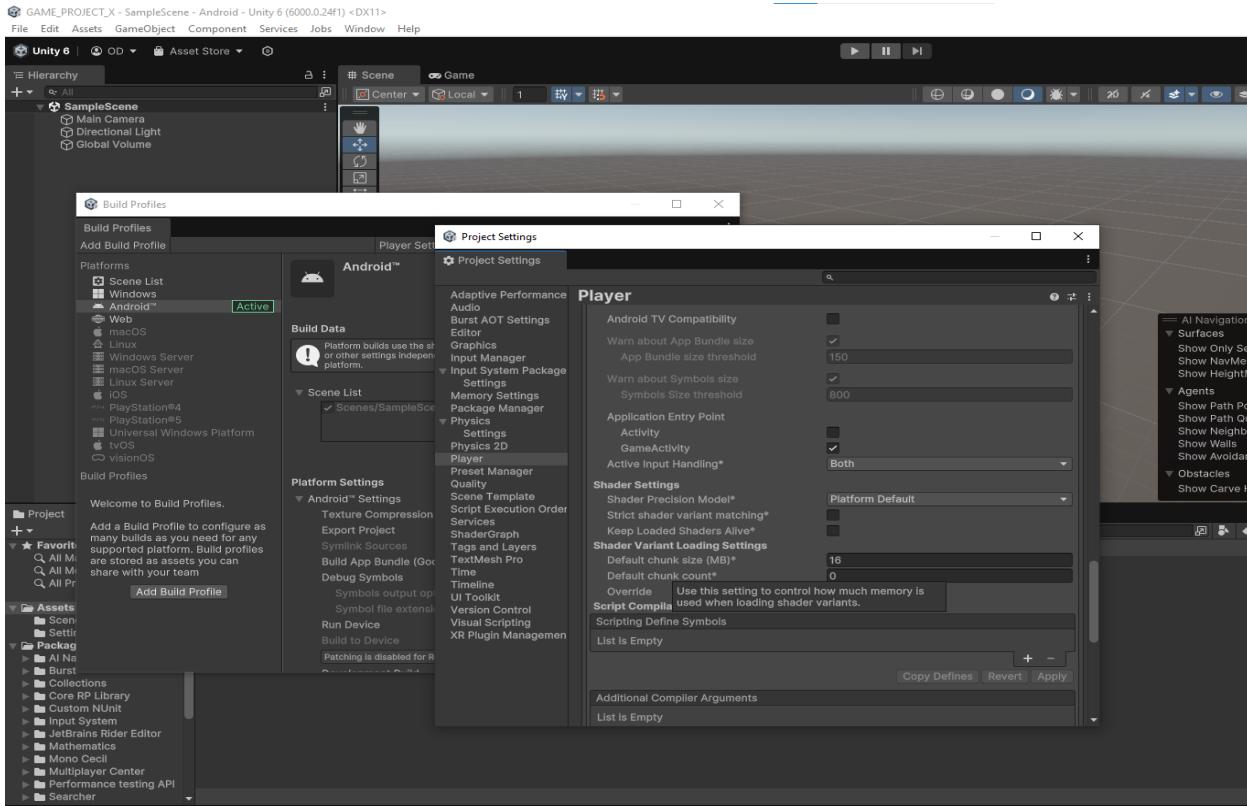
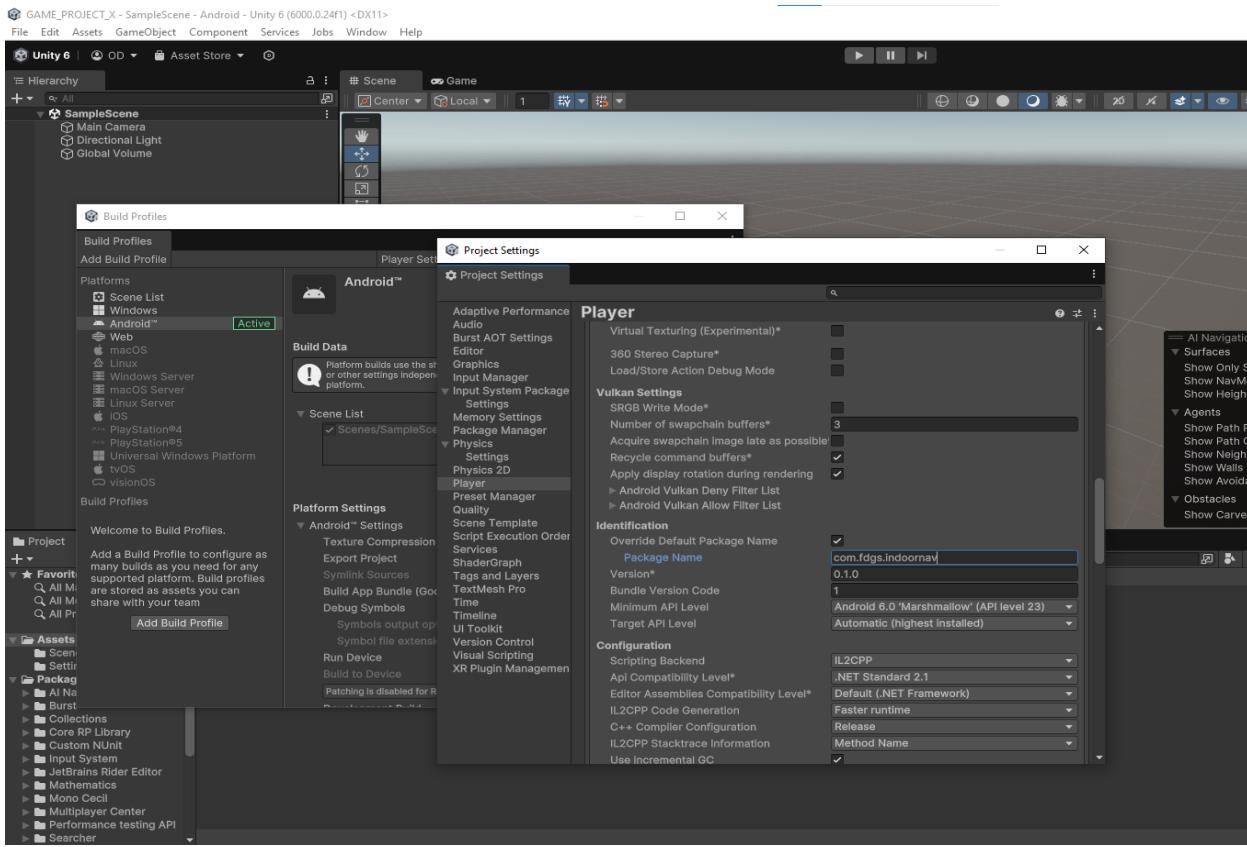
Visual Studio Installer

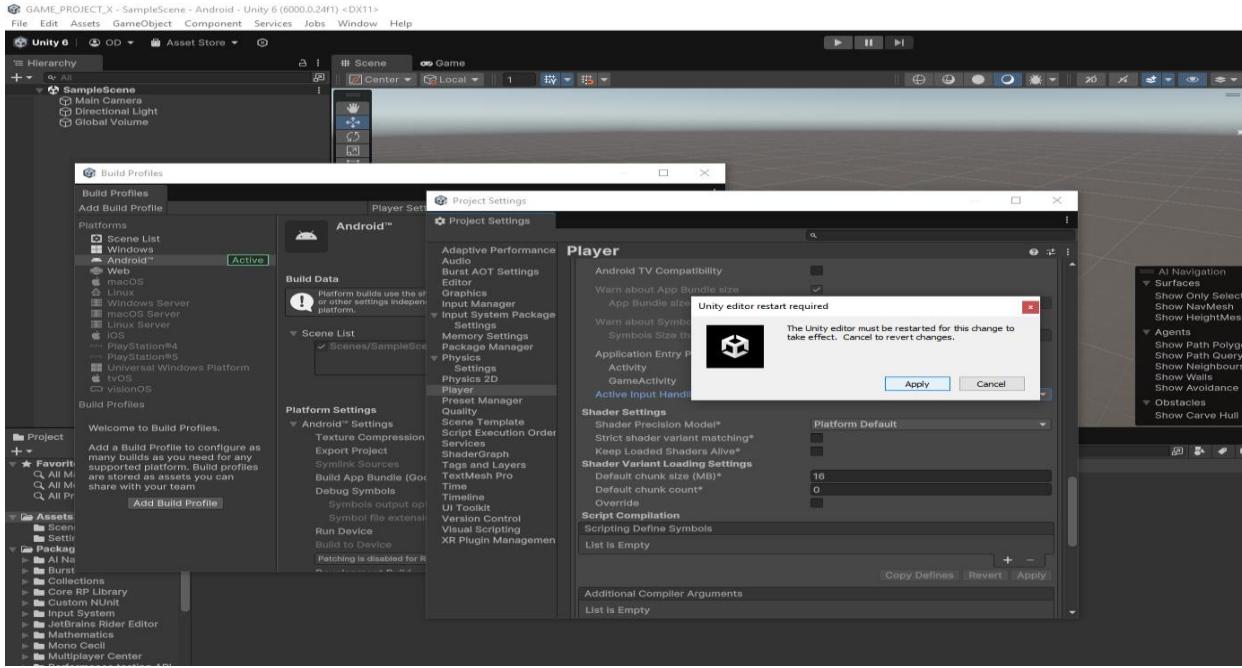
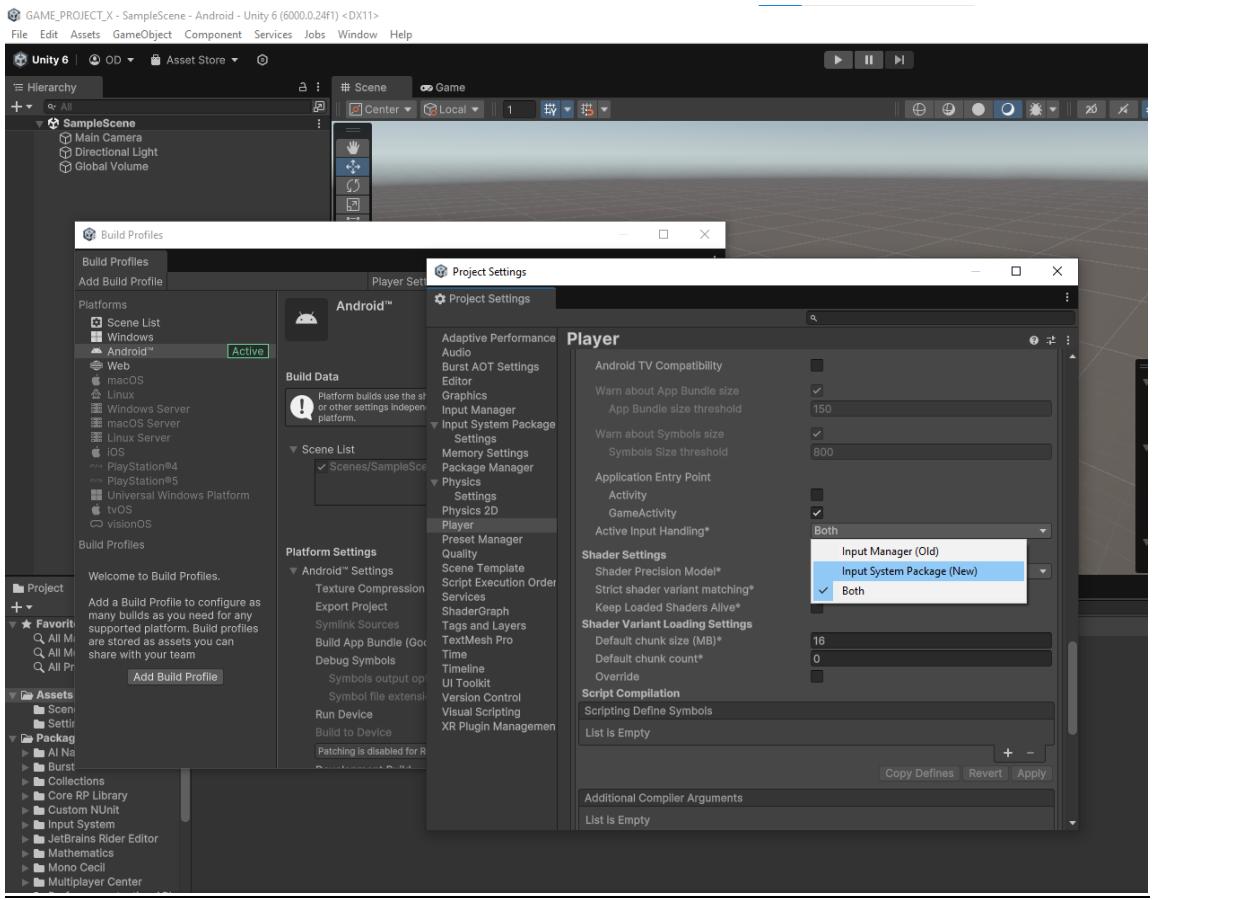


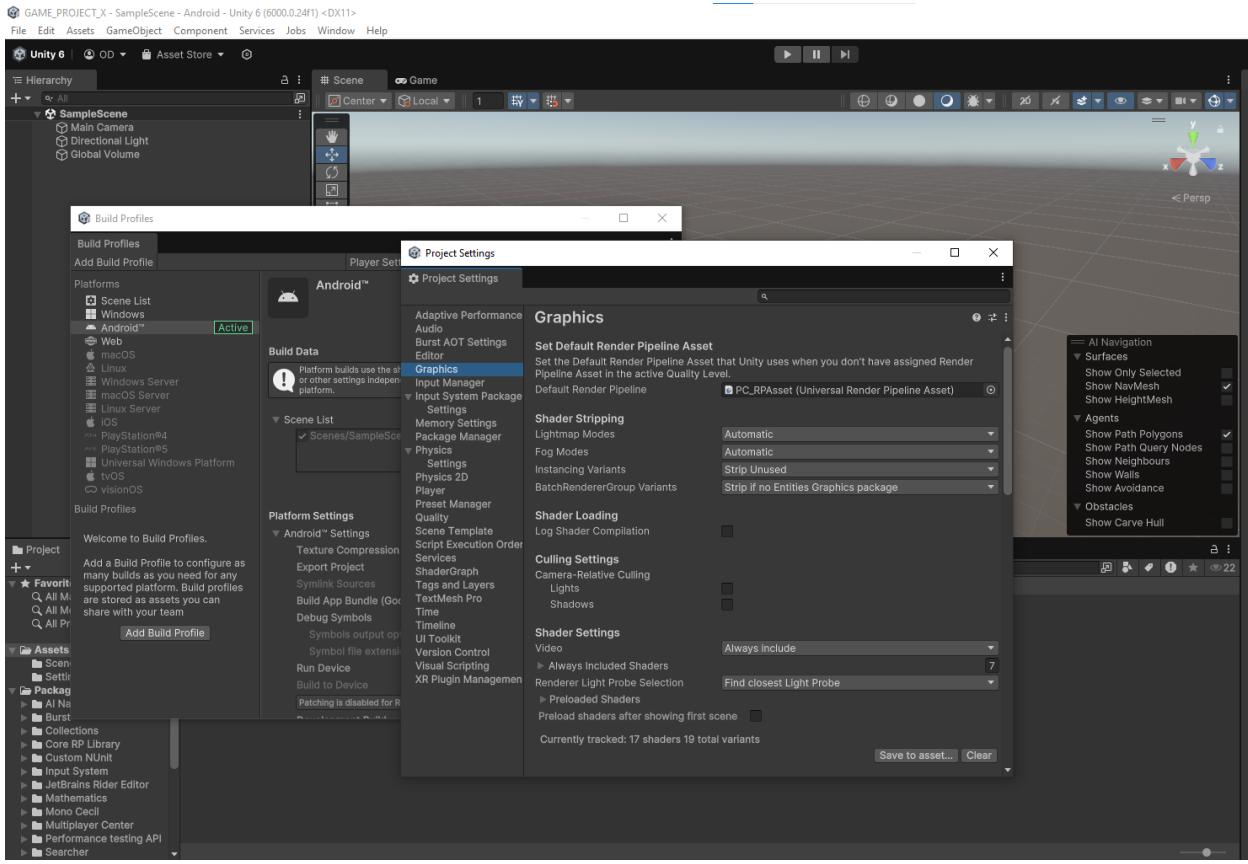
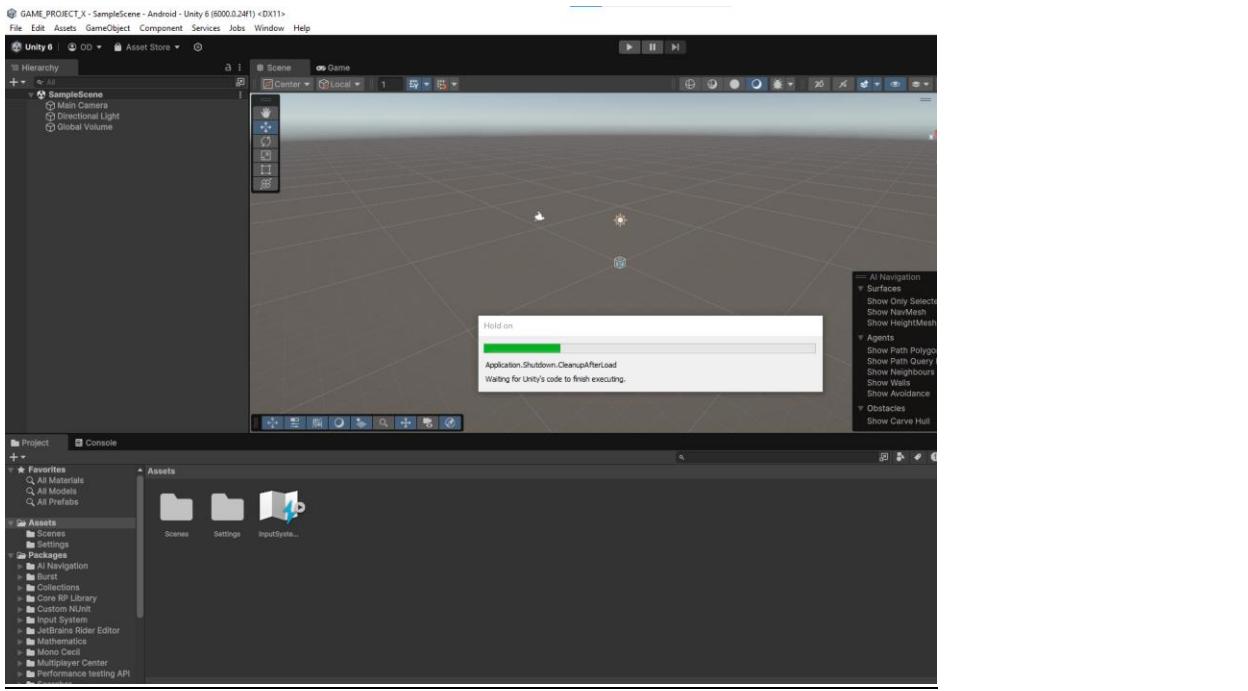


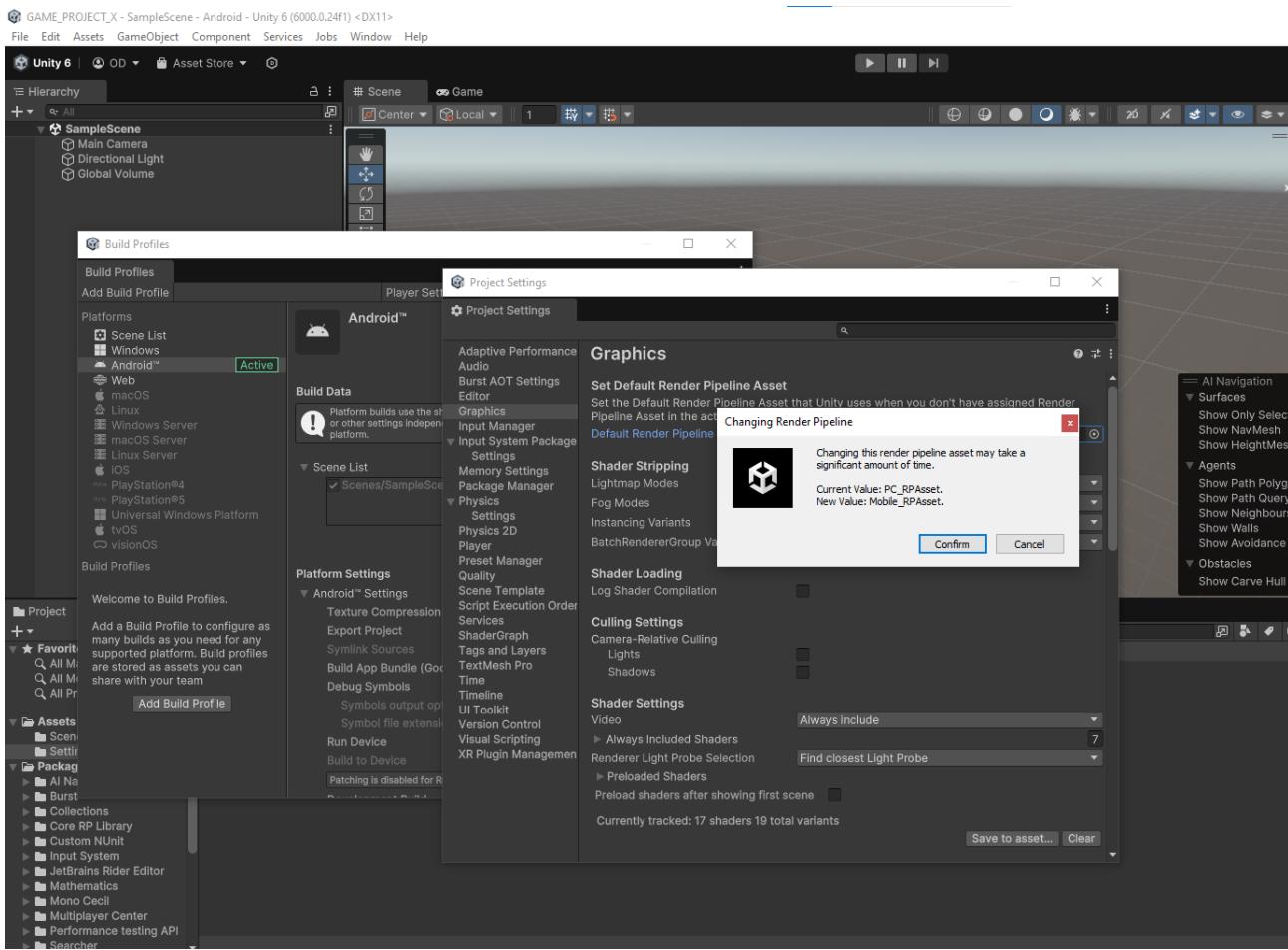
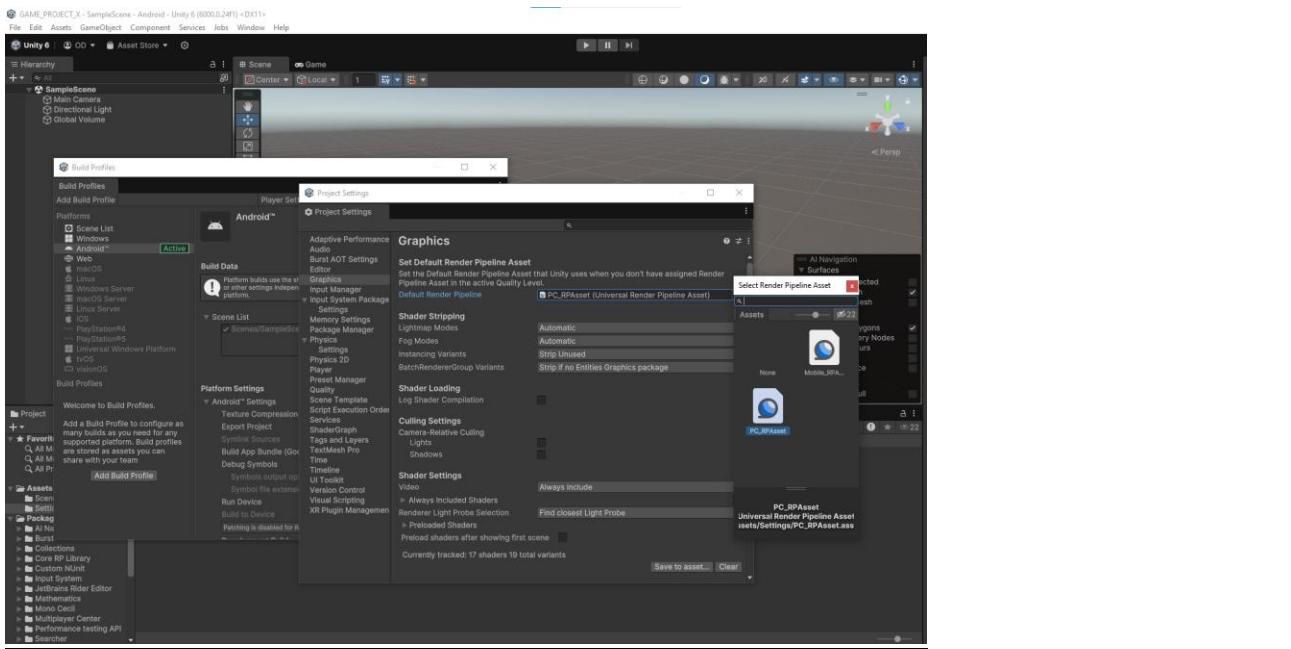


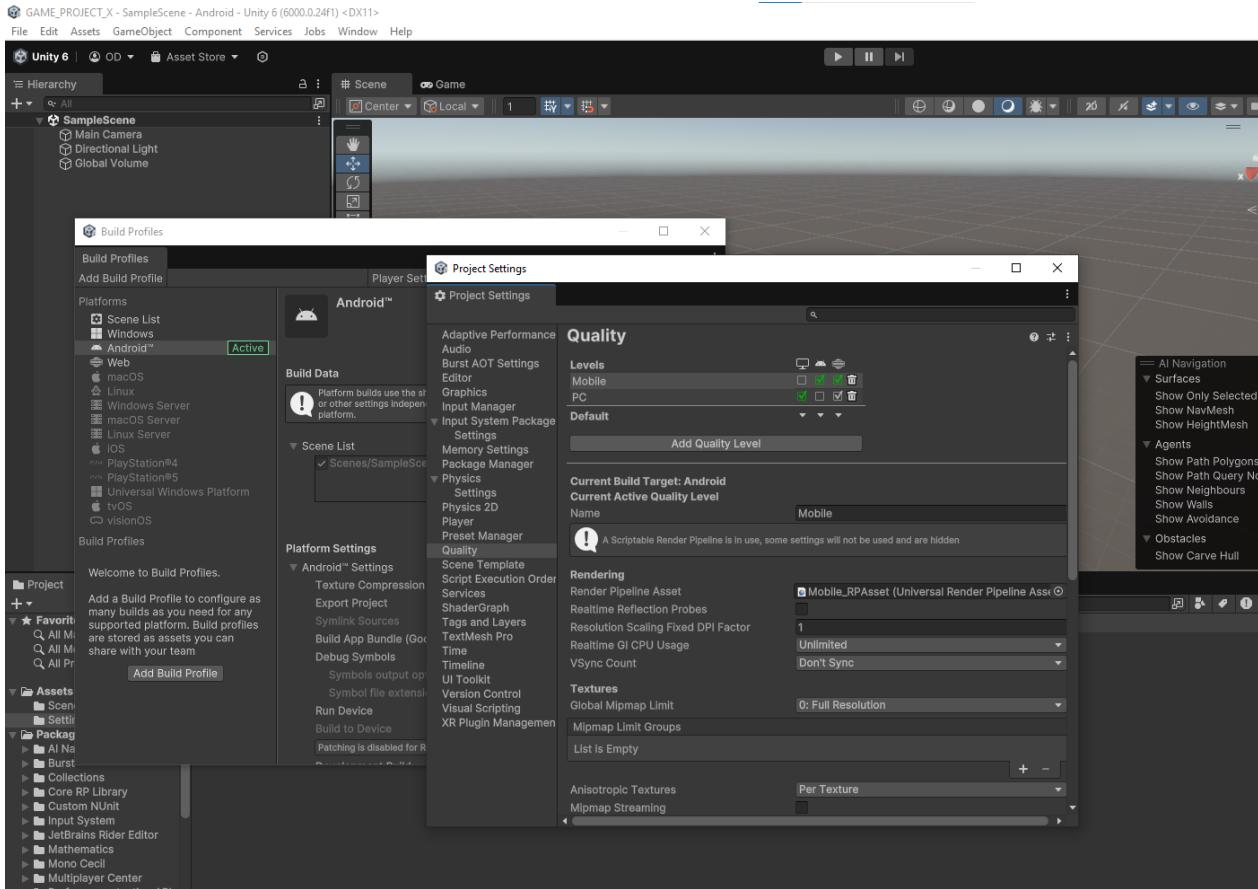
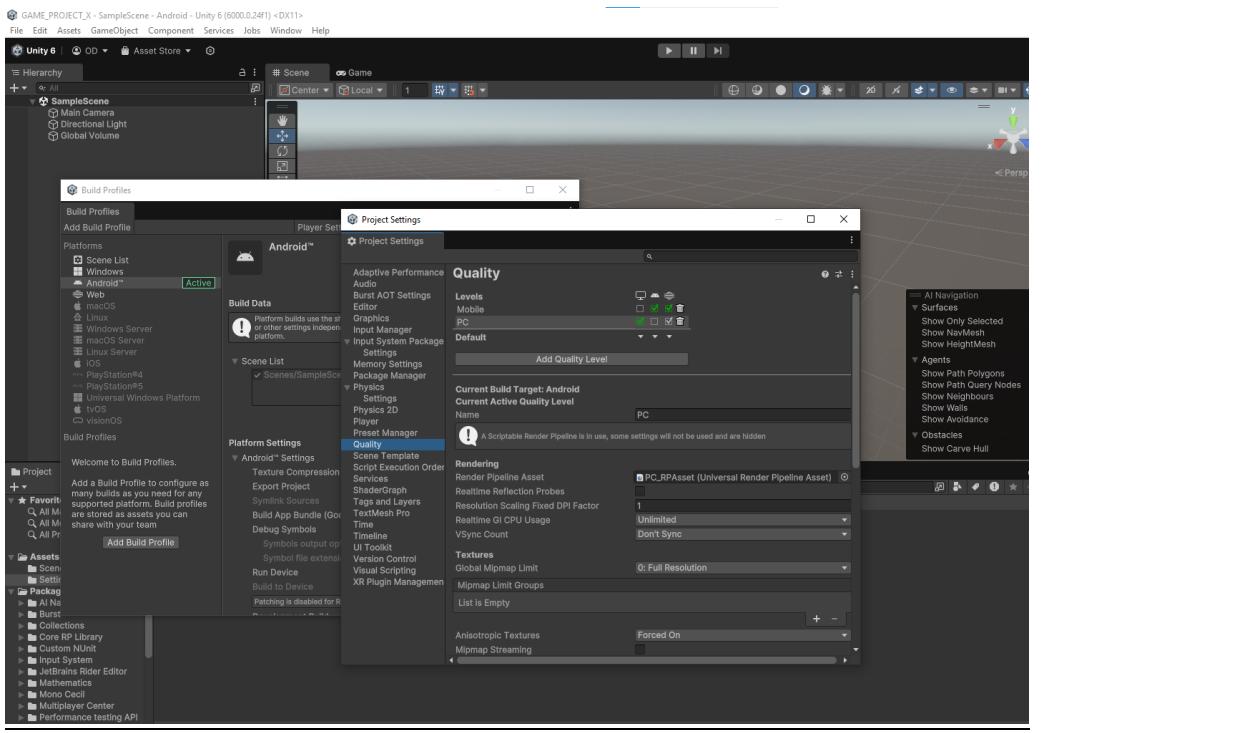


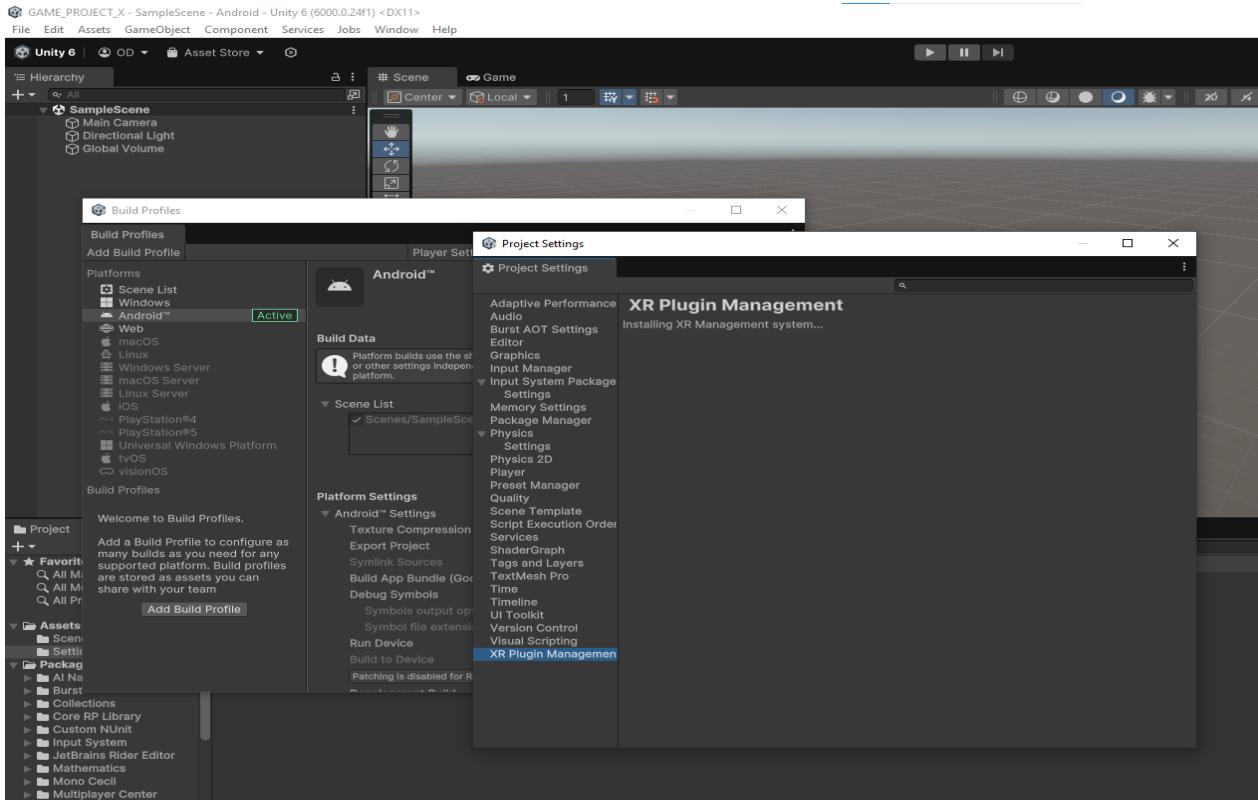
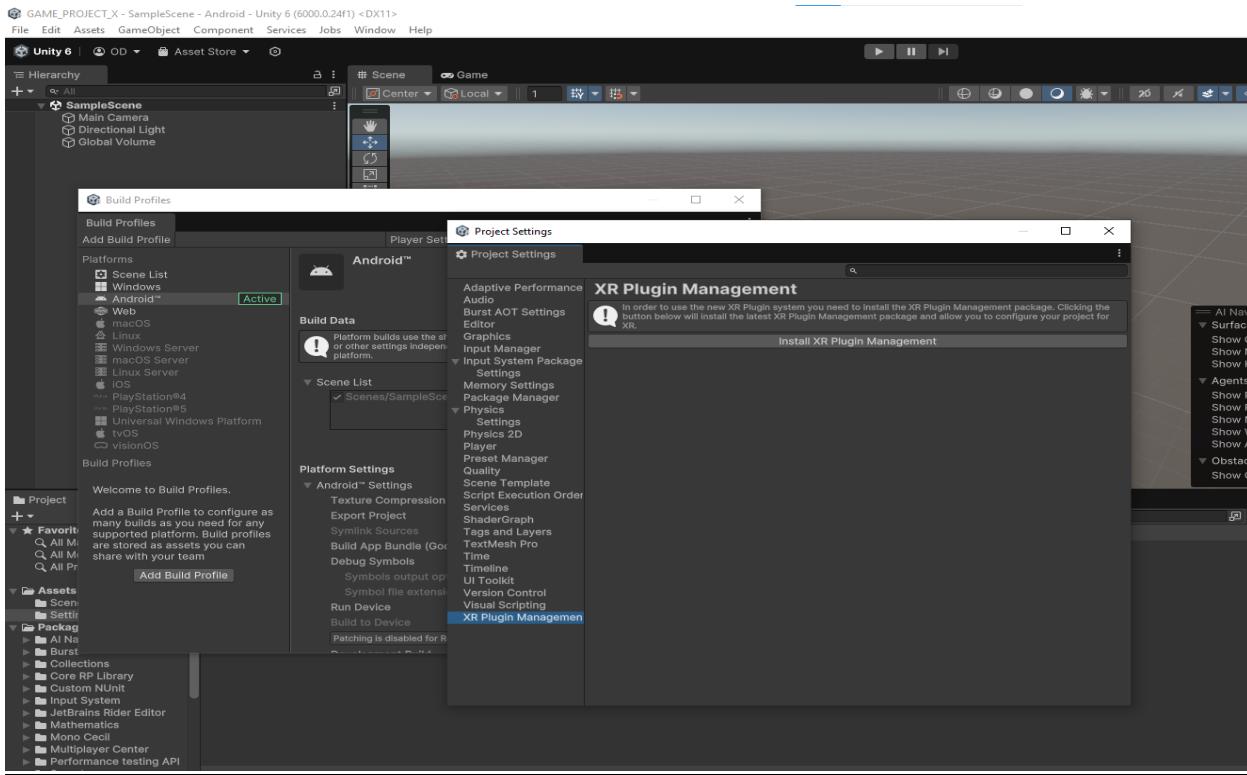


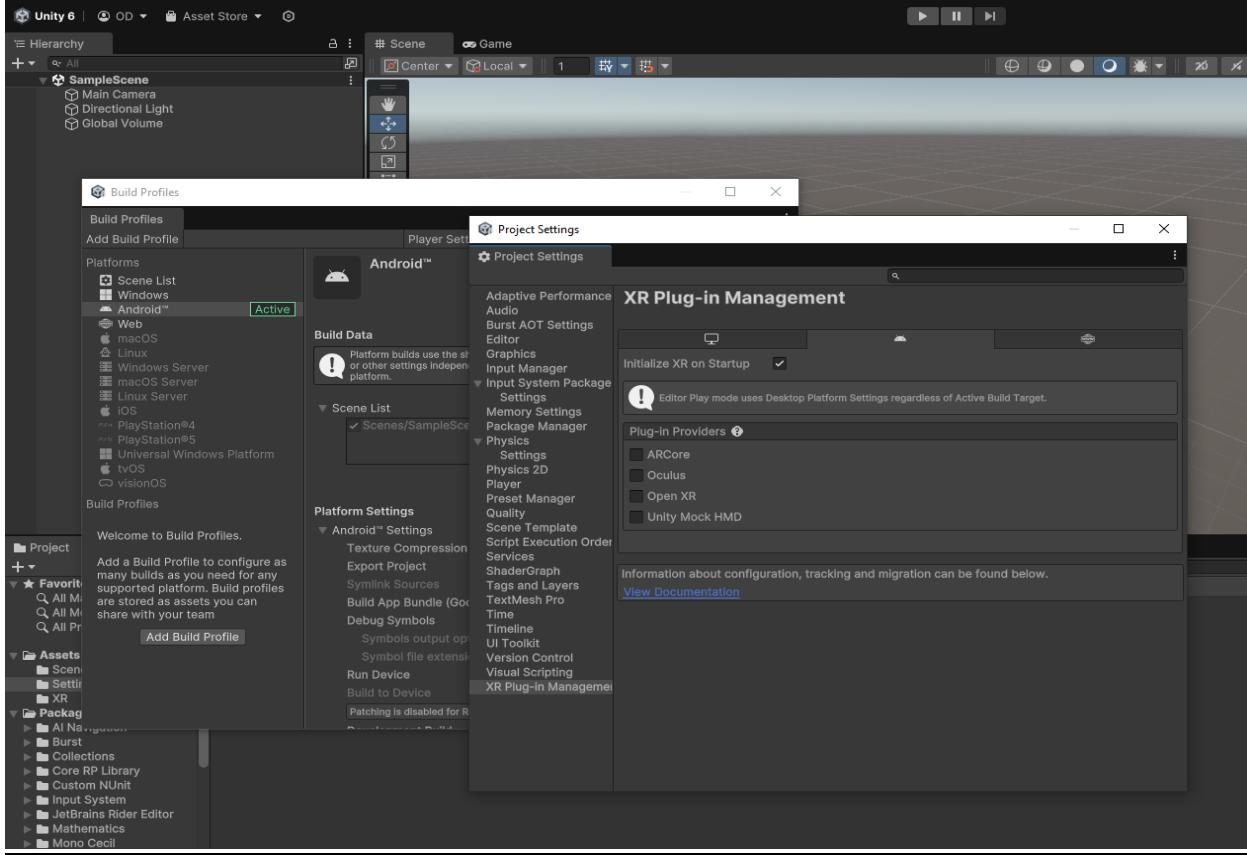


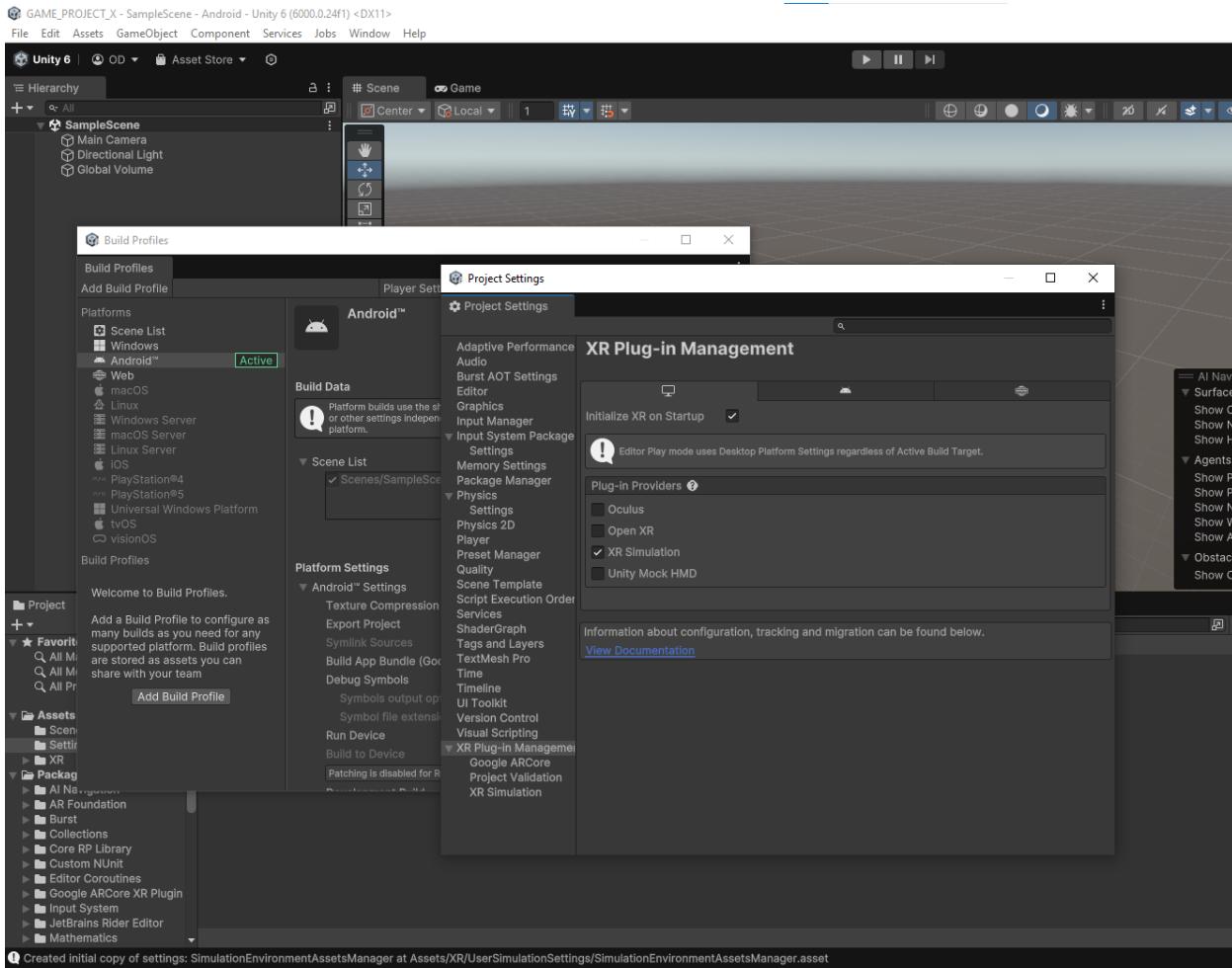
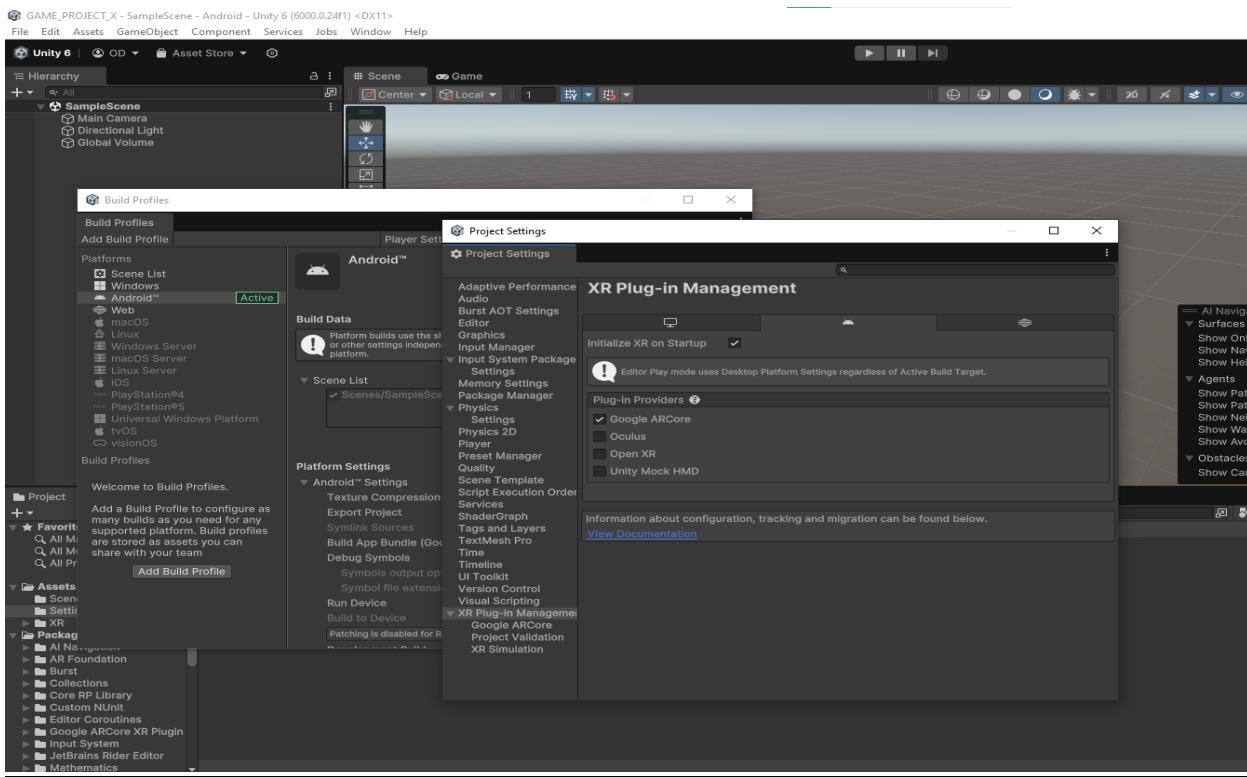


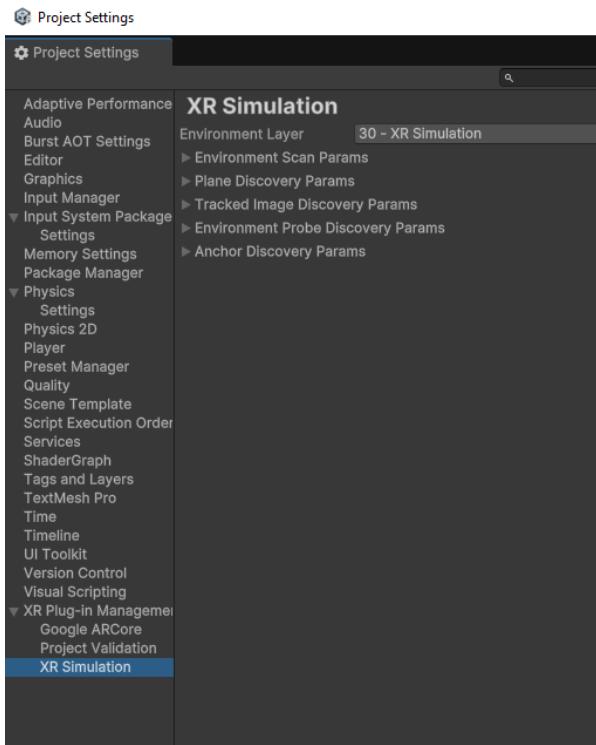
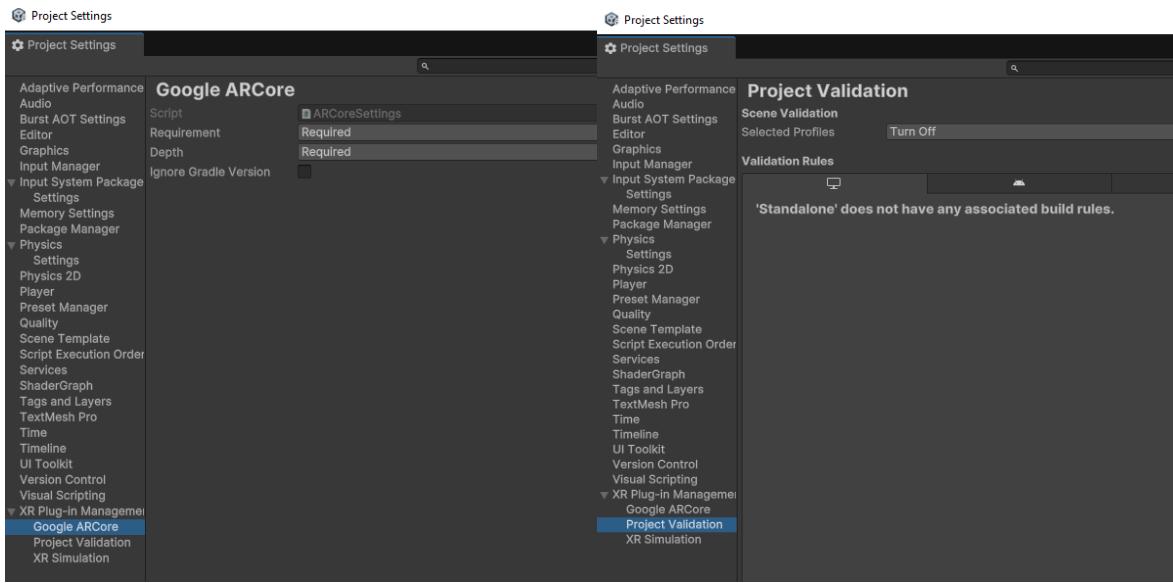


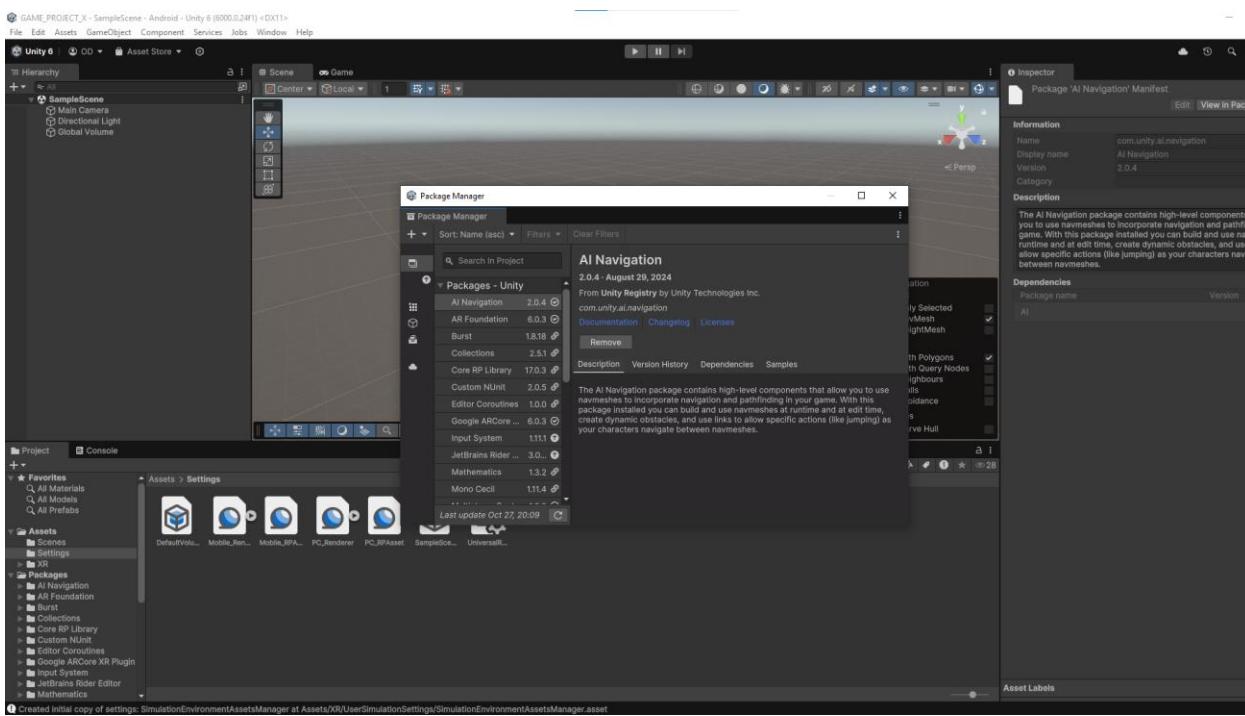
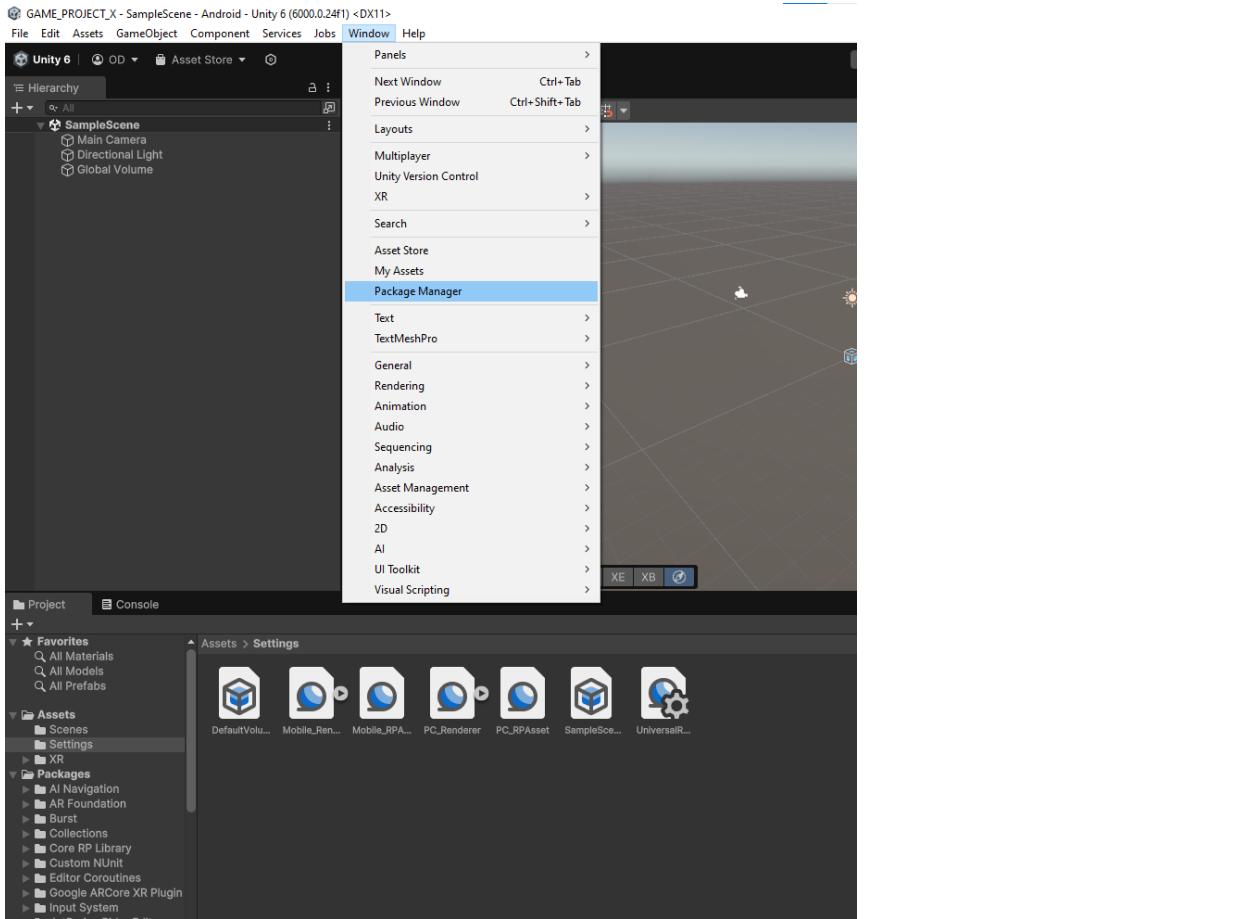














ARCore

Geospatial Creator

Documentation

Reference

Filter

ARCore

ARCore overview

What's New in ARCore

Supported devices

Downloads

AR codelabs

Sample apps

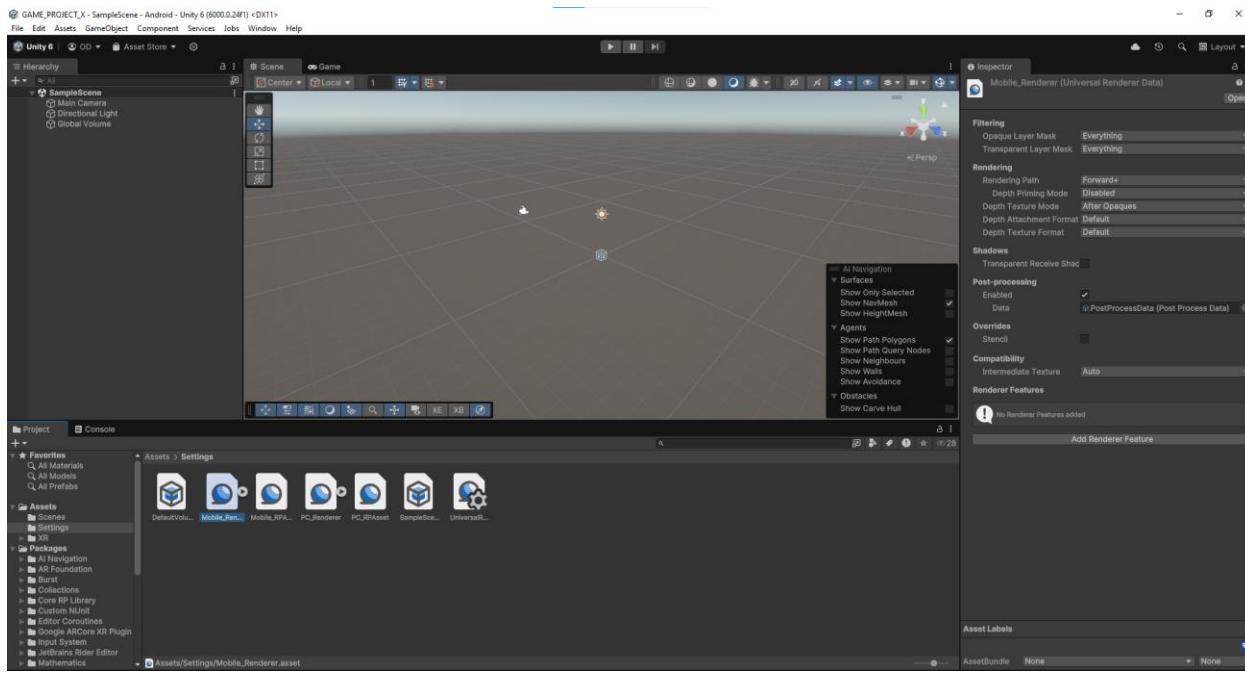
Android & iOS development

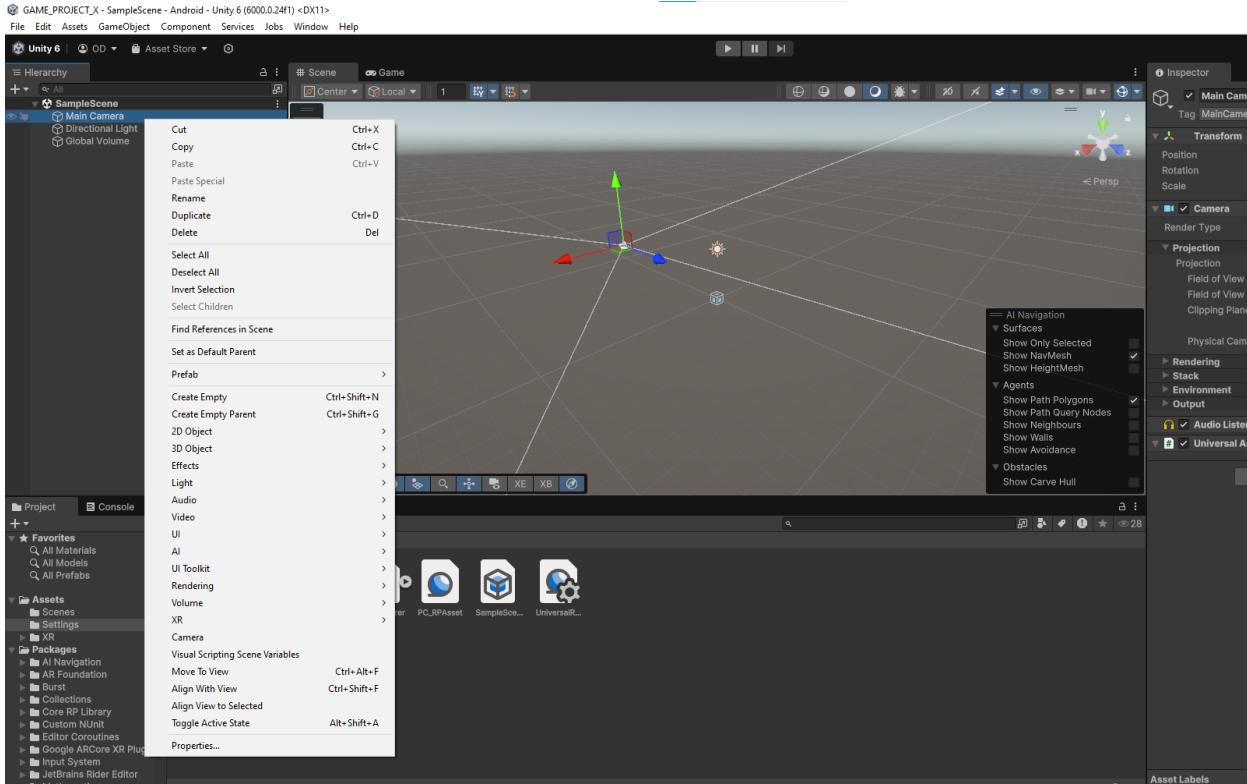
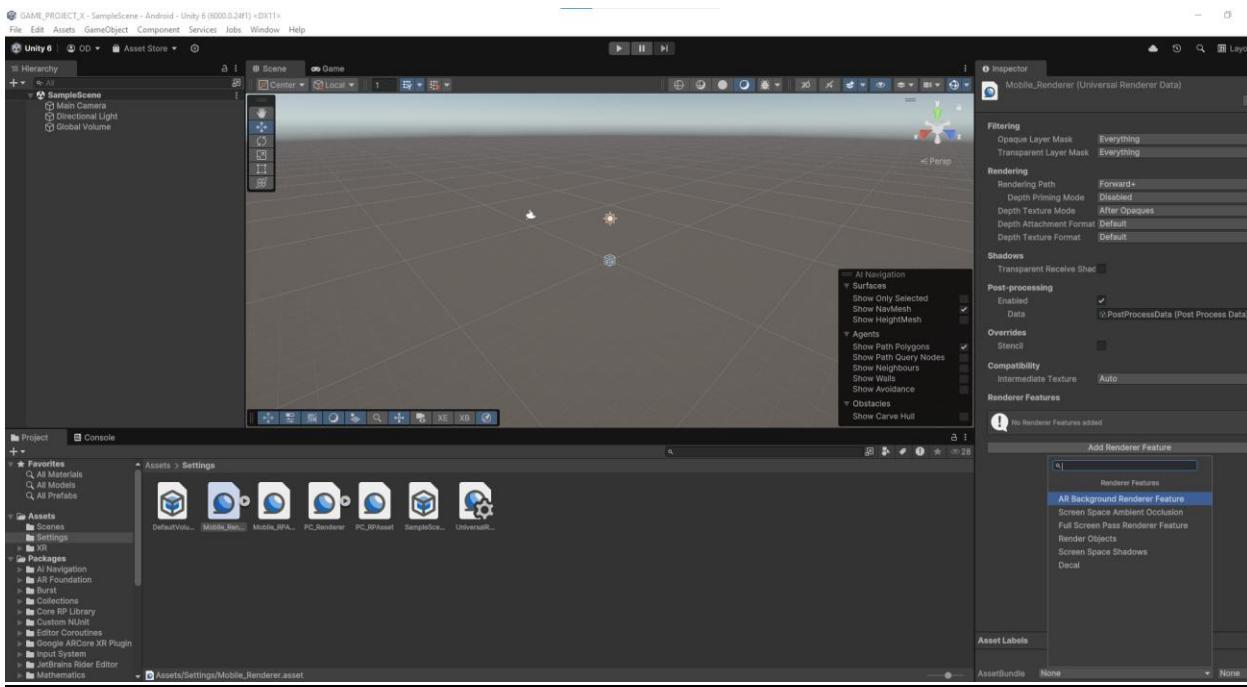
- ▶ Augmented Reality essentials
- ▶ Getting started
- ▶ Debugging
- ▶ Camera
- ▶ Hit-test
- ▶ Recording and Playback
- ▶ Instant Placement
- ▶ Depth
- ▶ Lighting Estimation
- ▶ Augmented Faces
- ▶ Augmented Images
- ▶ ARCore API on Google Cloud
- ▶ Cloud Anchors
- ▶ Geospatial
- ▶ Geospatial Creator
- ▶ Scene Semantics
- ▶ Vulkan Rendering
- ▶ Electronic Image Stabilization
- ▶ Machine learning with ARCore

Web development

WebXR

Samsung	Galaxy M14 5G	Supports Depth API
Samsung	Galaxy M15 5G	Supports Depth API
Samsung	Galaxy M20	Requires Android 10 or later
Samsung	Galaxy M21	
Samsung	Galaxy M22	
Samsung	Galaxy M23 5G	
Samsung	Galaxy M30s	
Samsung	Galaxy M31	
Samsung	Galaxy M31s	
Samsung	Galaxy M32	
Samsung	Galaxy M33 5G	
Samsung	Galaxy M34 5G	Supports Depth API
Samsung	Galaxy M51	
Samsung	Galaxy M52 5G	Supports Depth API
Samsung	Galaxy M53 5G	Supports Depth API
Samsung	Galaxy M54 5G	Supports Depth API
Samsung	Galaxy M62	
Samsung	Galaxy Note8	Supports multiple GPU texture resolutions - 1080p, 720p, 480p Supports Depth API
Samsung	Galaxy Note9	Supports Depth API
Samsung	Galaxy Note10	Supports multiple GPU texture resolutions - 1080p, 720p, 480p Supports Depth API
Samsung	Galaxy Note10 5G	Supports multiple GPU texture resolutions - 1080p, 720p, 480p Supports Depth API

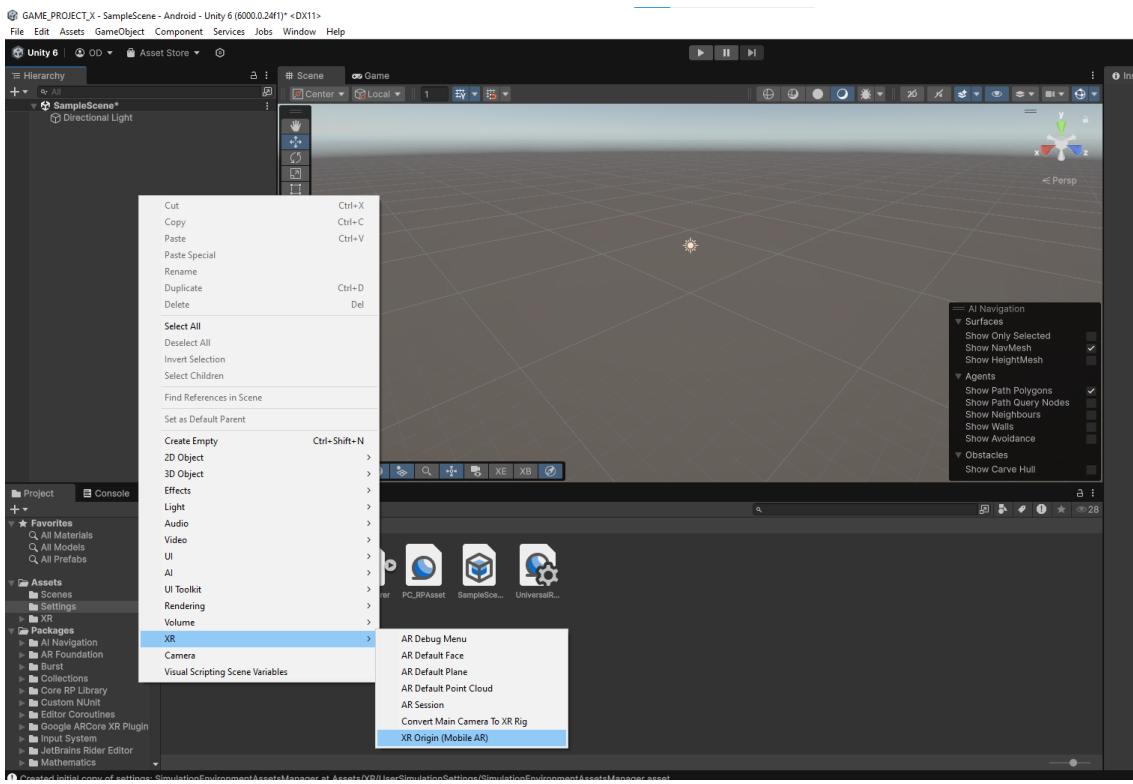




Initial Setup

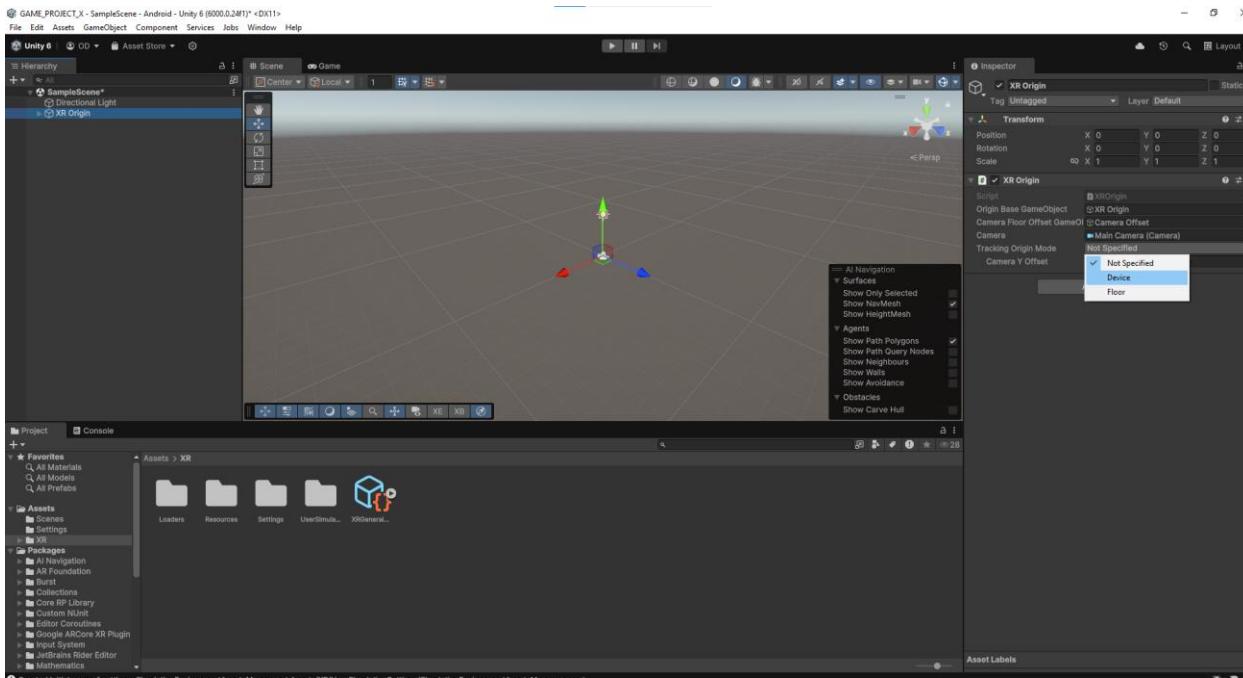
1. Delete Main Camera and Global Volume

Removing the main camera and global volume from the default setup is essential to avoid conflicts with the AR Camera that will be added later. The AR Camera will handle all rendering requirements, enabling a seamless overlay of AR components onto the device's display.



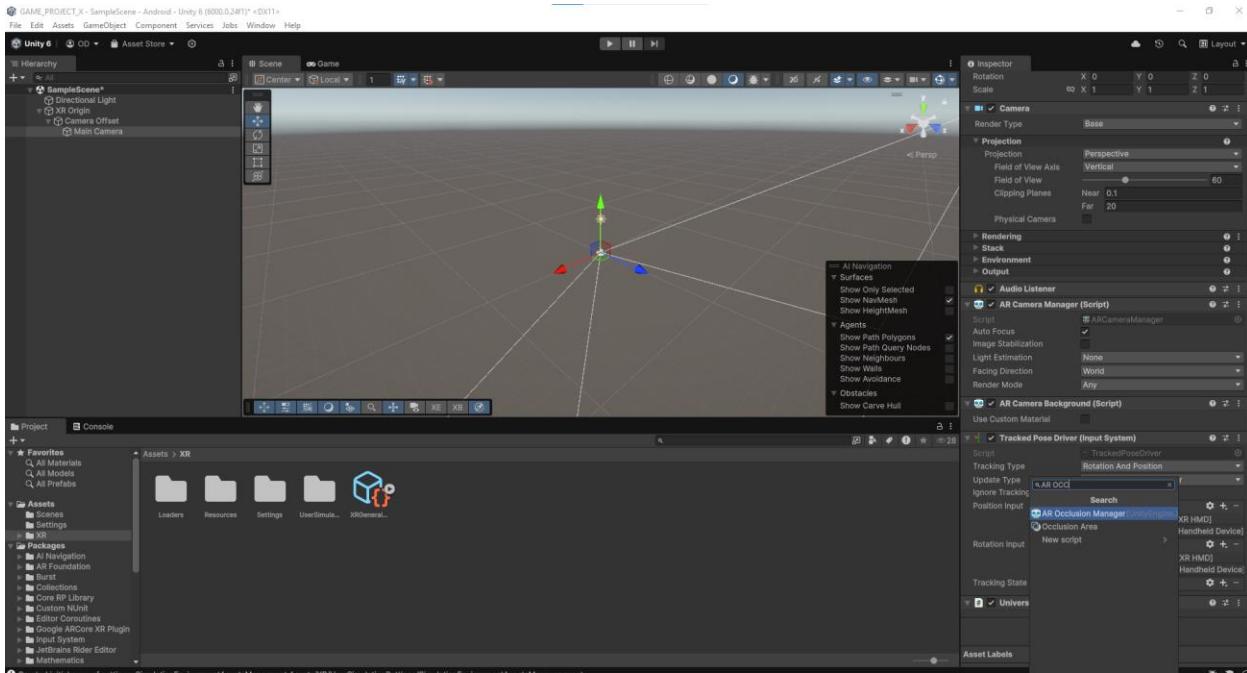
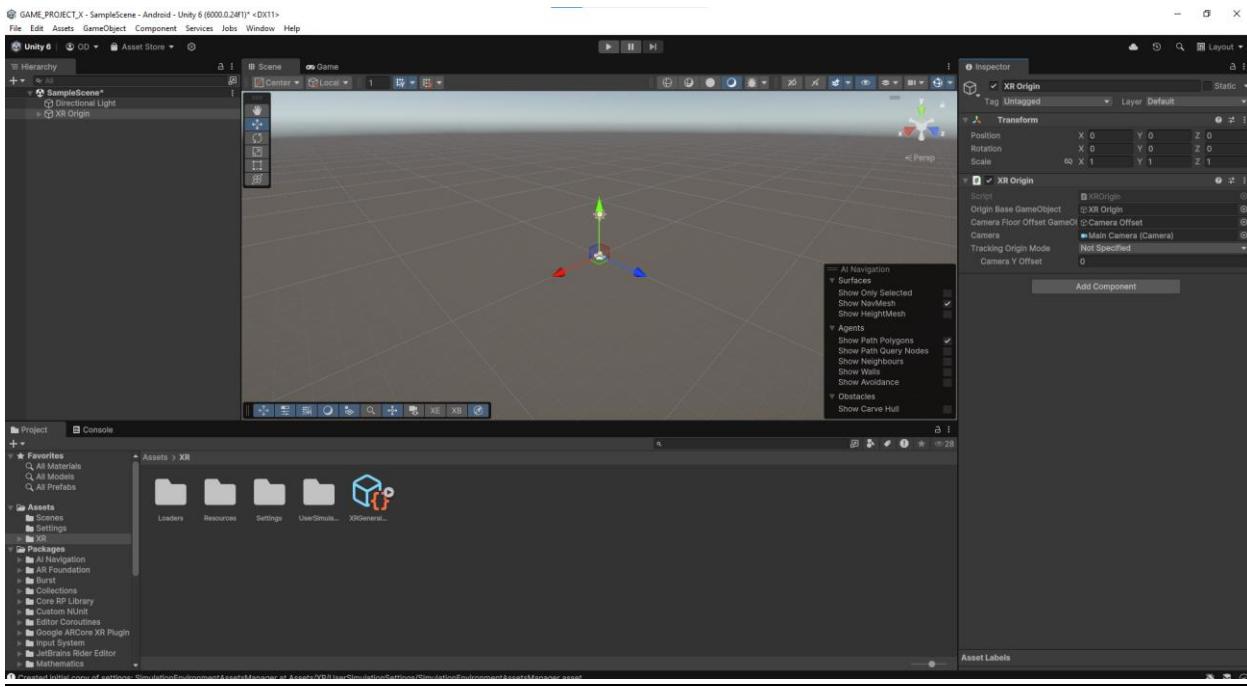
Setting up from ‘Not Specified’ to ‘Device’

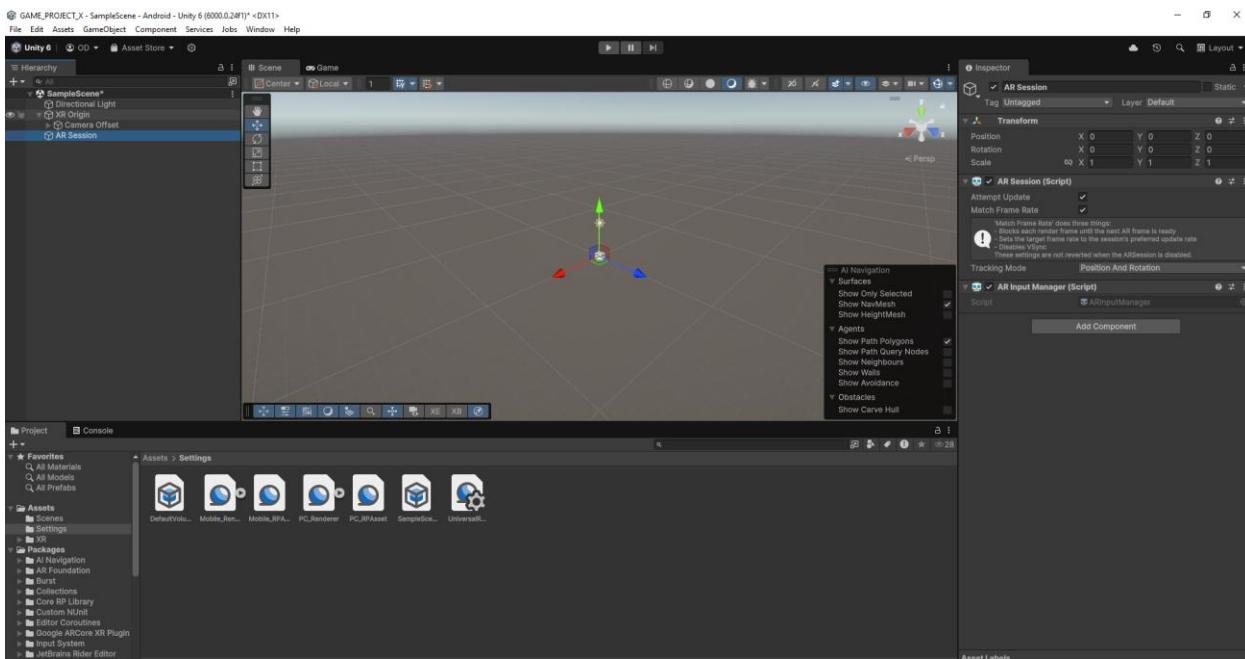
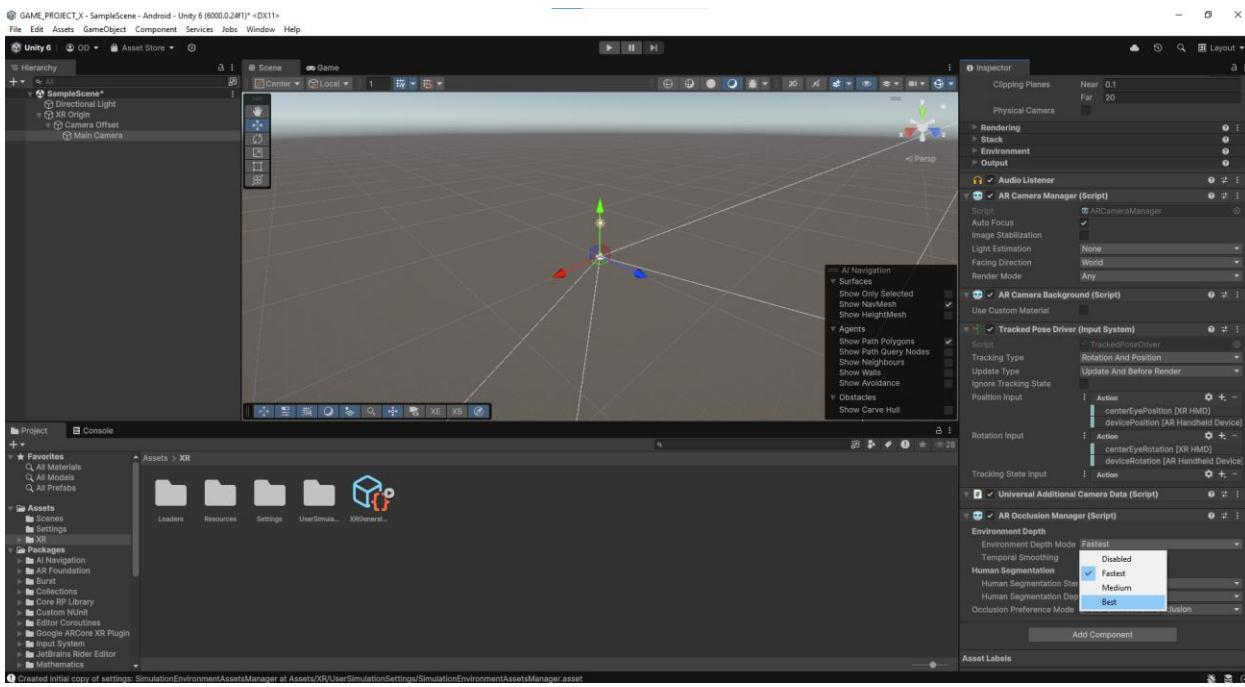
In the Unity editor, changing the setup specification from ‘Not Specified’ to ‘Device’ configures the environment for AR functionality. This setting ensures that Unity will use device-specific rendering and tracking options necessary for an AR application.

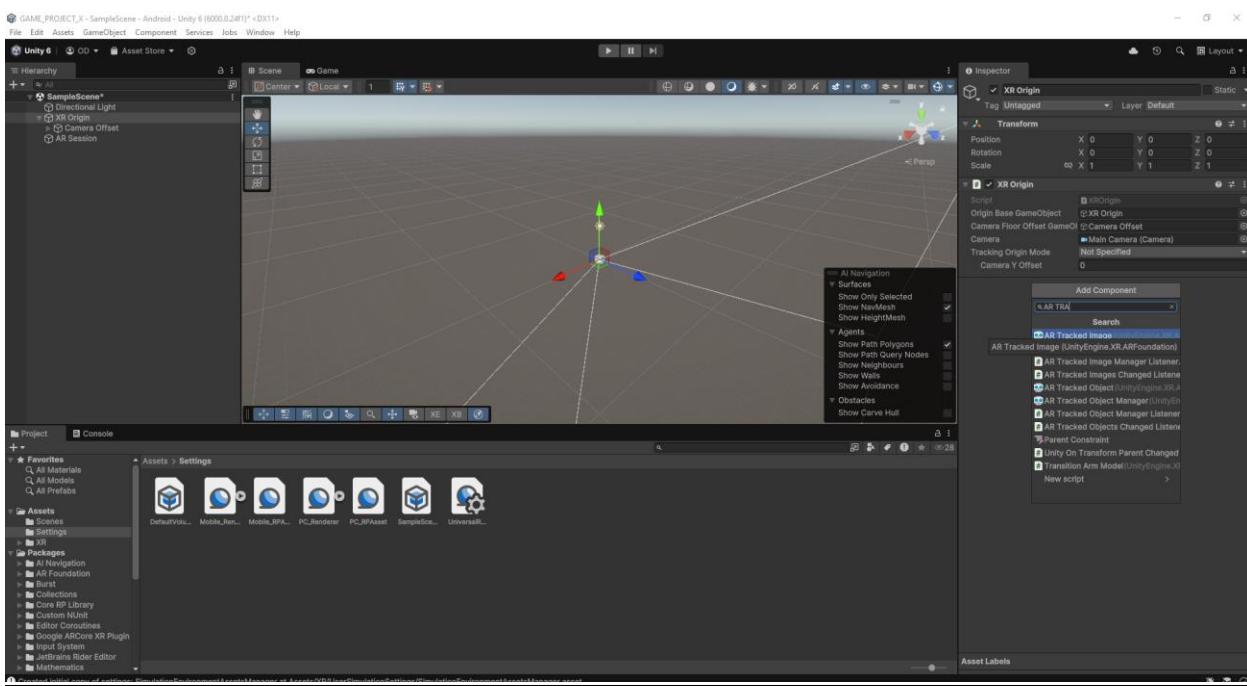


Adjust Camera Y Offset to ‘0’

Resetting the camera’s Y offset to zero aligns it with the ground level in AR, ensuring that all virtual elements, such as markers and pathways, are grounded accurately in relation to real-world objects.



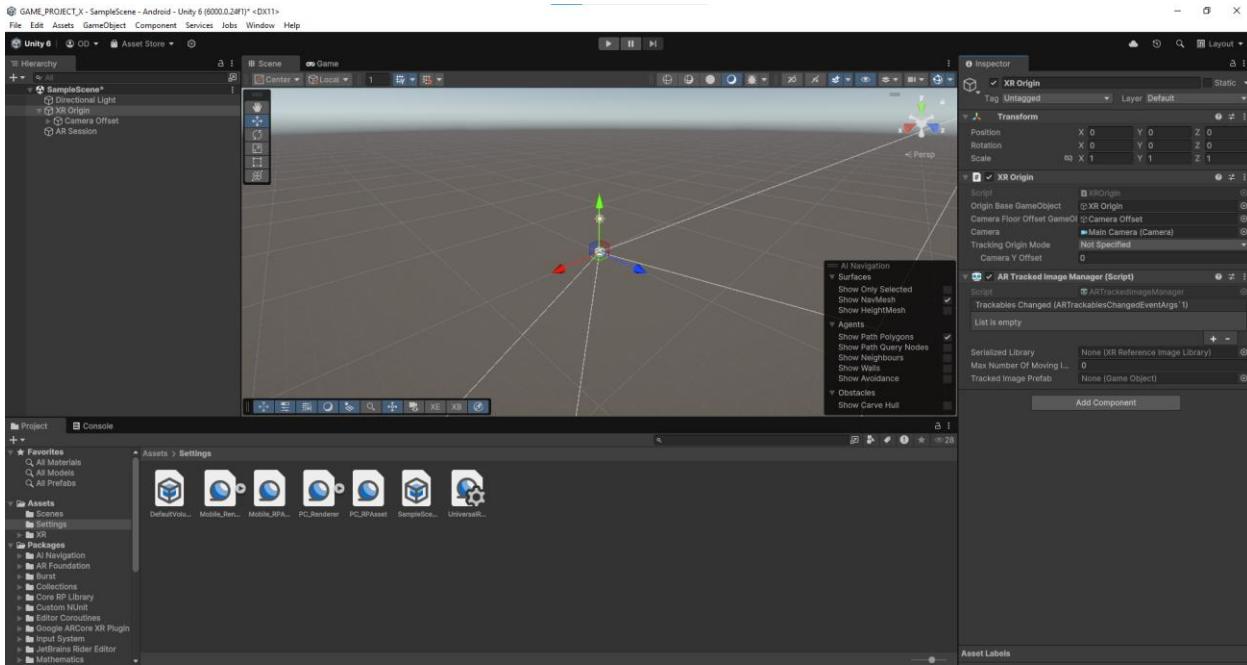


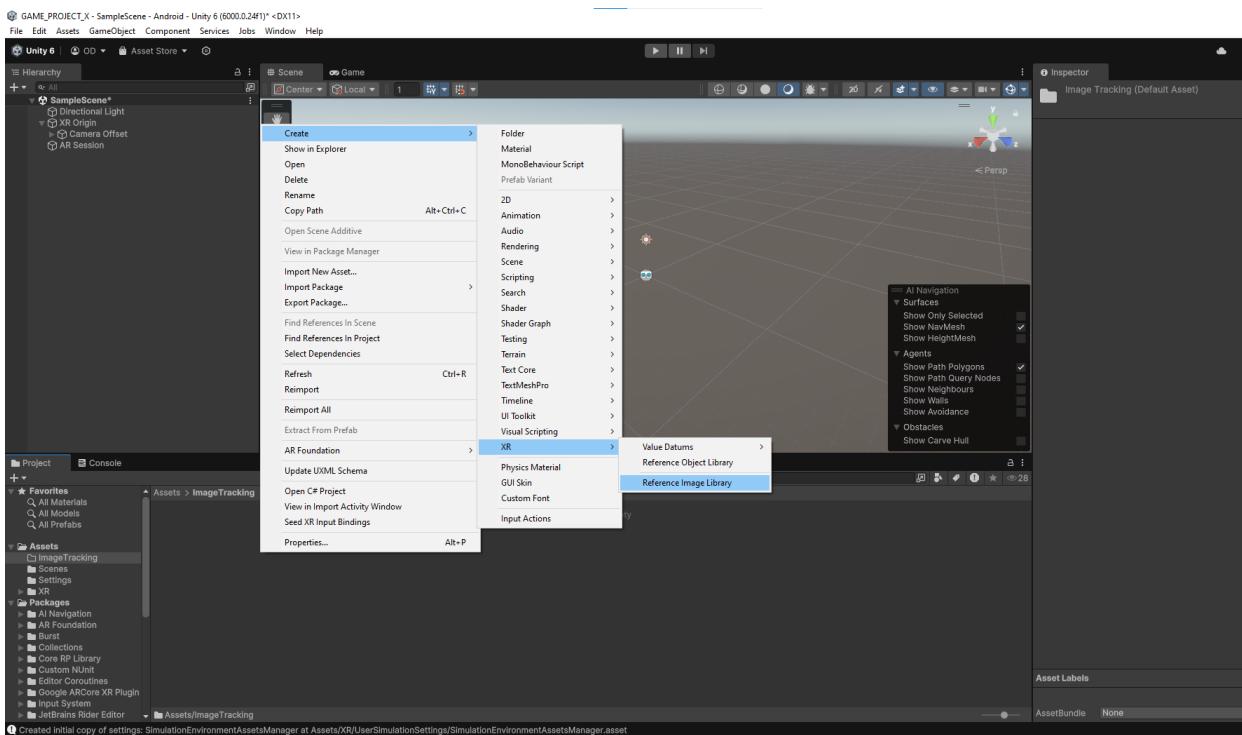
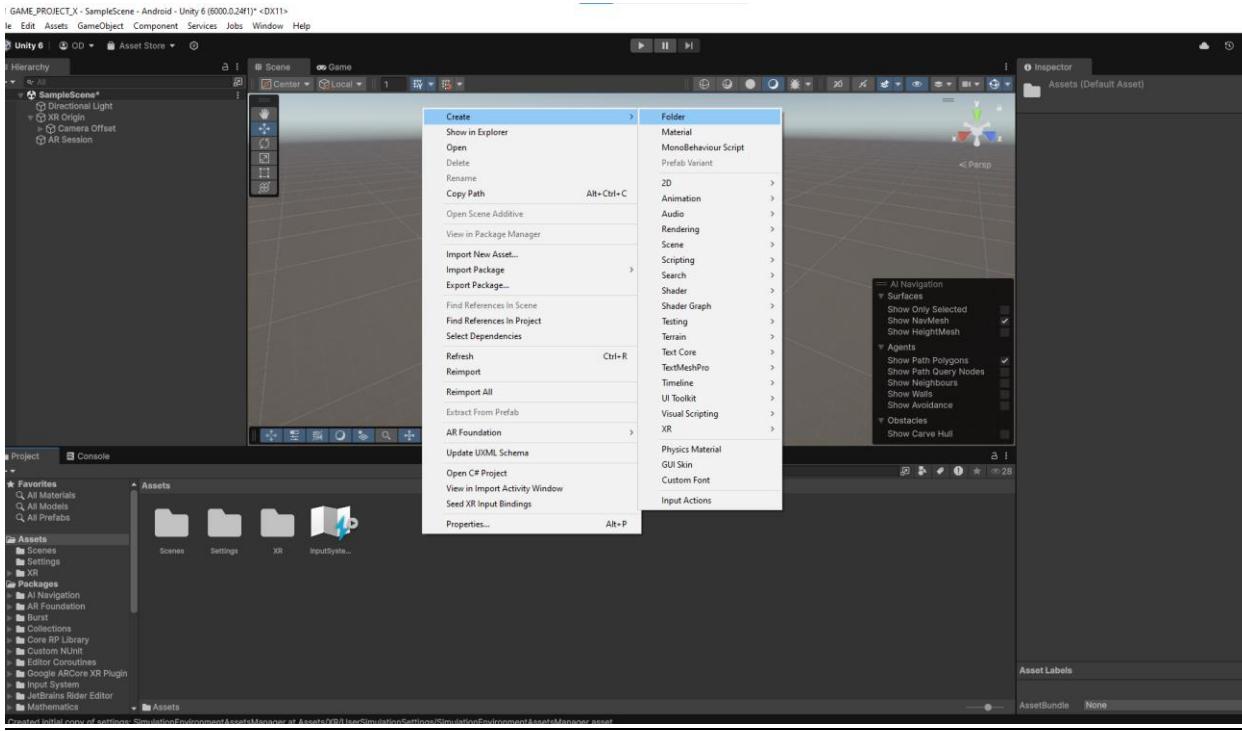


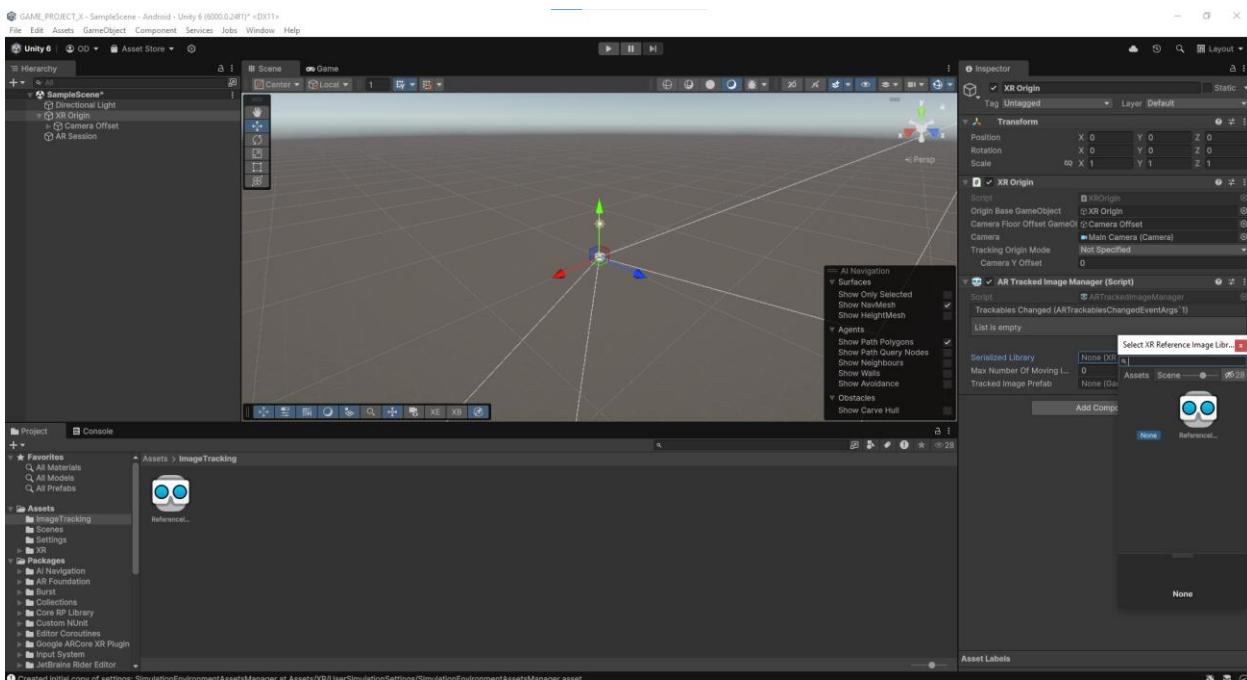
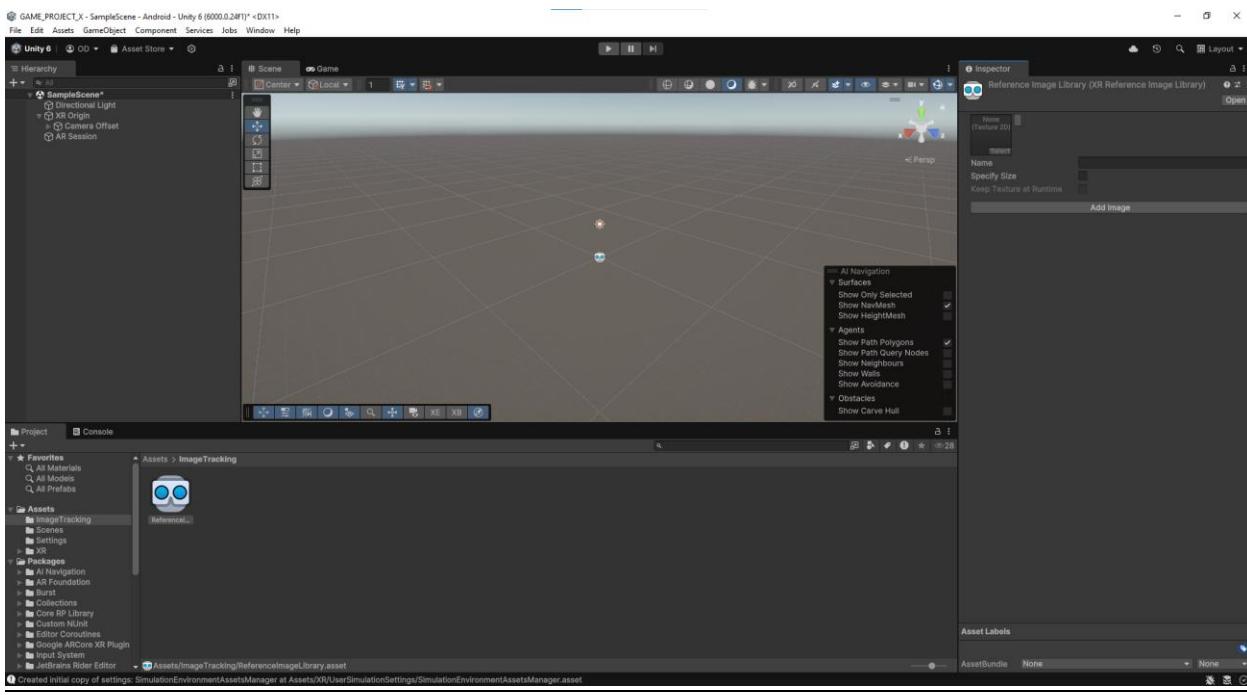
AR Image Tracking

1. Add AR Tracked Image Manager under AR Session

The AR Tracked Image Manager component is added to handle image-based tracking, allowing the application to recognize specific images and associate them with virtual objects or instructions. This is a crucial component for dynamic navigation cues.

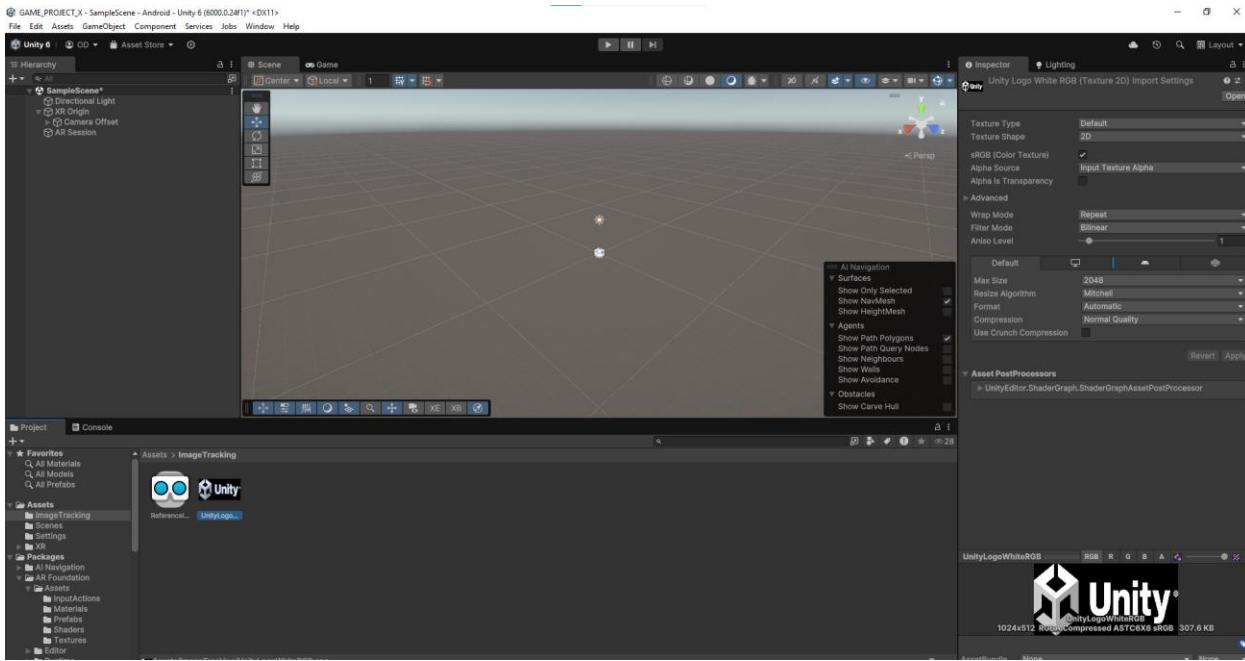
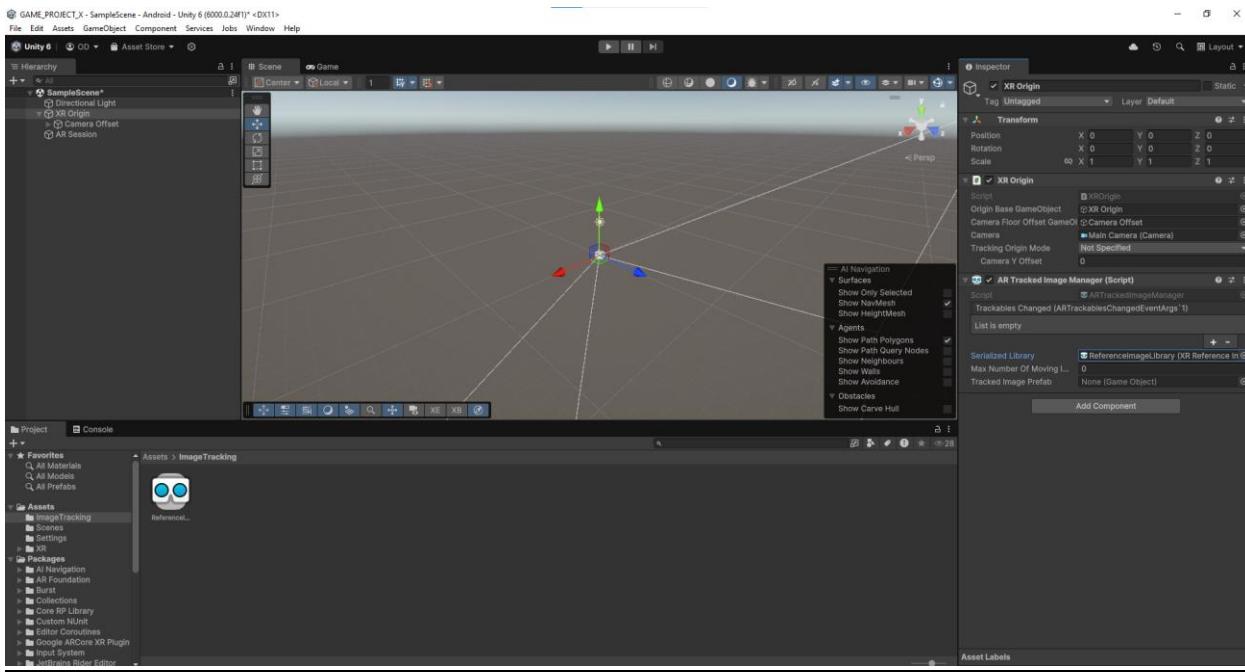


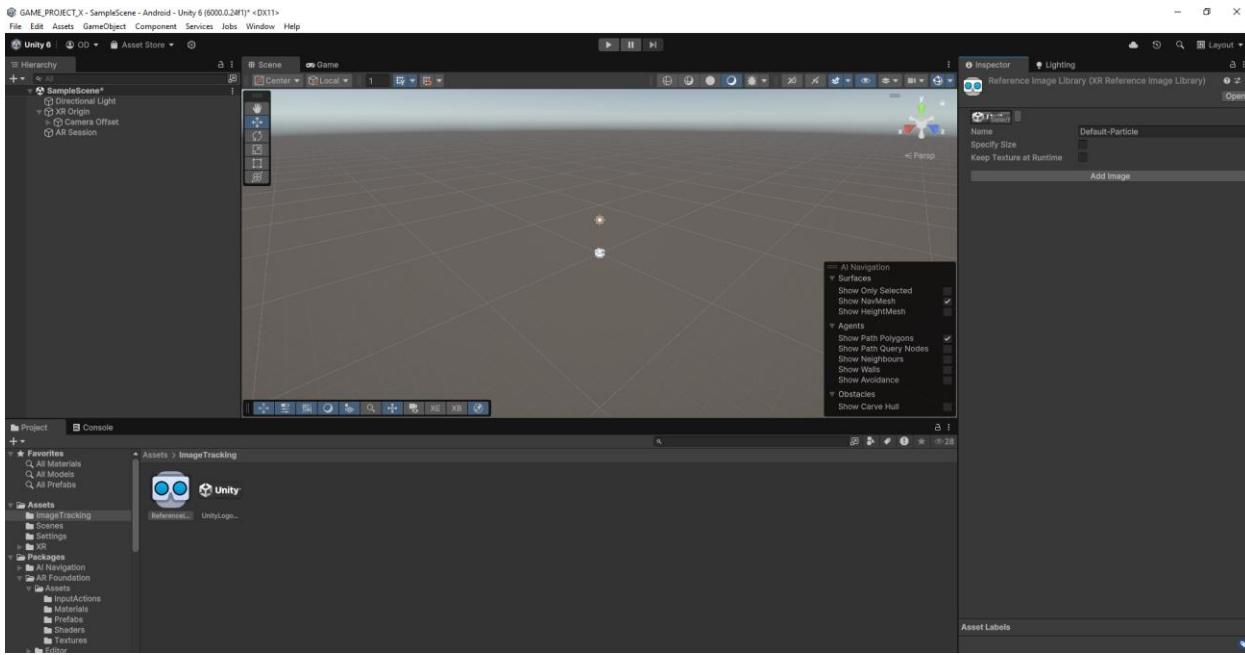
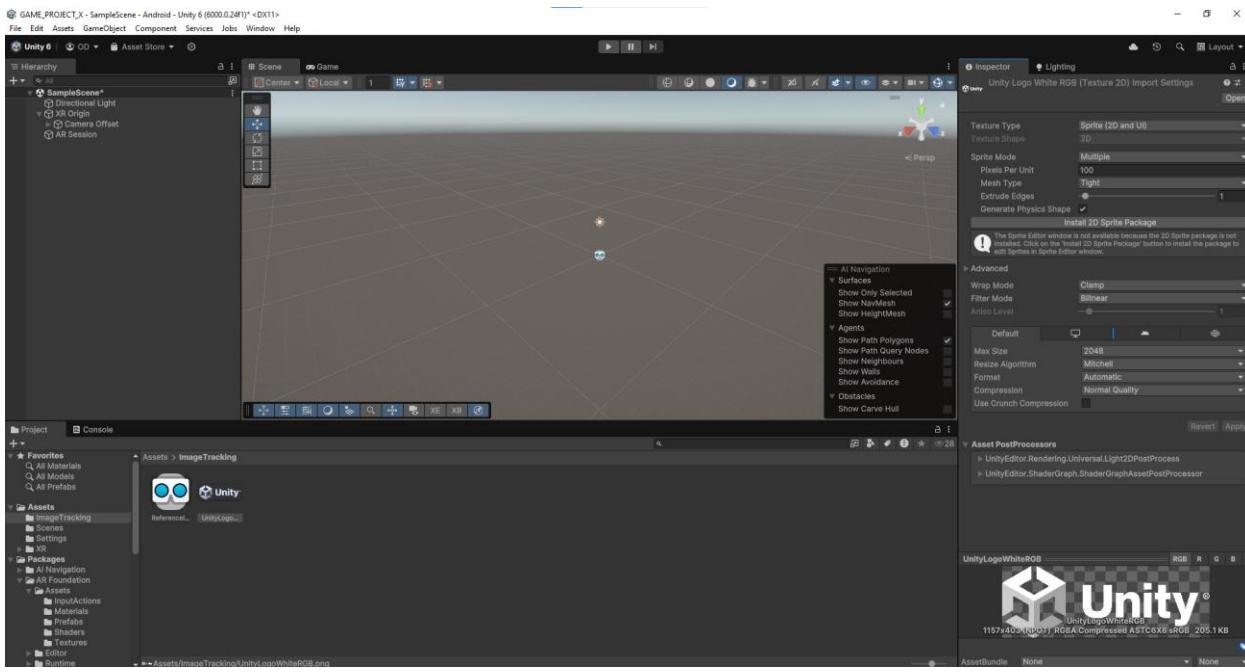


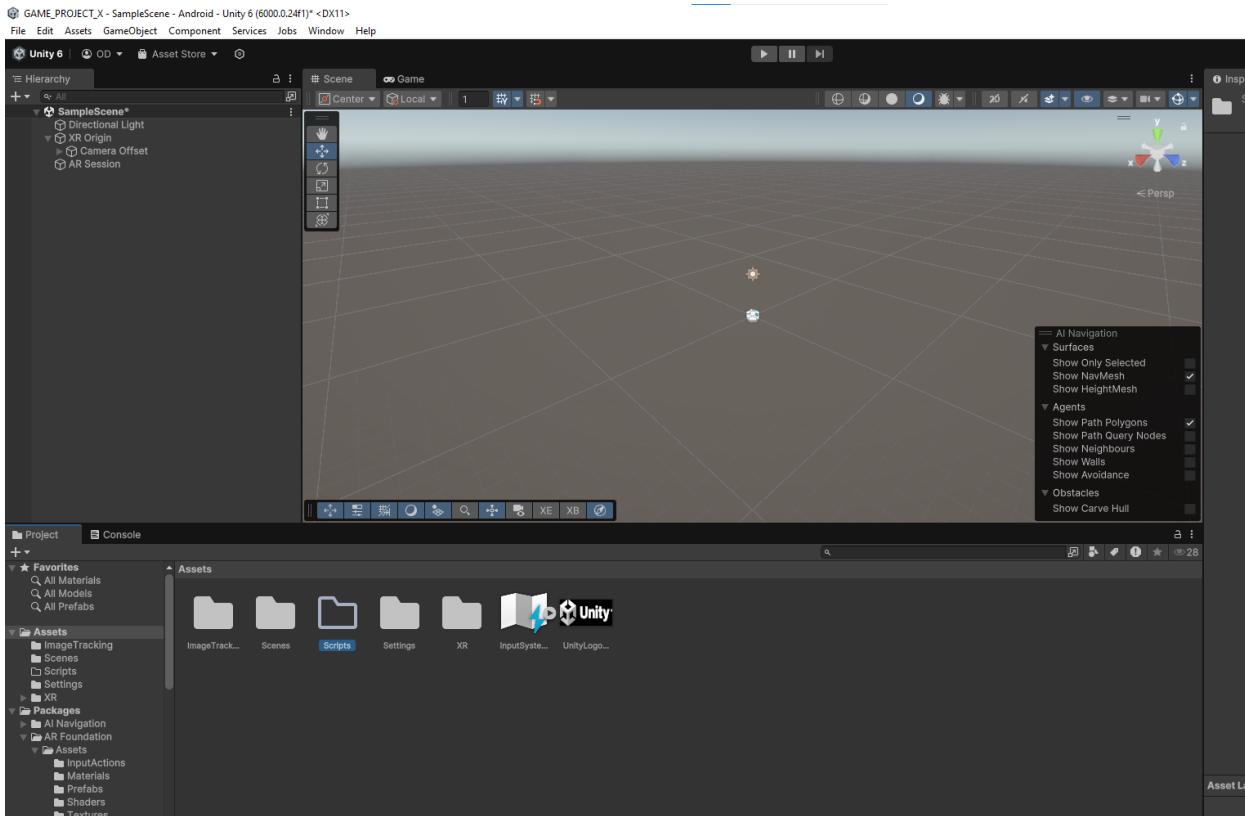
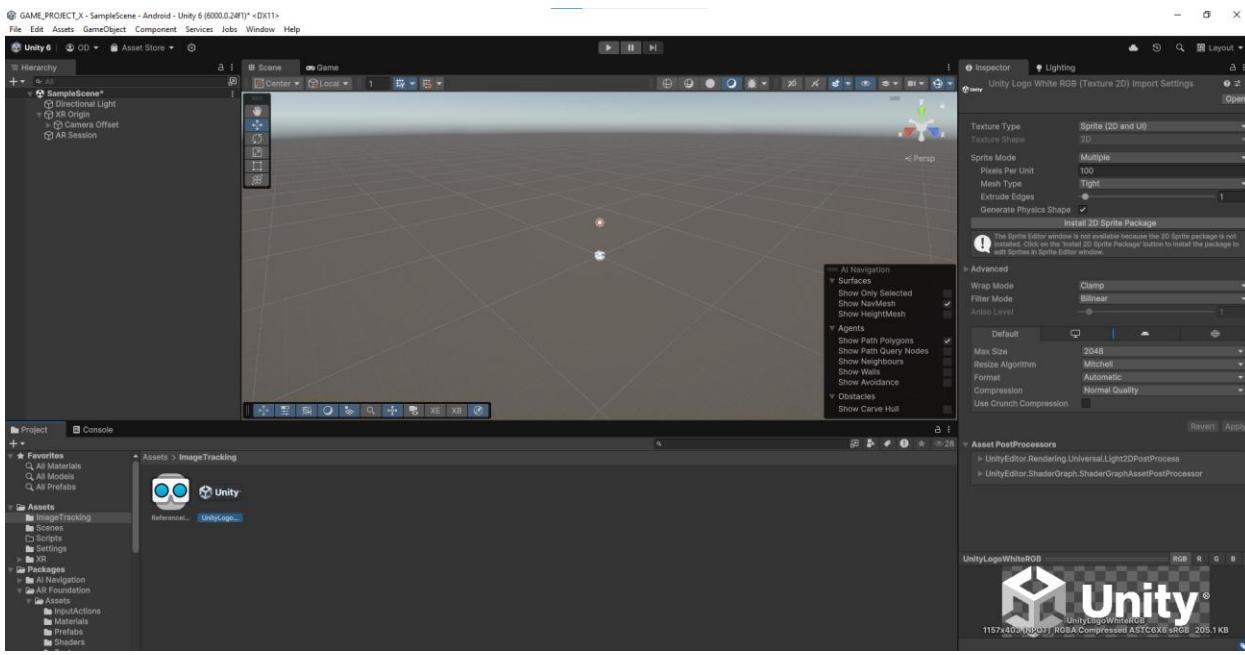


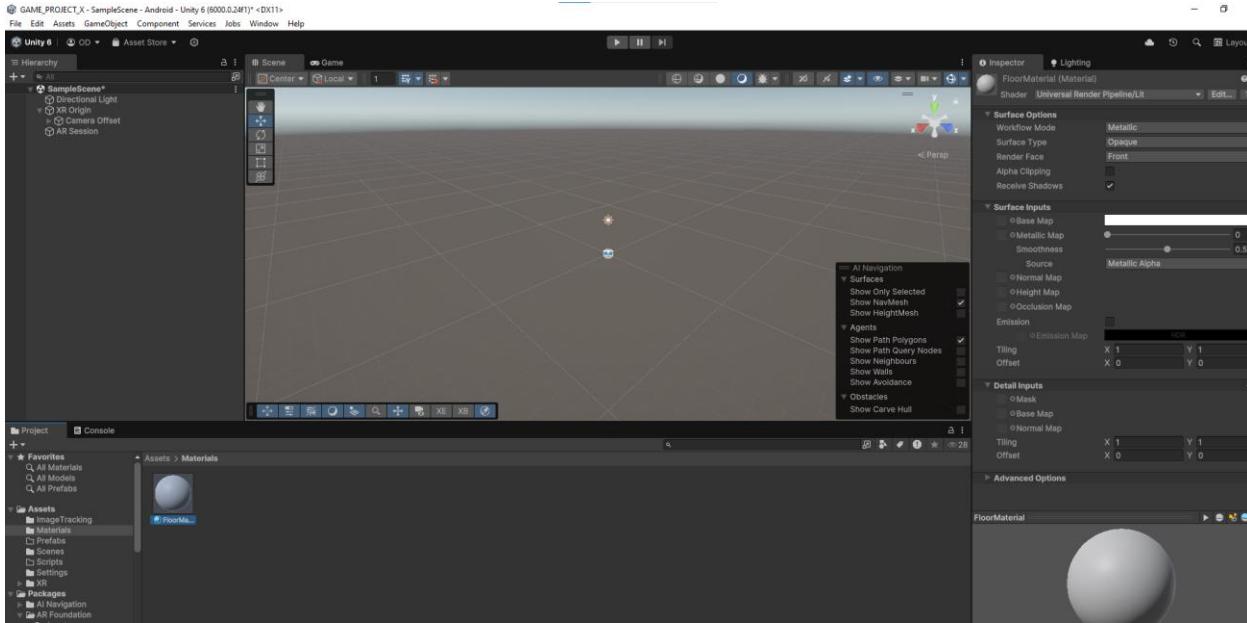
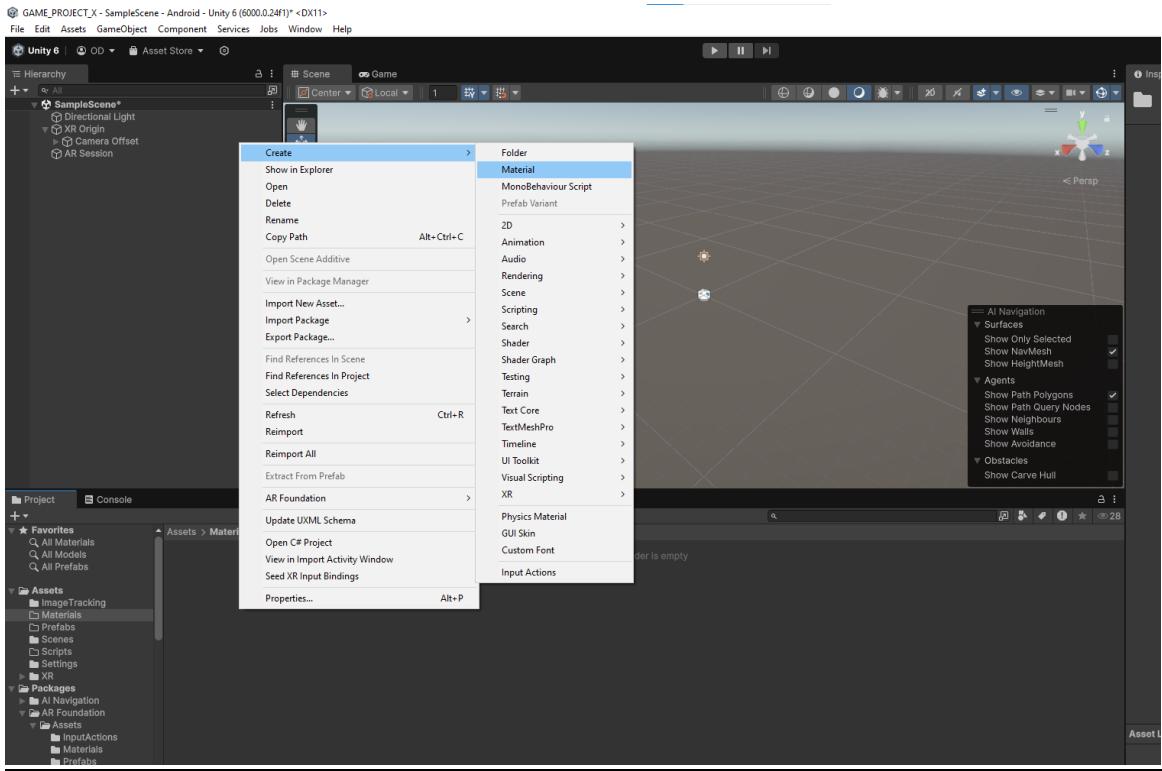
Select Serialized Library as the Reference Image Library

Assign the previously created reference image library under XR Origin. This library contains images that the application will recognize and track, triggering corresponding actions, such as showing directions or pulling up relevant information about locations on campus.







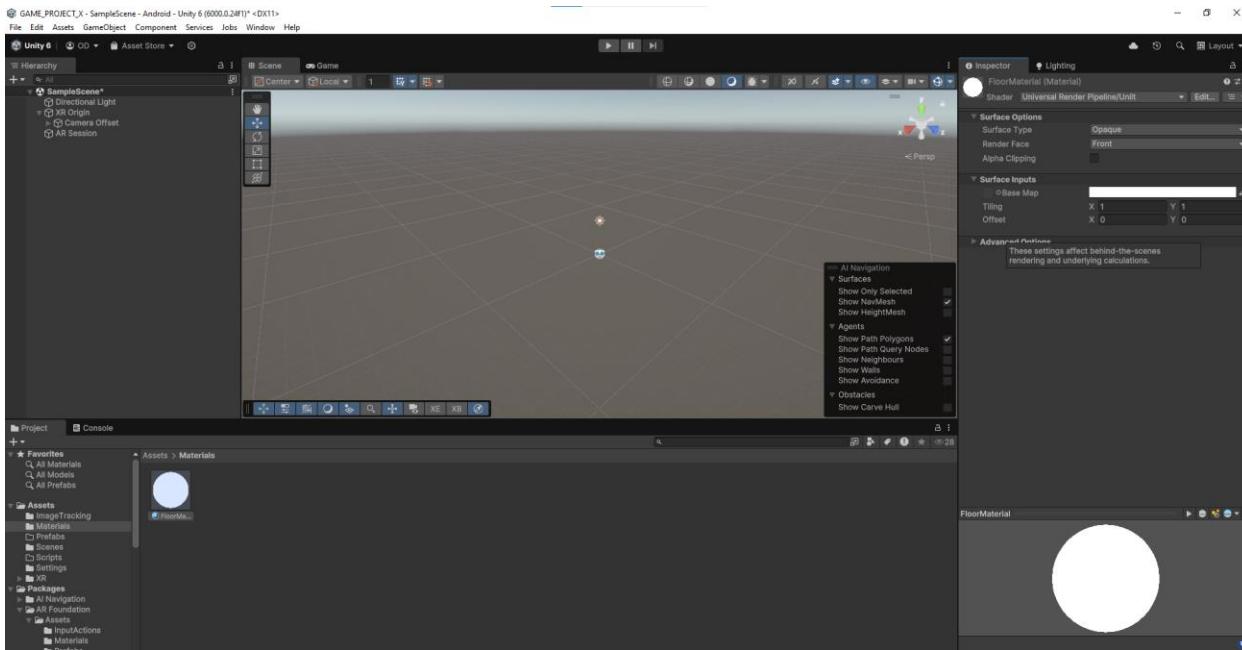
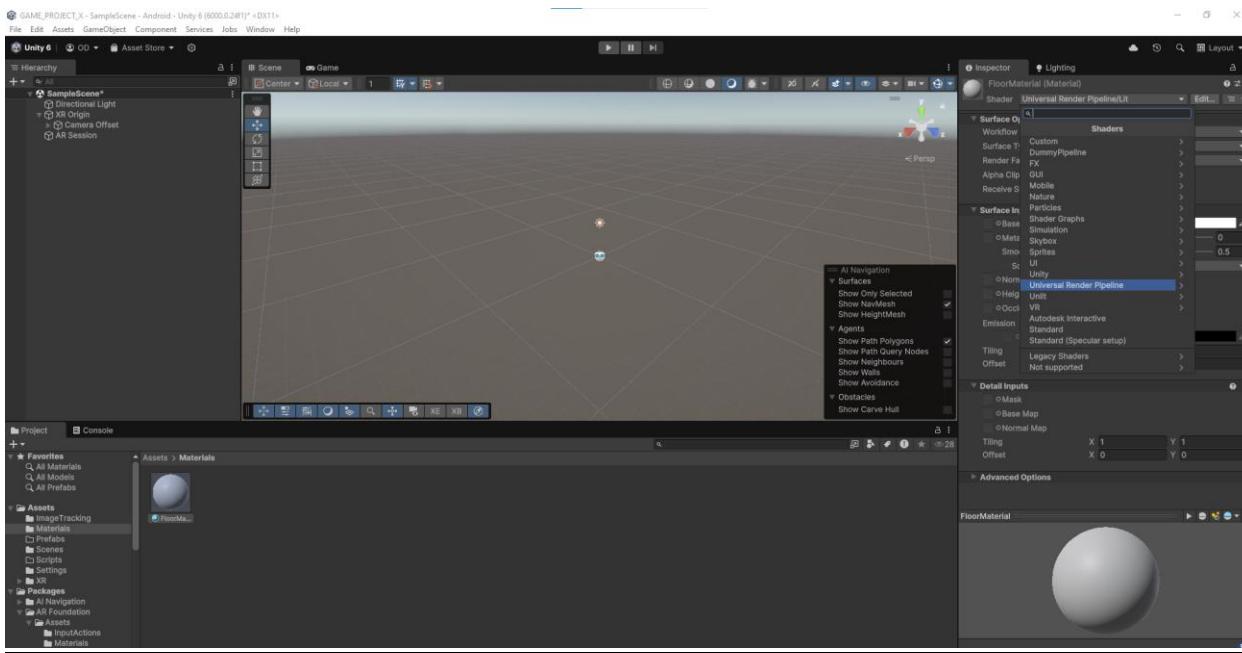


Graphics Settings

1. Change Material to Universal Render Pipeline/Unlit

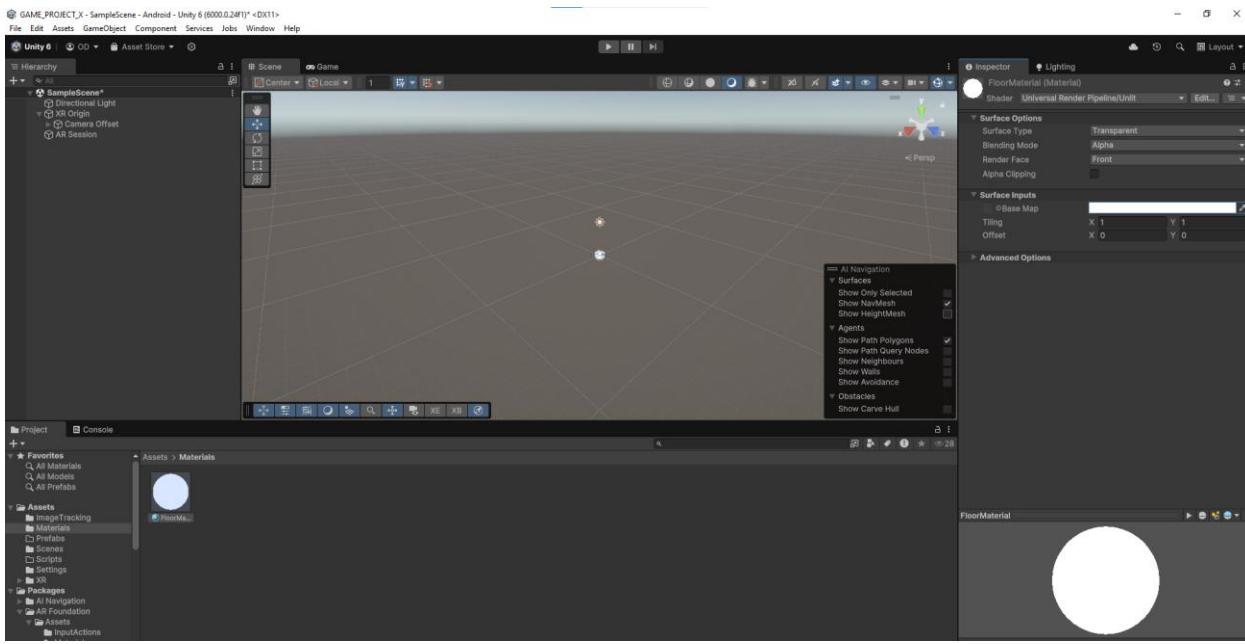
To optimize graphics for an AR experience, change the material's shader to the Unlit variant under the Universal Render Pipeline. This ensures that the AR objects remain visible and do not interact with lighting conditions in unintended ways.

Universal Render Pipeline/Unlit -> Universal Render Pipeline -> Unlit



Set	Surface	Type	to	Transparent
-----	---------	------	----	-------------

Changing the surface type to transparent helps to overlay digital elements over the real-world feed without obscuring it, allowing for a clearer AR experience where objects or markers are visible without obstructing the user's view of the campus.



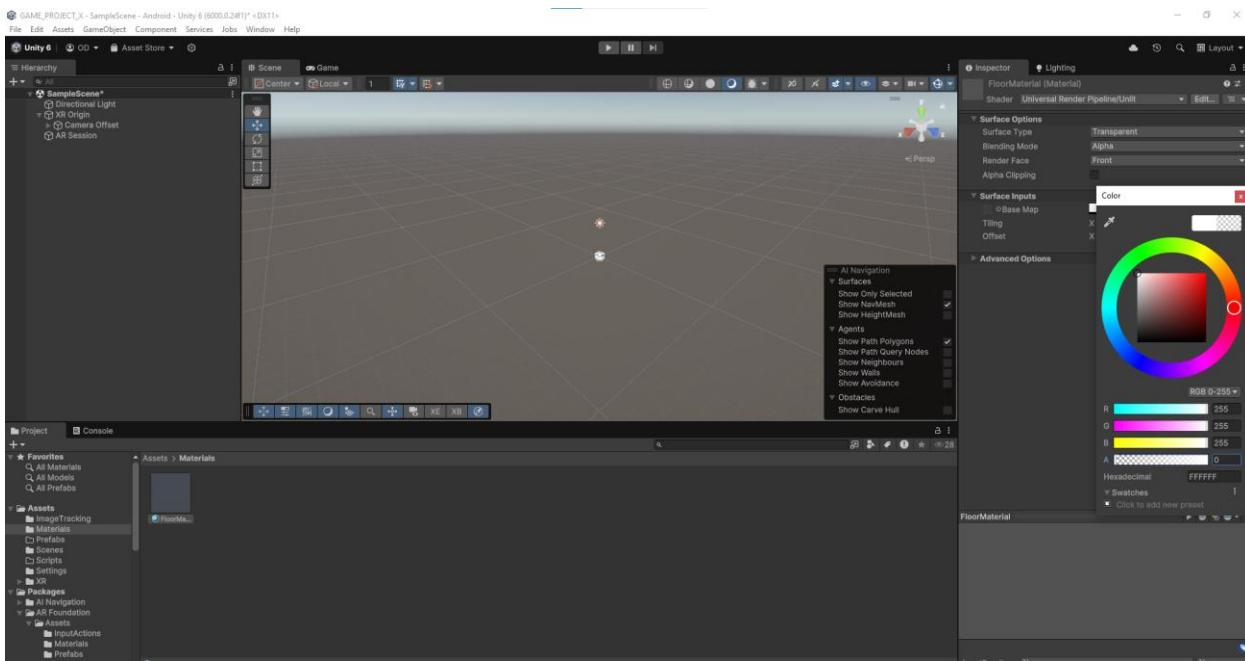
Adjust

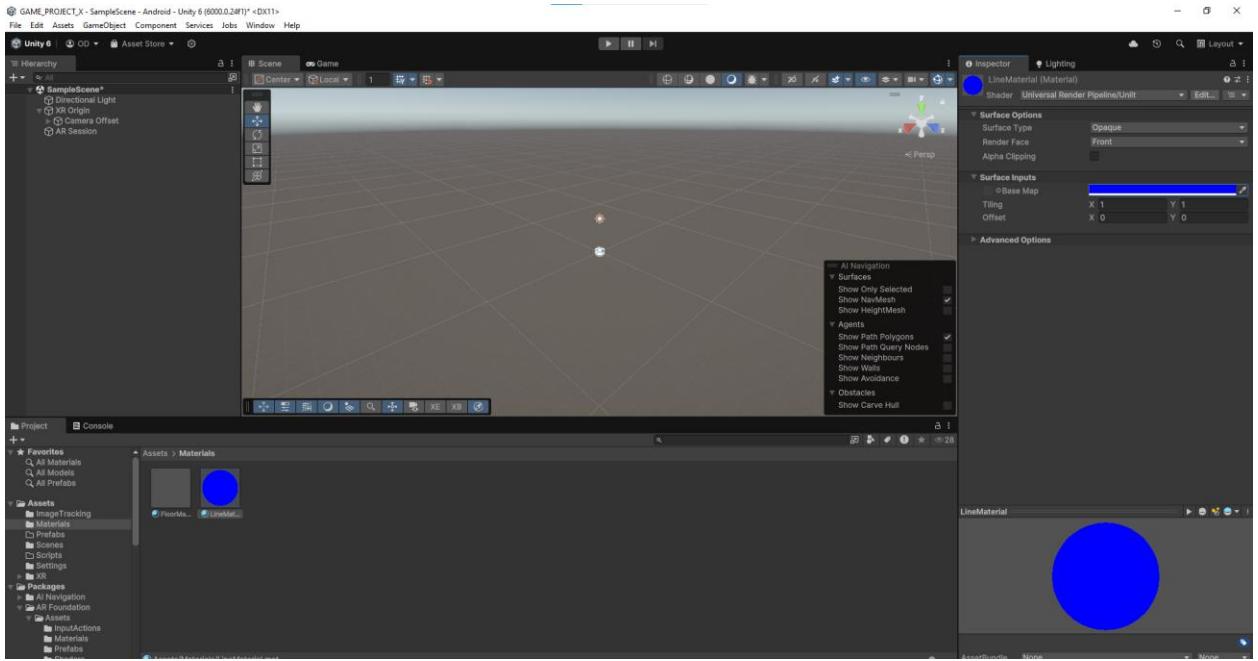
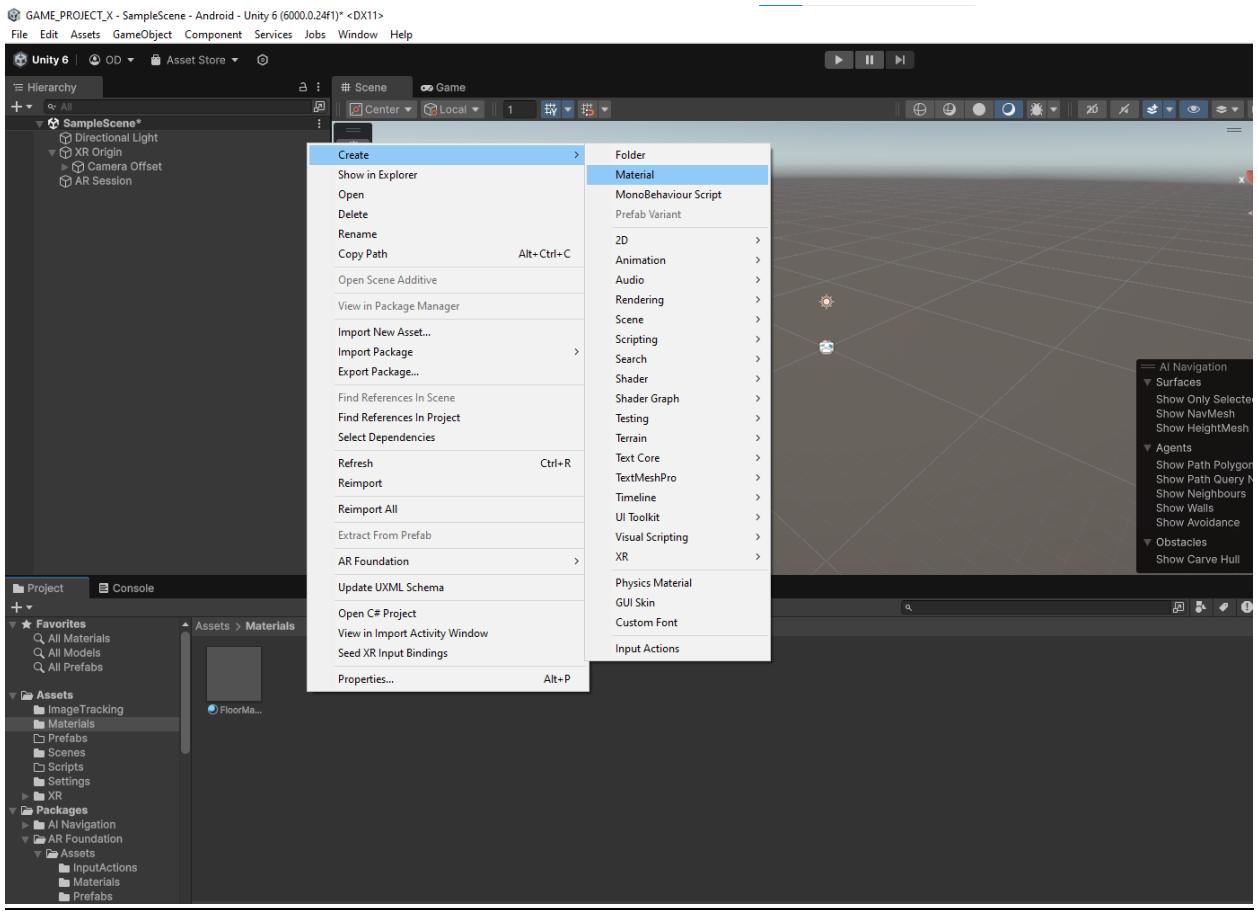
Base

Map

Intensity

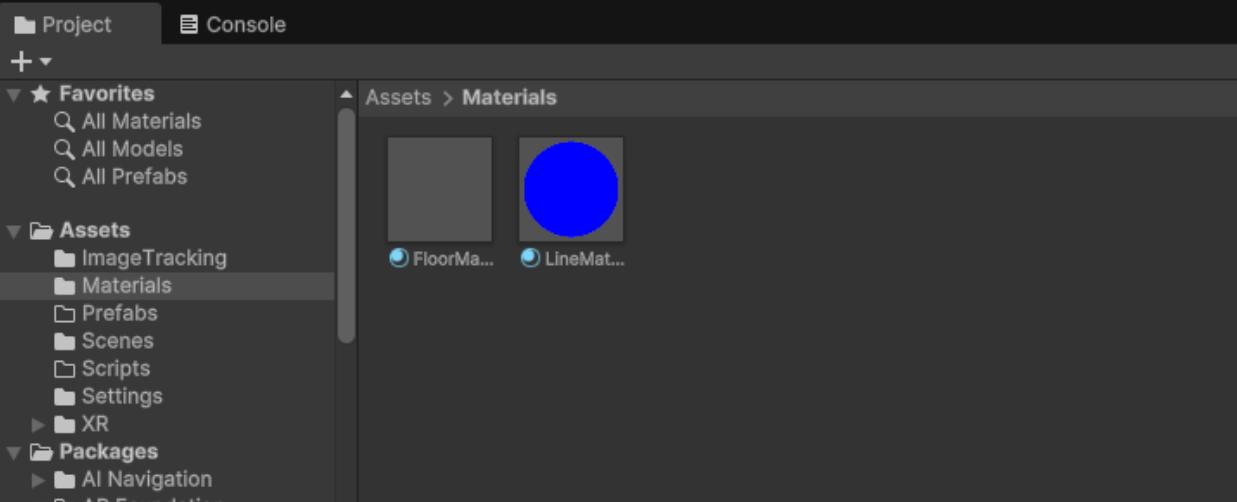
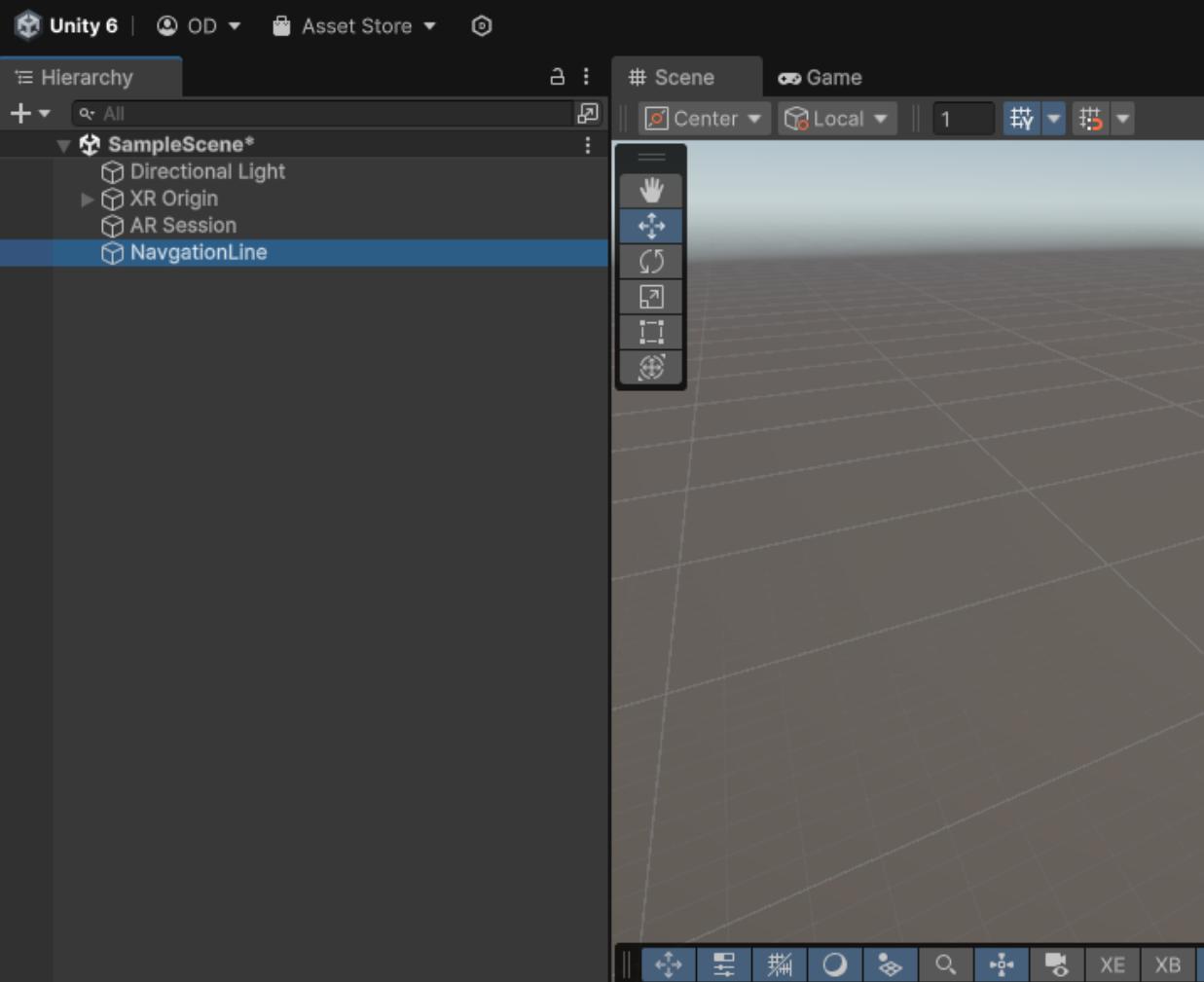
By modifying the Base Map intensity from 255 to 0, the opacity and brightness levels can be controlled, making digital elements blend naturally with real-world surroundings while being easily distinguishable.

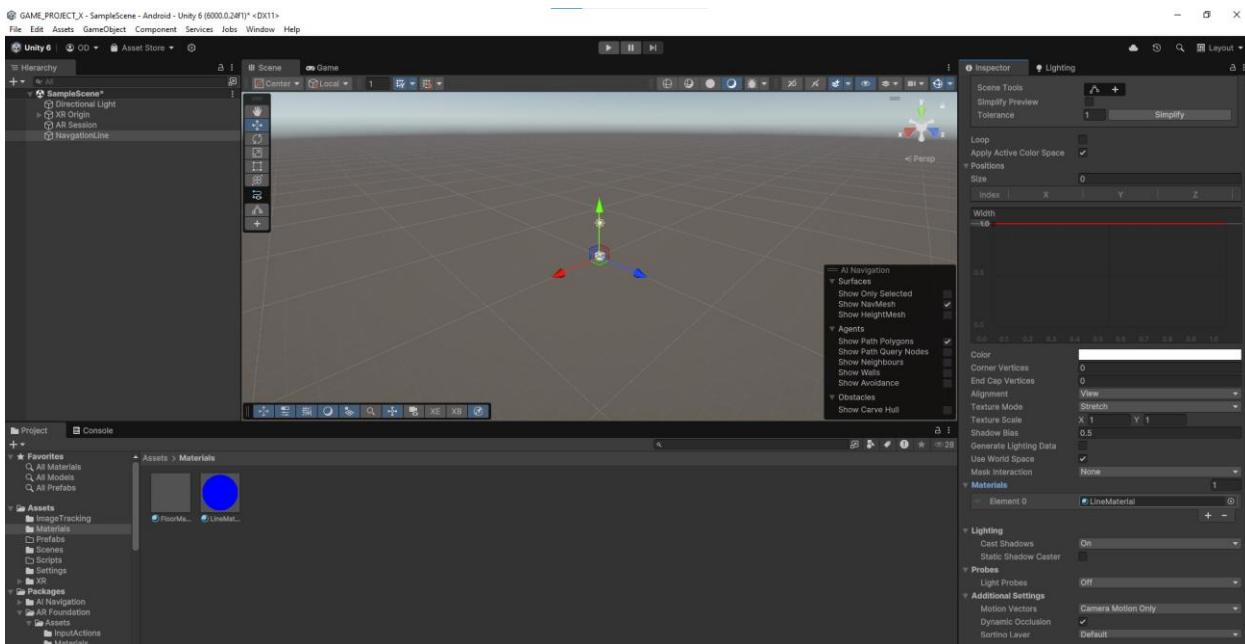
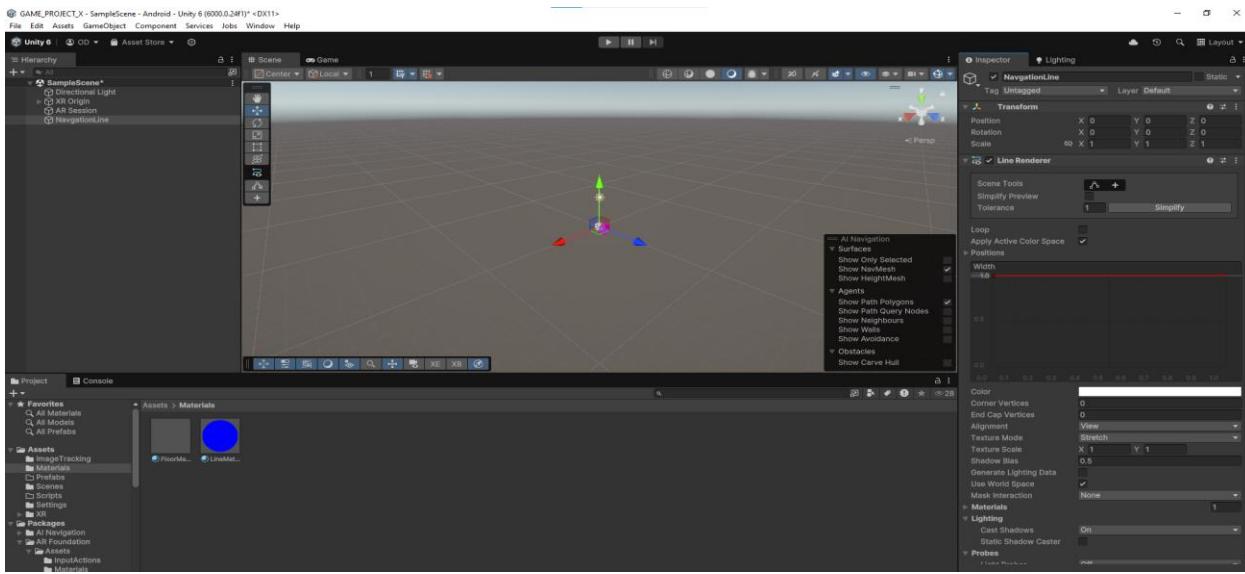
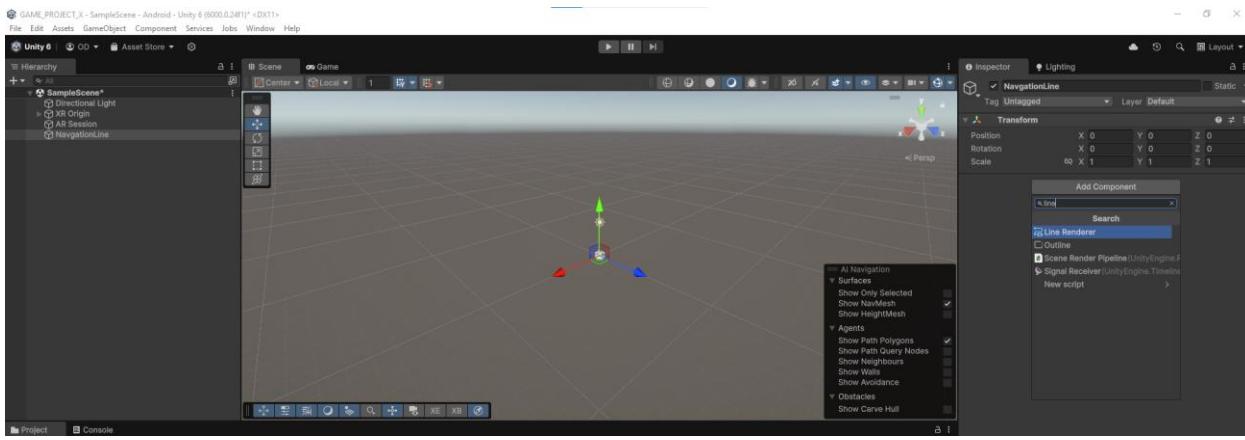




GAME_PROJECT_X - SampleScene - Android - Unity 6 (6000.0.24f1)* <DX11>

File Edit Assets GameObject Component Services Jobs Window Help

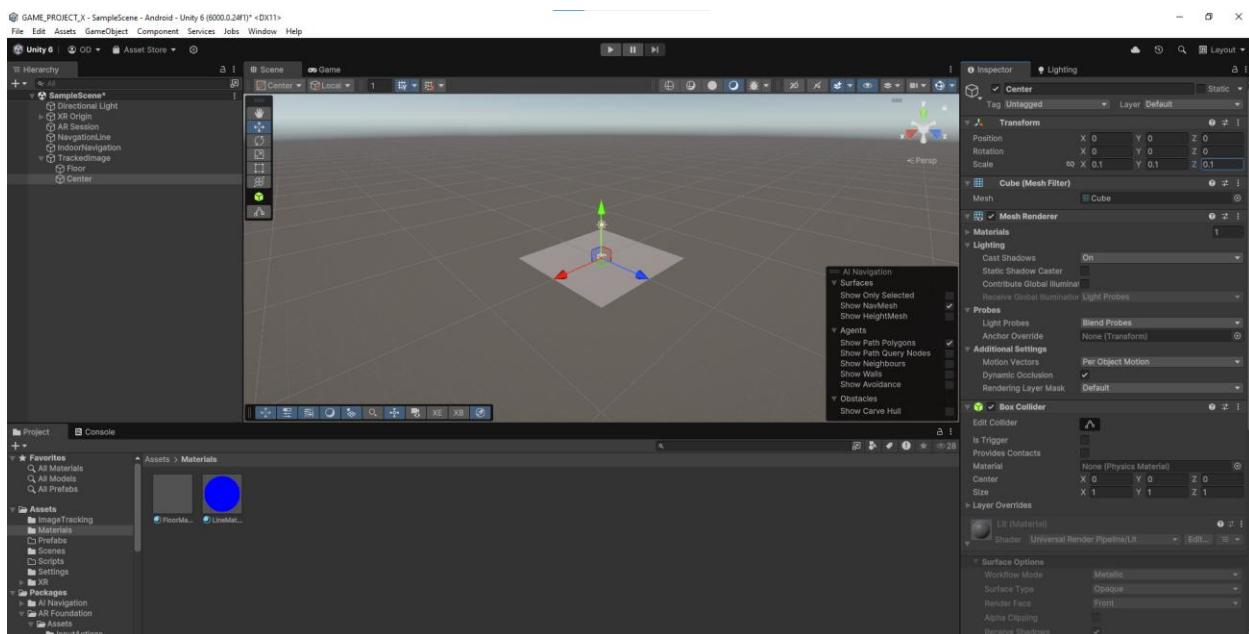
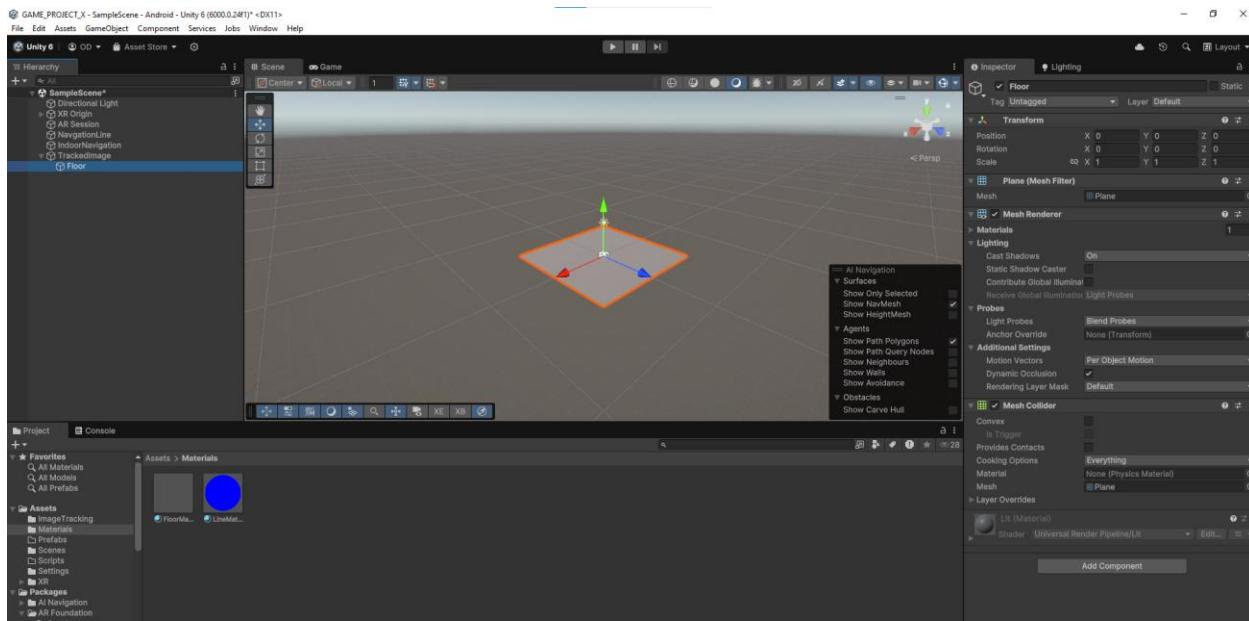


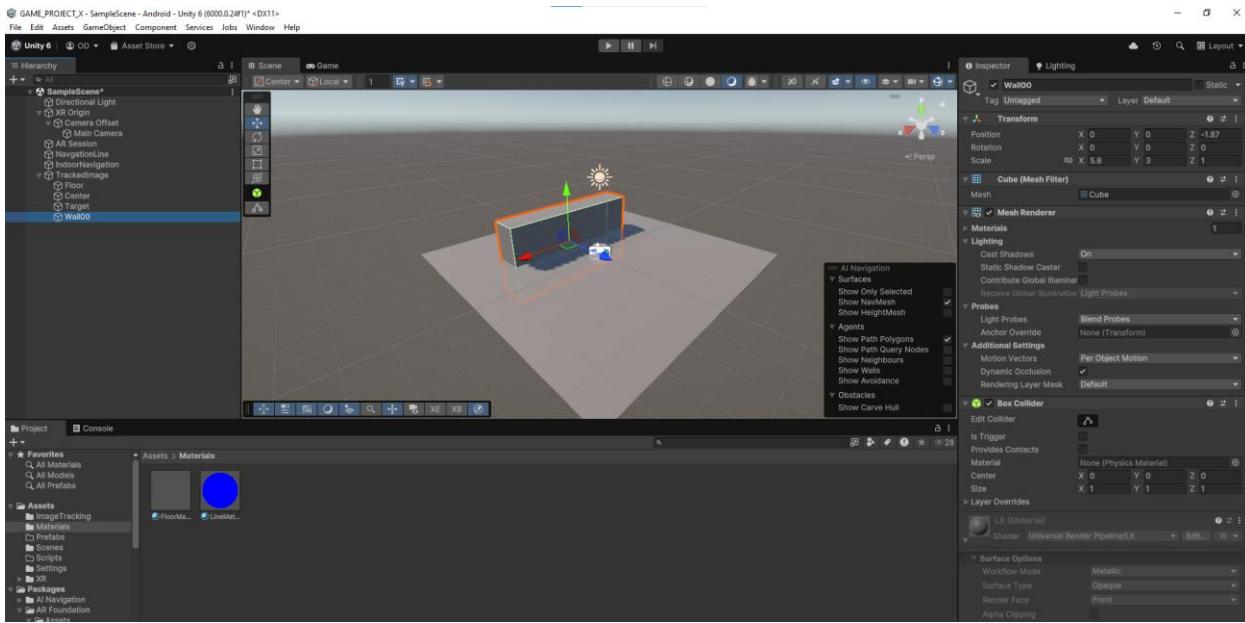
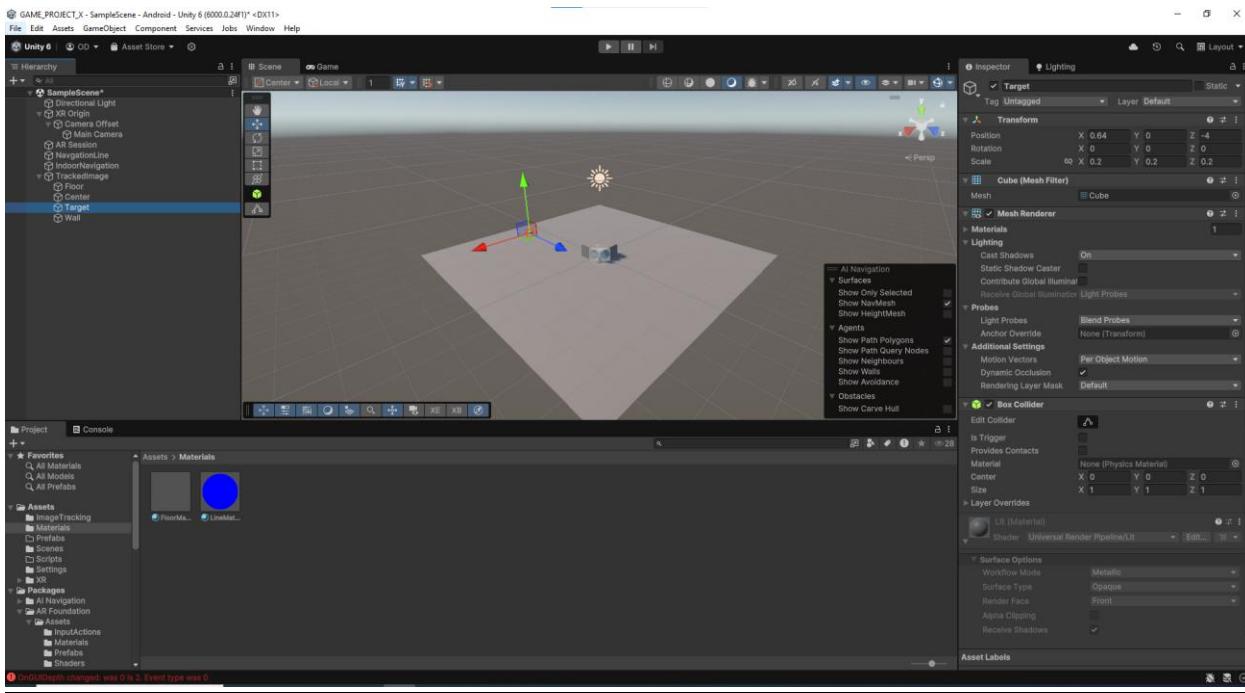


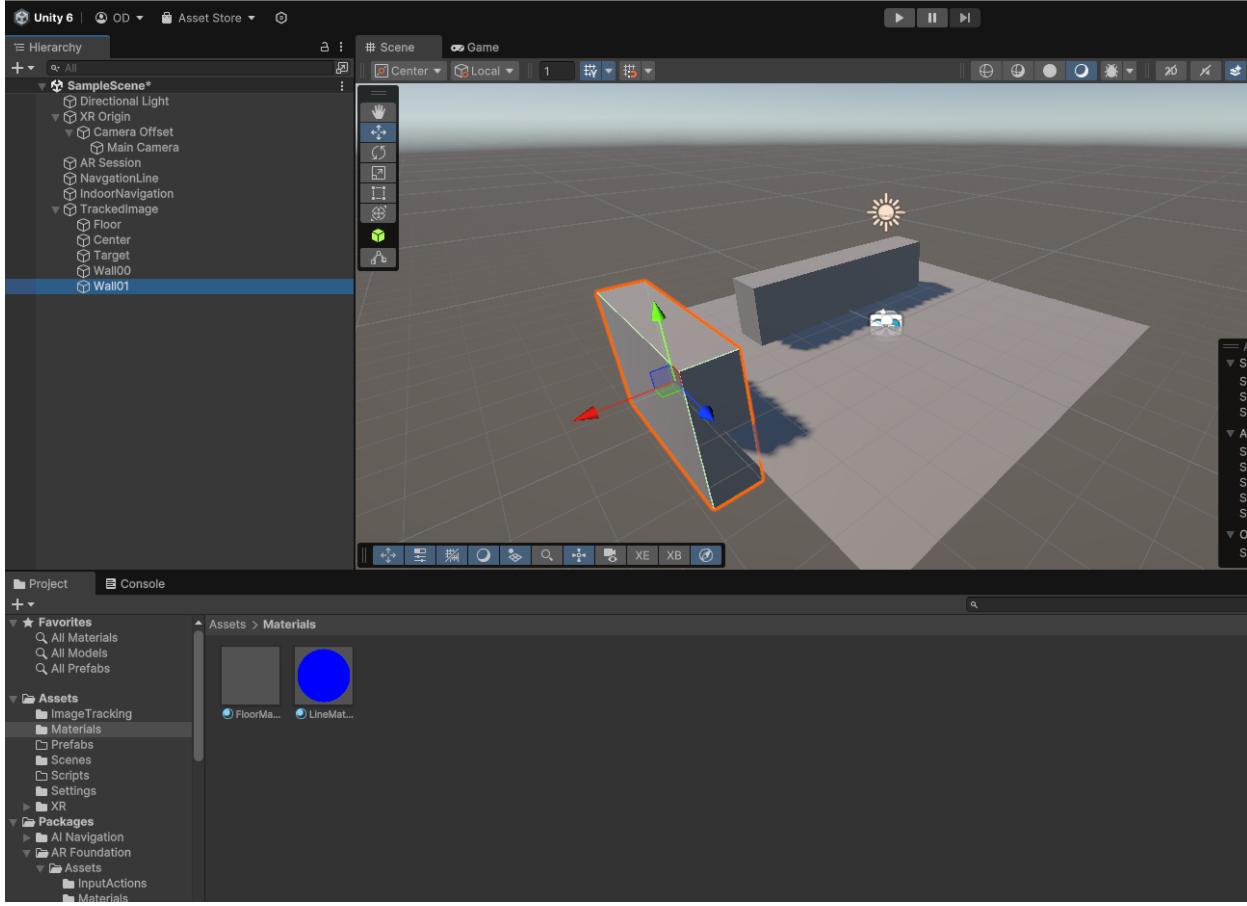
Scene Configuration

- Arrange Objects in the Scene (Walls, Targets, Center)**
- Each object, such as walls and navigation targets, is positioned accurately within the virtual campus model.

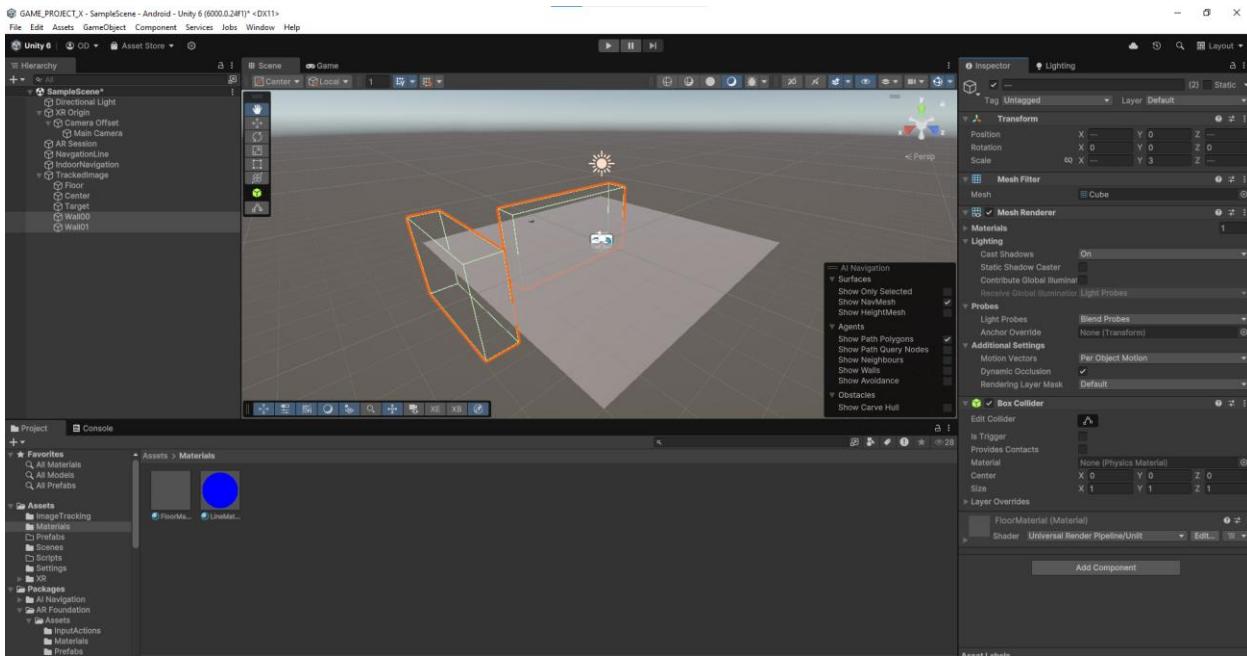
This includes walls, targets, and other elements like paths that guide users through the AR navigation interface.

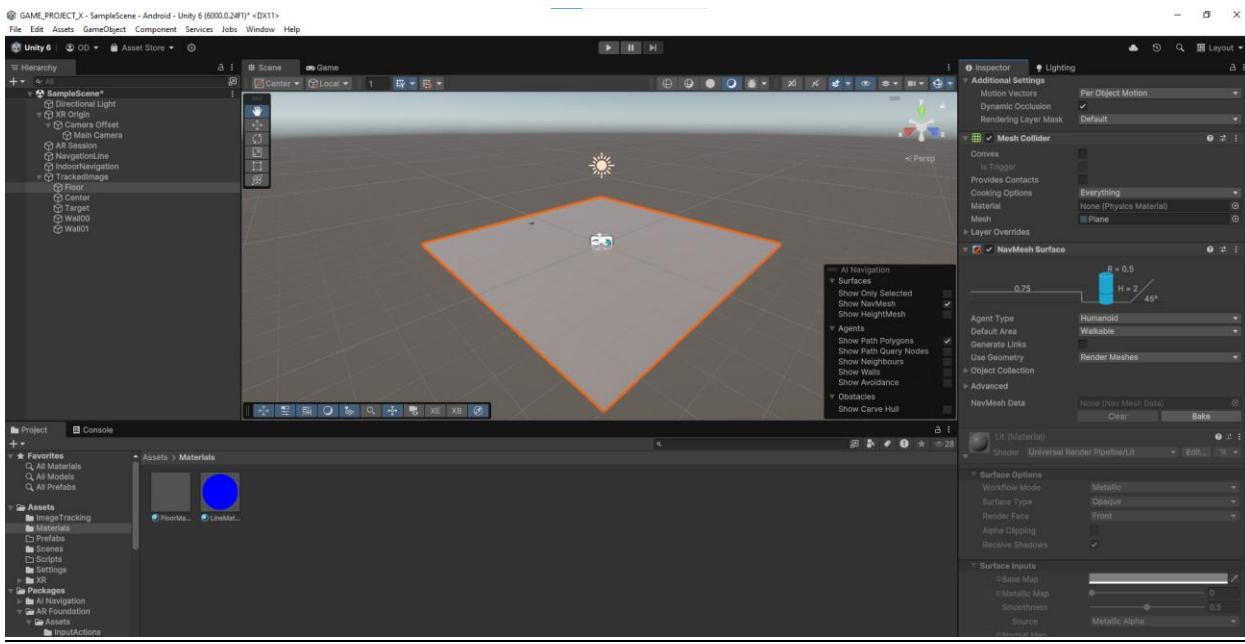






Rearranging co-ordinates of all objects accordingly in the scene of Wall00,Wall01 and other cubes respectively.



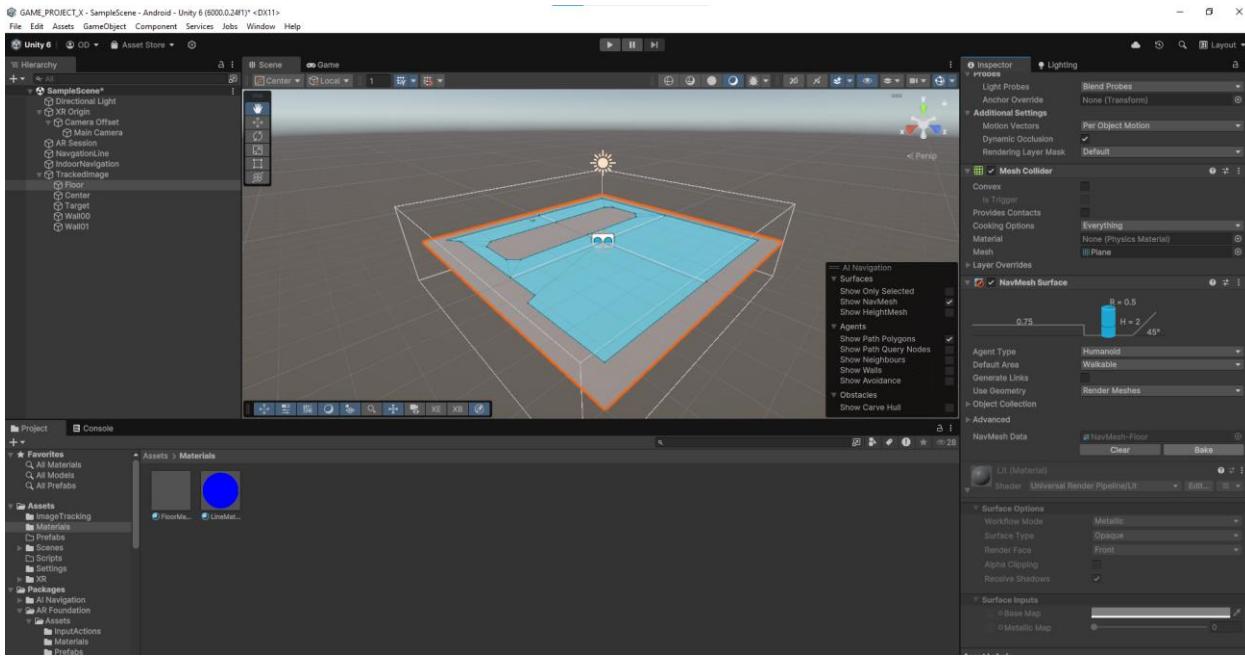


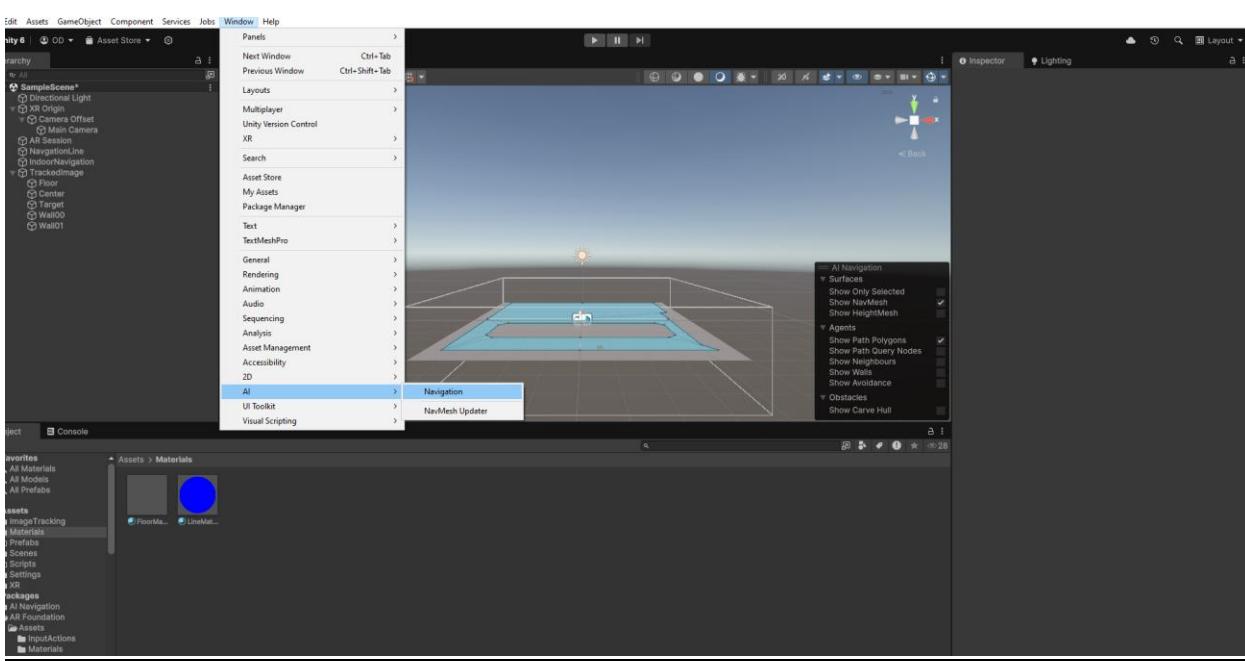
Bake

Lighting

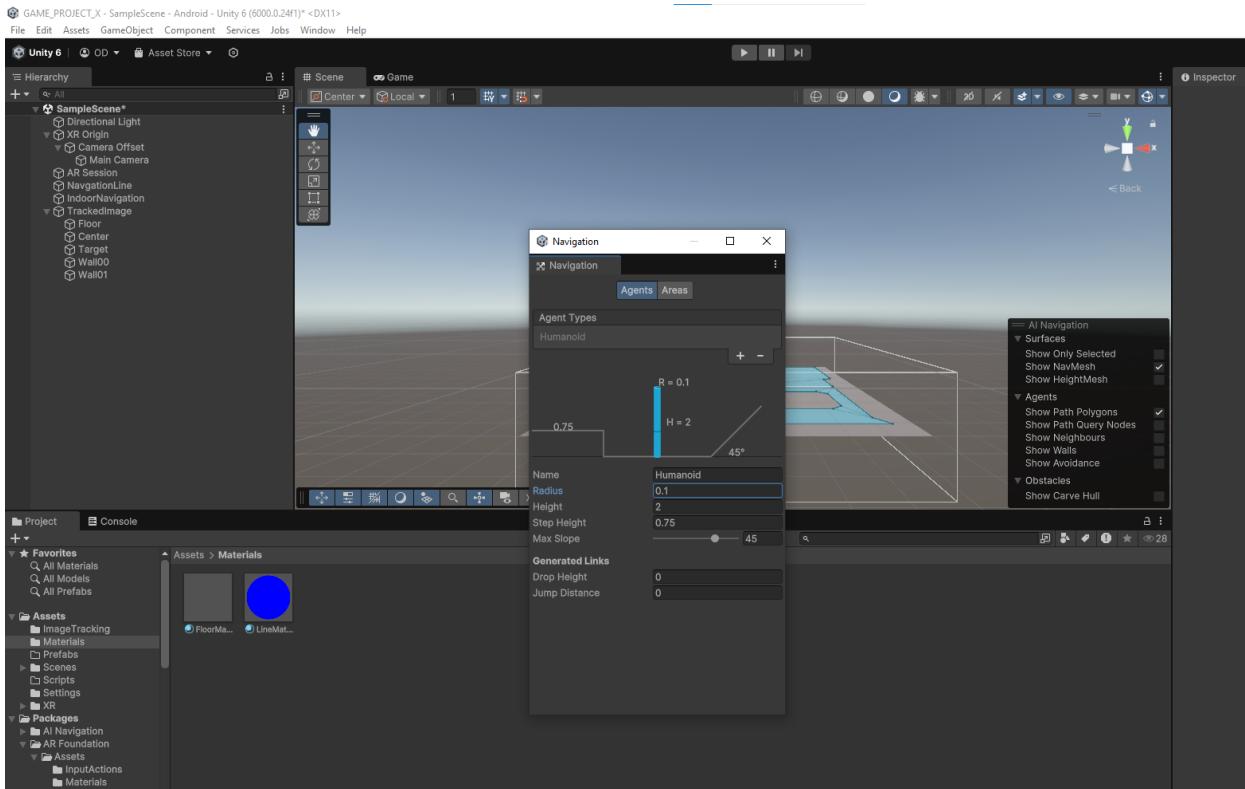
Settings

Baking light and shadow data helps in rendering static lighting more efficiently, enhancing the realism of the virtual objects. This step is crucial for AR environments as it improves visual consistency, especially when virtual objects are overlaid on real-world elements.

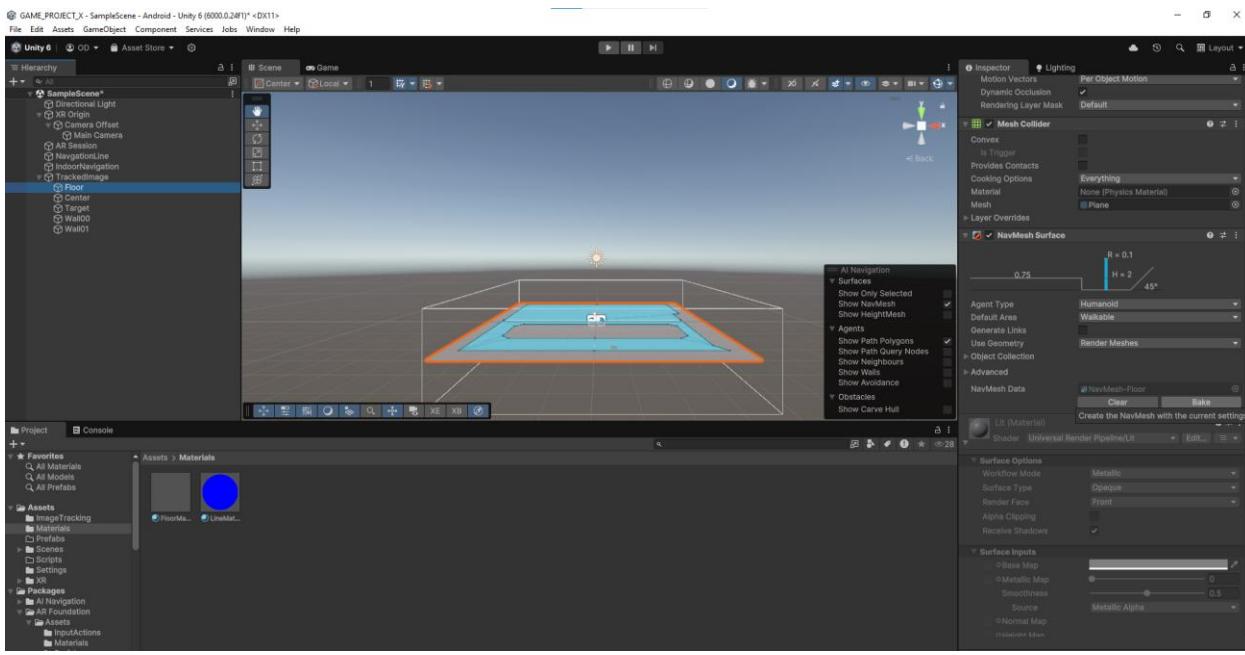




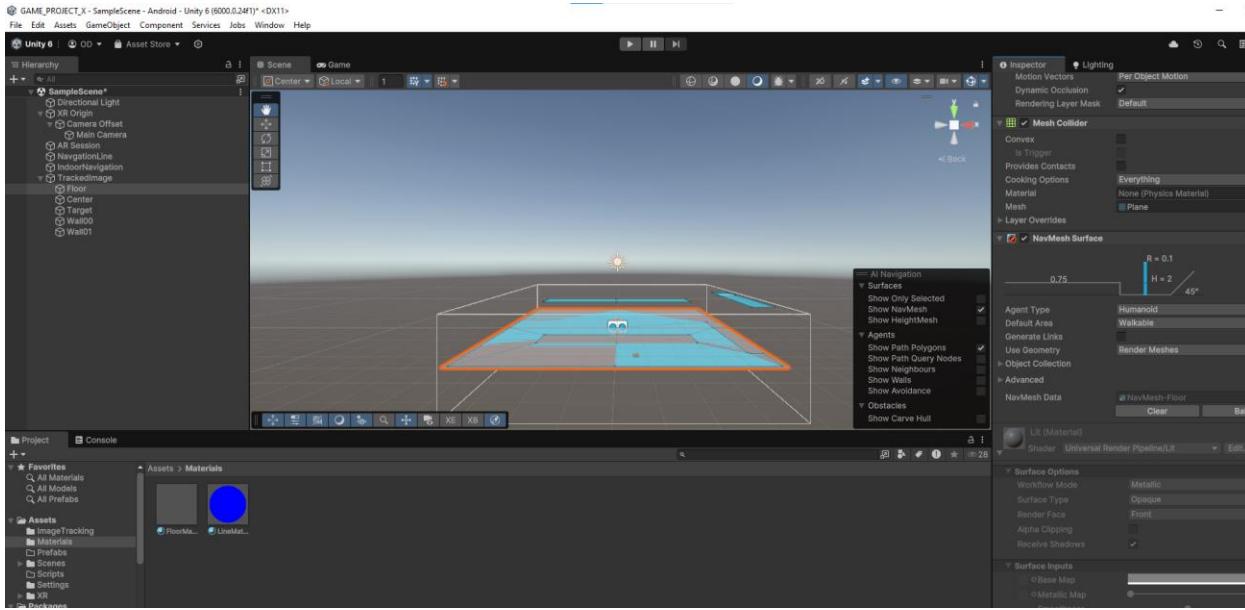
Changing radius from 0.5 to 0.1:



Before Baking 0.1 radius:

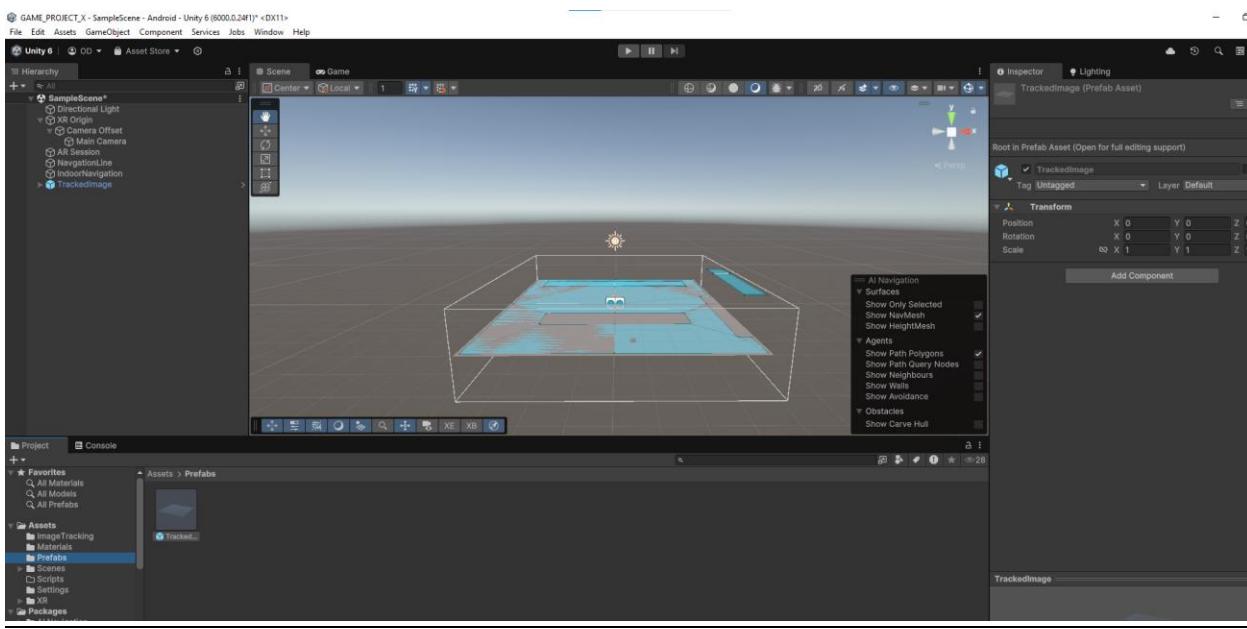


After Baking 0.1 radius:



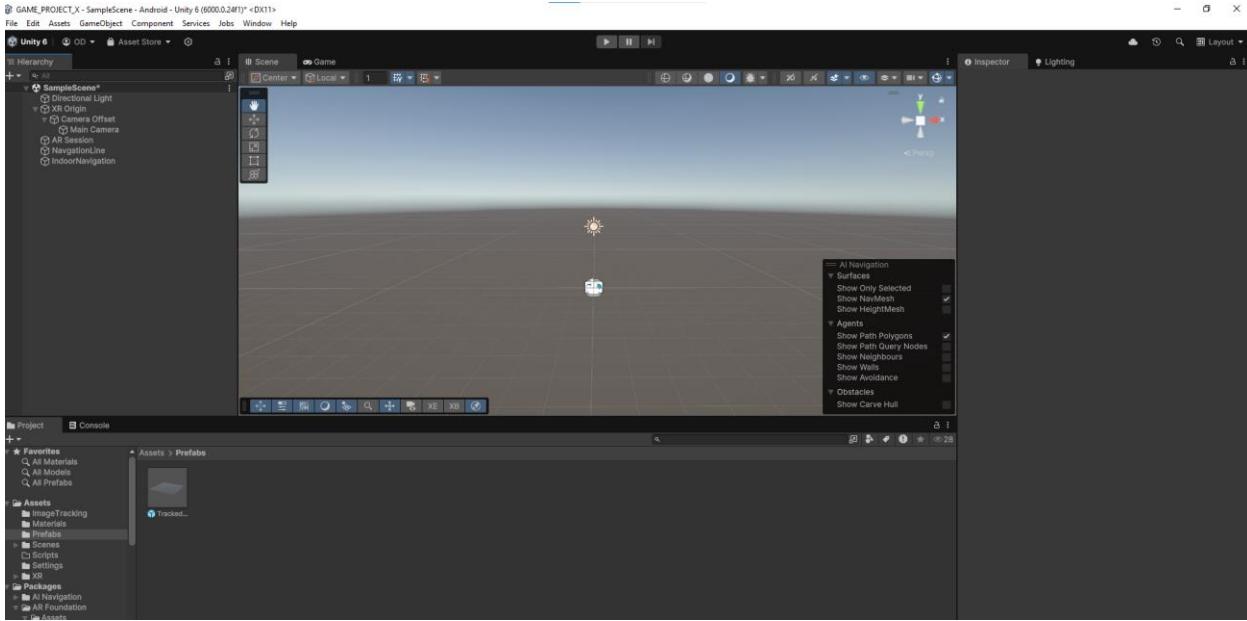
Prefab and Object Management

- Move Tracked Image Data to Prefabs
- Prefs are reusable objects in Unity, making it easier to manage multiple tracked images within the AR environment. Transferring tracked image data to Prefabs streamlines the development and makes updates more manageable.



Deleting TrackedImage from SampleScene

Once the Prefab setup is completed, the TrackedImage object in the SampleScene is removed to avoid duplication, ensuring a more organized and efficient project hierarchy.



Scripting and Code Integration

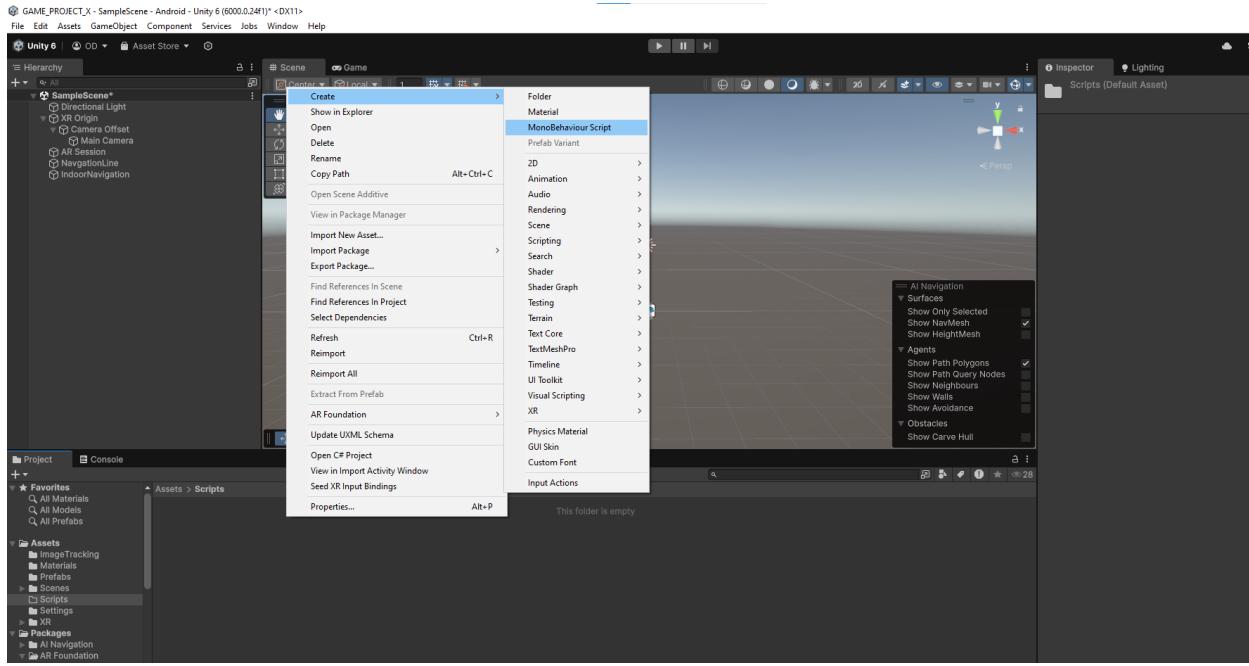
1. Open TrackedImage Prefab and Attach Navigation Script

Opening the TrackedImage Prefab and attaching the navigation script allows it to track images and initiate appropriate navigation actions. This script controls directional prompts and other interactions users experience in the app.

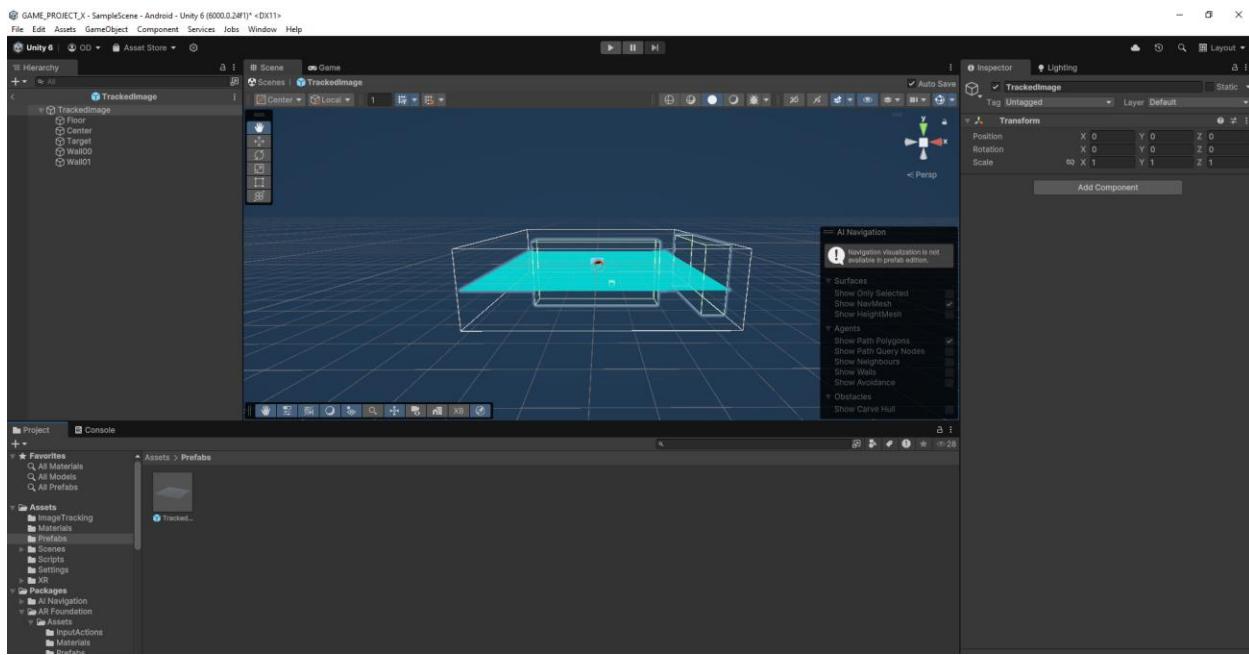
2. Update and Move NewIndoorNav.cs Script

The NewIndoorNav.cs script is customized to handle AR navigation logic, integrating pathfinding, user

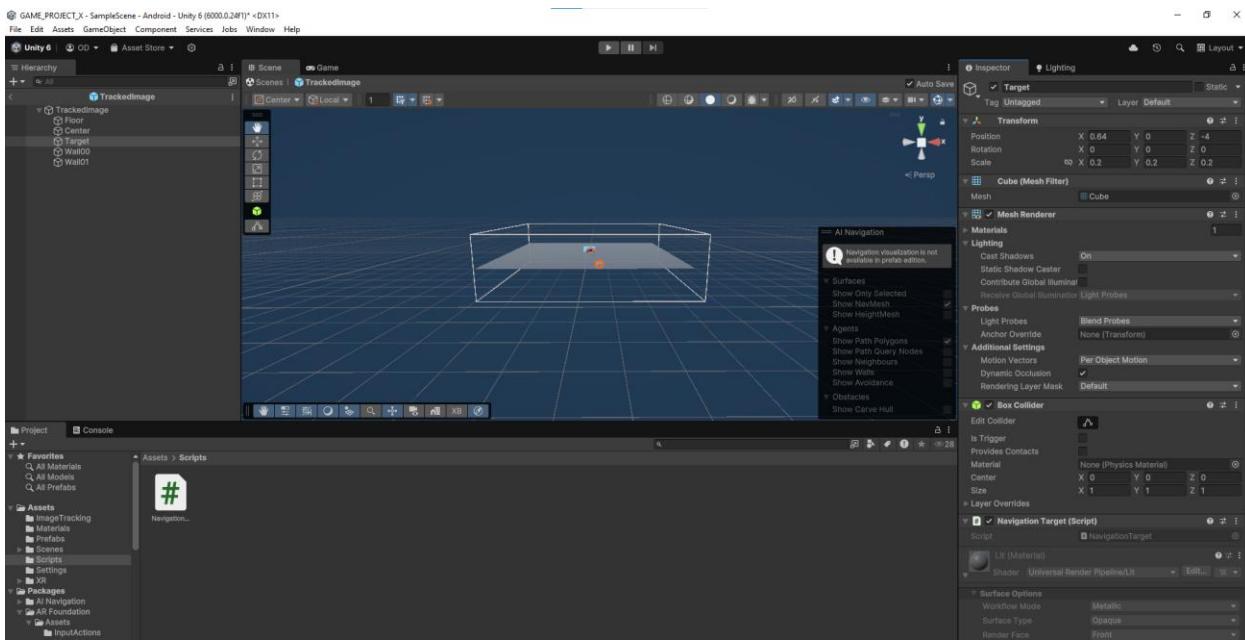
position updates, and directional cues within the AR experience. Relocating this script under IndoorNavigation helps keep the project organized.



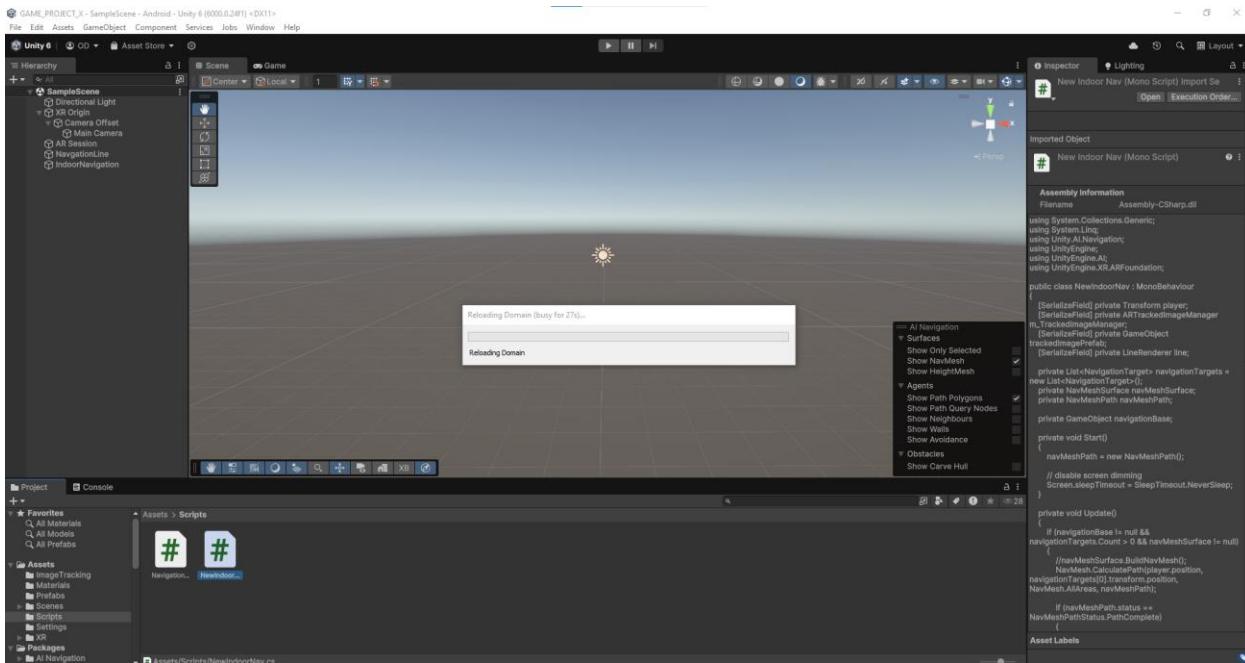
Open TrackedImage Prefab:



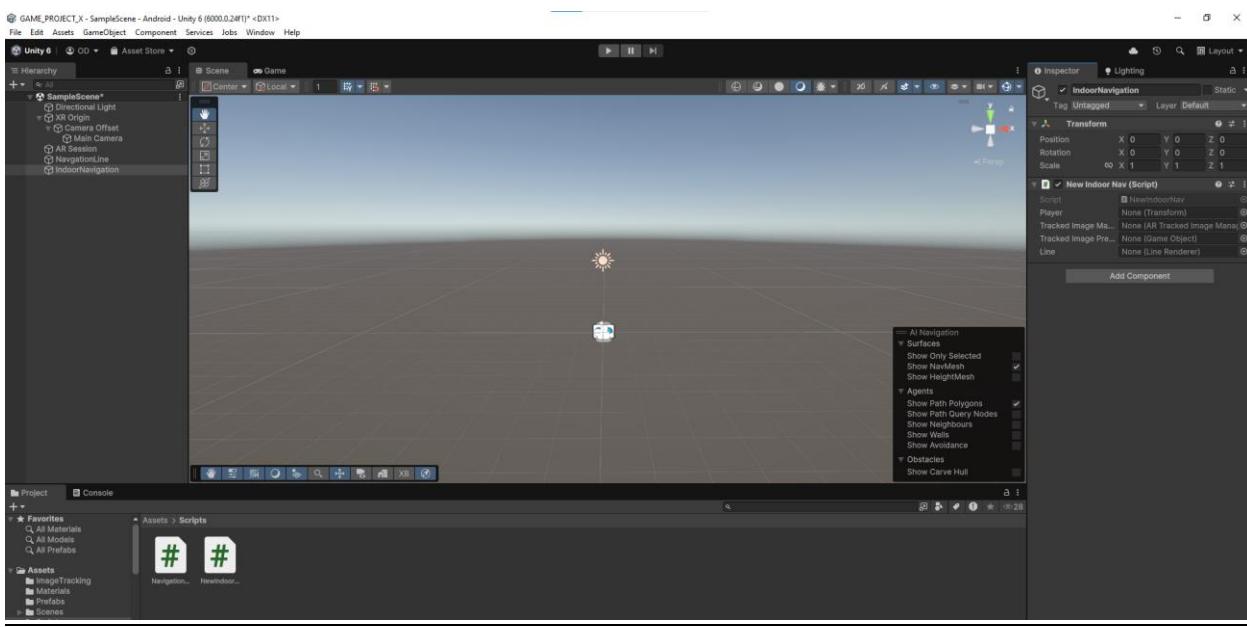
Move Navigation Script to Target:



Updating the code for NewIndoorNav.cs script:



Moving NewIndoorNav script under IndoorNavigation:

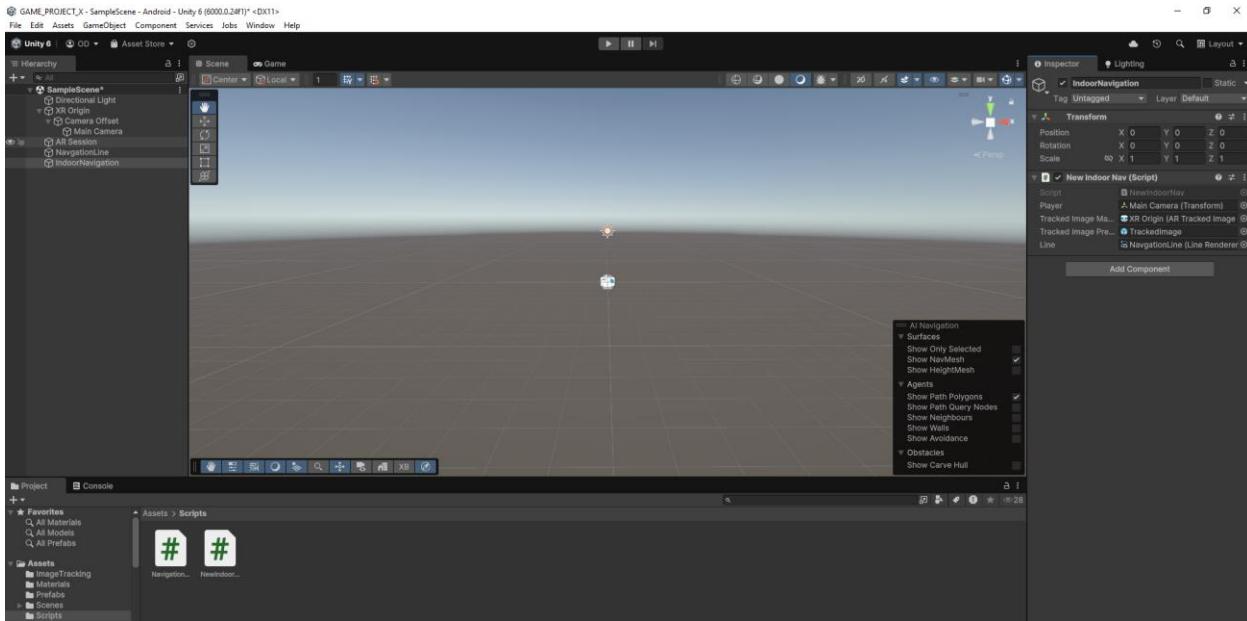


Final AR Navigation System Setup

1. Drag Components for AR Setup

For the final setup of the AR navigation system, drag the following into the New Indoor Nav script component:

- **Player:** The main camera, representing the user's perspective.
- **Tracked Image Manager:** The XR Origin, responsible for handling the entire AR setup.
- **Tracked Image Prefab:** The template used to instantiate recognized images within the AR scene.
- **Line Renderer:** Used to create a navigation line that guides users through campus.



Change the width of Line Renderer:

Modify

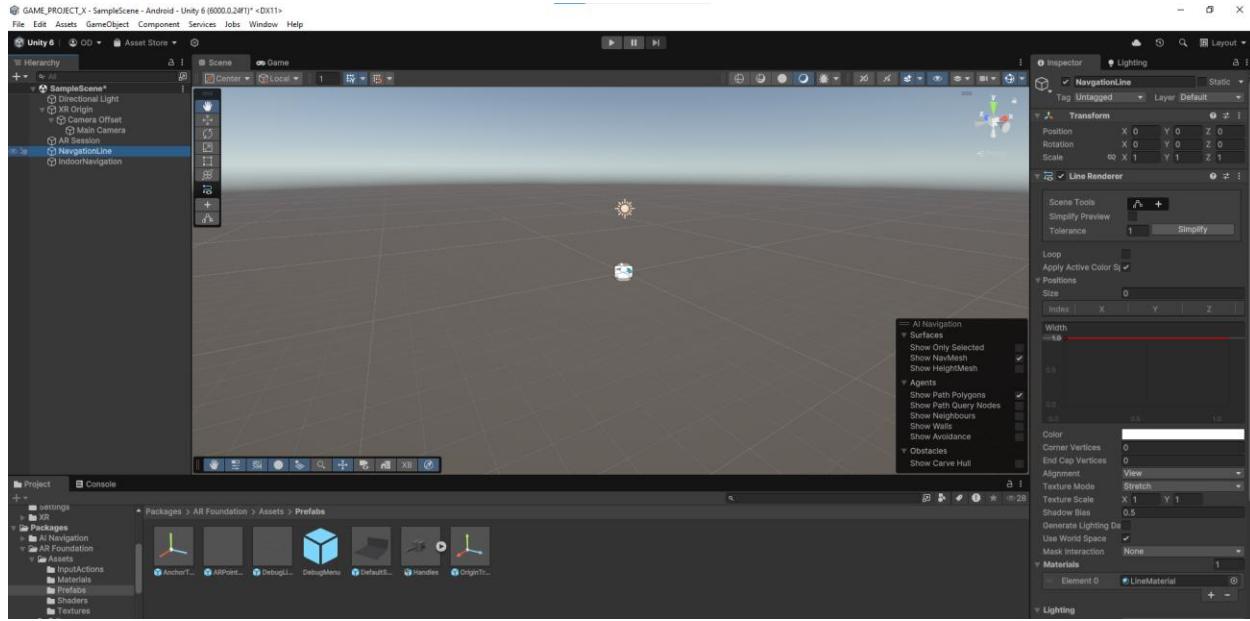
Line

Renderer

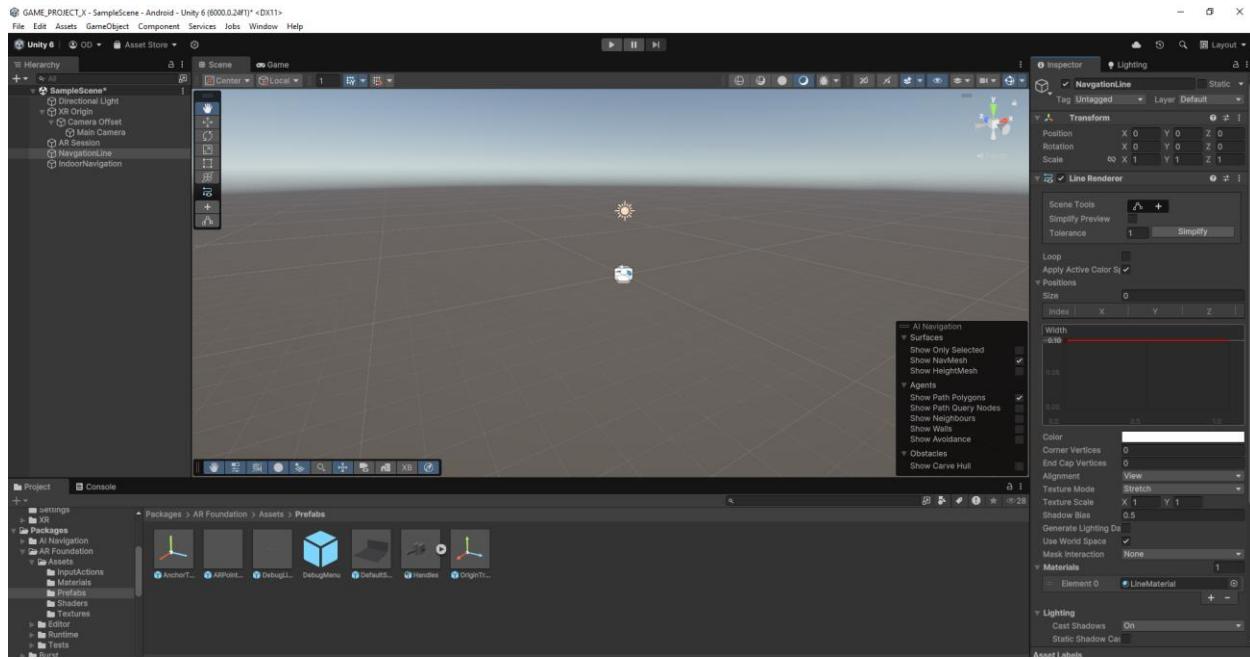
Width

Adjust the line renderer's width before and after the change, which visually impacts the clarity of navigation lines in the AR display. Fine-tuning the width ensures that the path is easily visible without obstructing the real-world view.

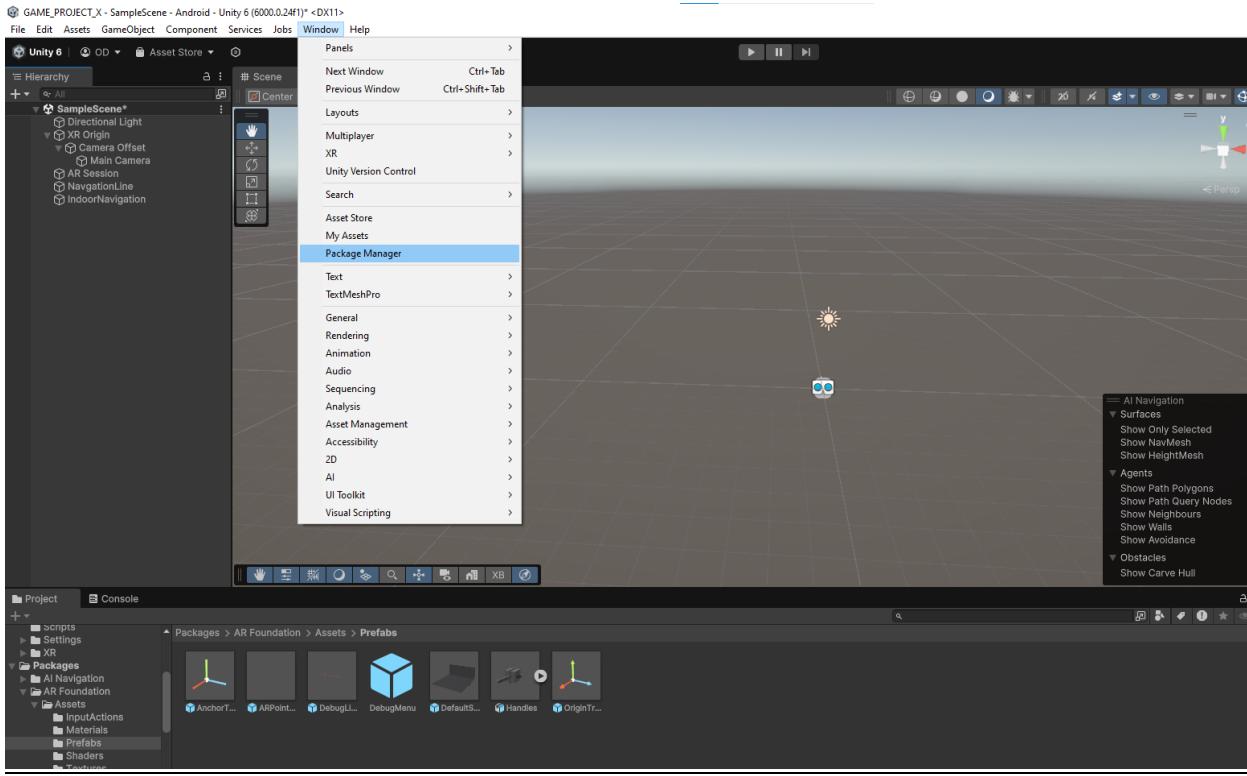
Before Change:



After Change:



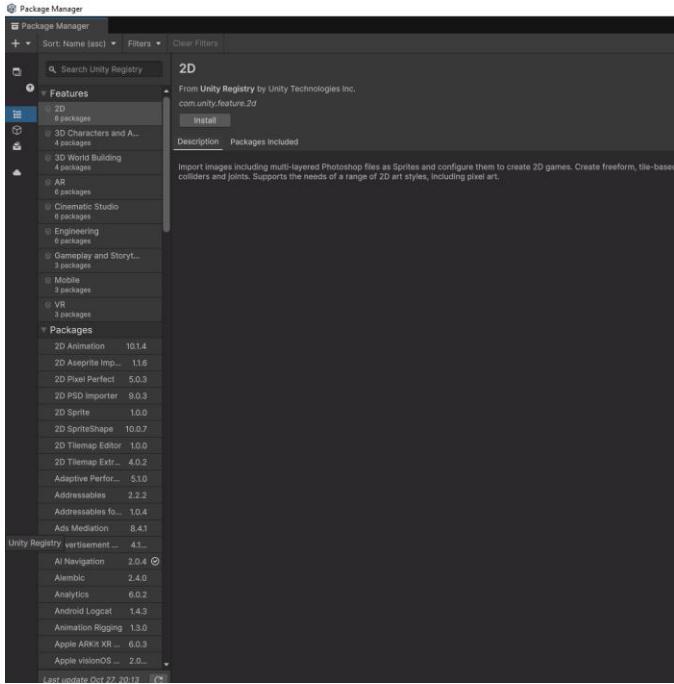
Connect device:



Building and Testing

1. Unity Registry and Build Profiles

Navigate to the Unity Registry to configure build profiles for different devices. This includes setting target devices for compatibility and adjusting performance settings specific to AR applications.



Package Manager

Package Manager

+ Sort: Name (asc) Filters Clear Filters

AR

Features

3D Characters and Animation... 4 packages

AR 6 packages

Packages

Apple ARKit XR Pl... 6.0.3

AR Foundation 6.0.3 ⓘ

Character Controller 1.2.4

Core RP Library 17.0.3 ⓘ

Google ARCore XR... 6.0.3 ⓘ

Leaderboards 2.1.0

SharpZipLib 1.3.8

Toolchain MacOS ... 2.0...

Toolchain WinArm... 1.0.4

Tutorial Authoring ... 1.2.2

Tutorial Framework 4.0.2

WebGL Publisher 4.2.3

ZivaRT Player 2.1.1

3D Characters and Animation

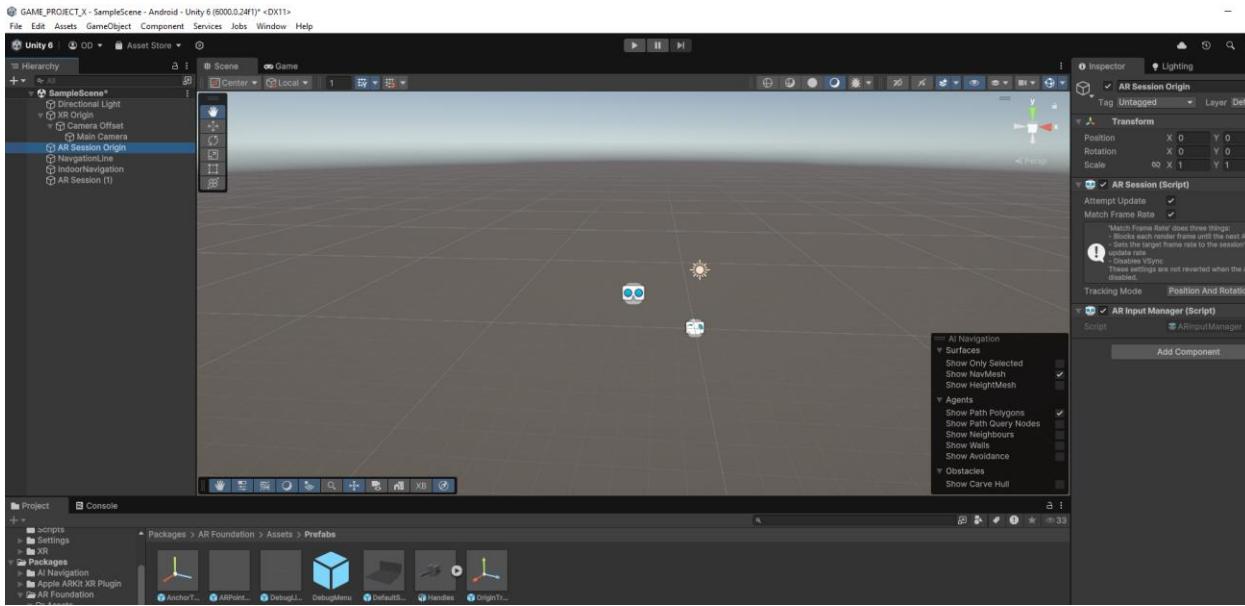
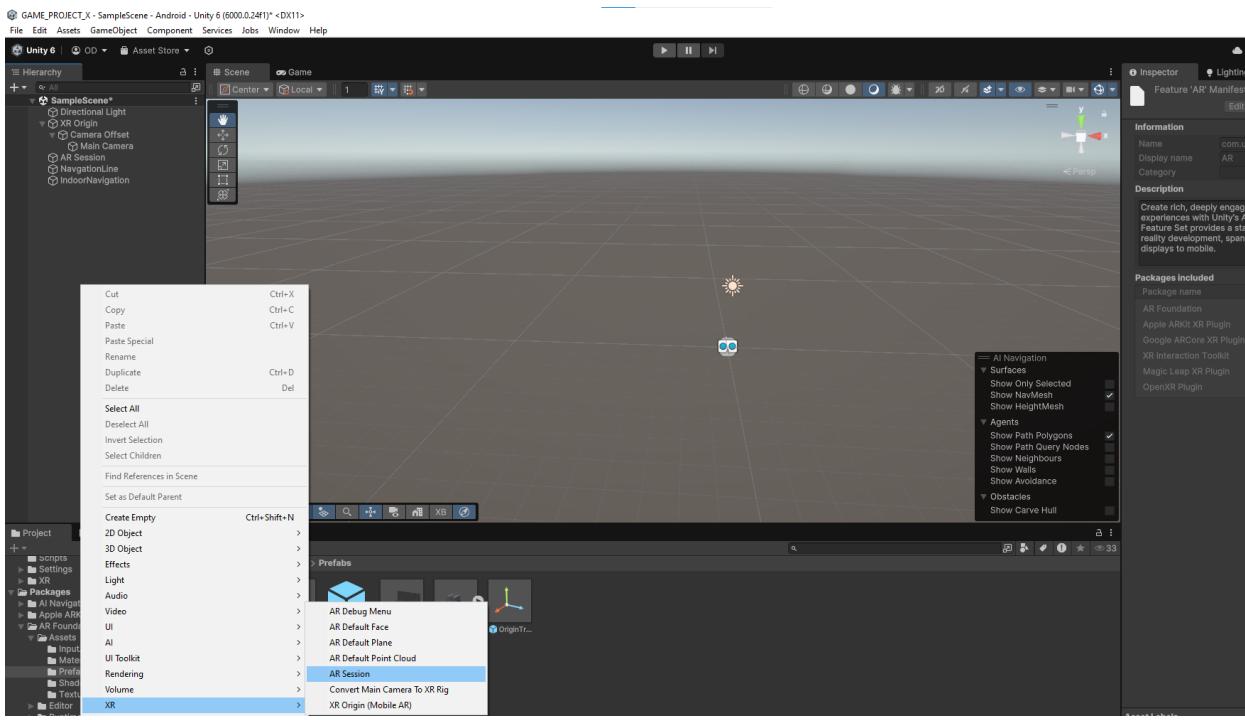
From Unity Registry by Unity Technologies Inc.
`com.unity.feature.characters-animation`

Install

Description Packages Included

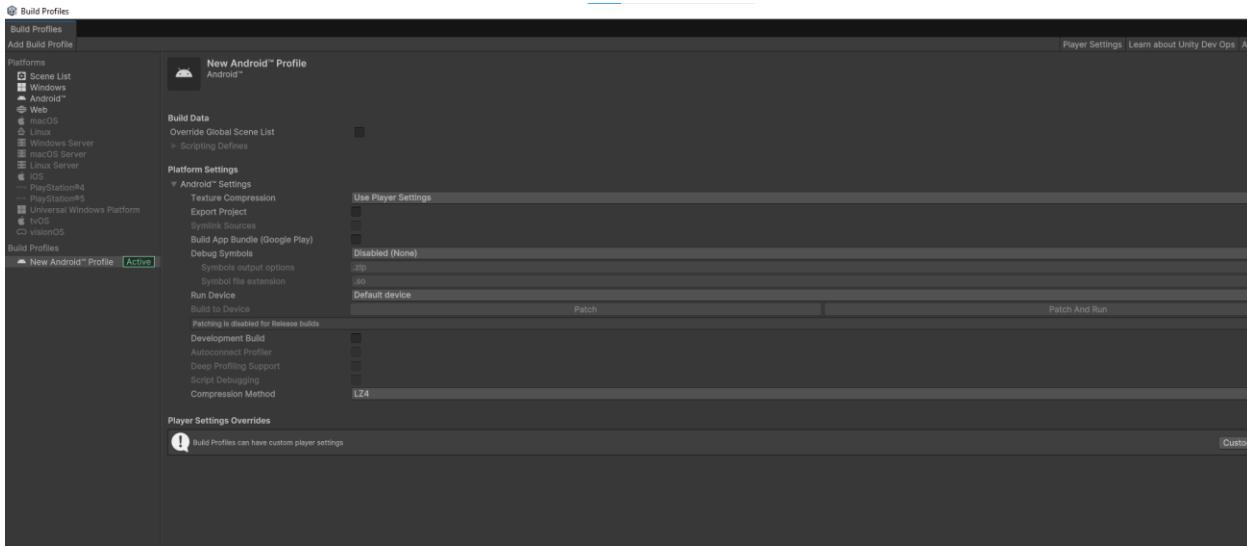
Animate characters and create cutscenes with the Characters and Animation feature set.

Build interactive character rigs and constraints to animate characters and GameObjects. Arrange tracks and clips to create gameplay sequences, cutscenes, or movies. Orchestrate, shoot, and record beautiful narratives for cutscenes or movies. Easily round-trip to your 3D modeling software

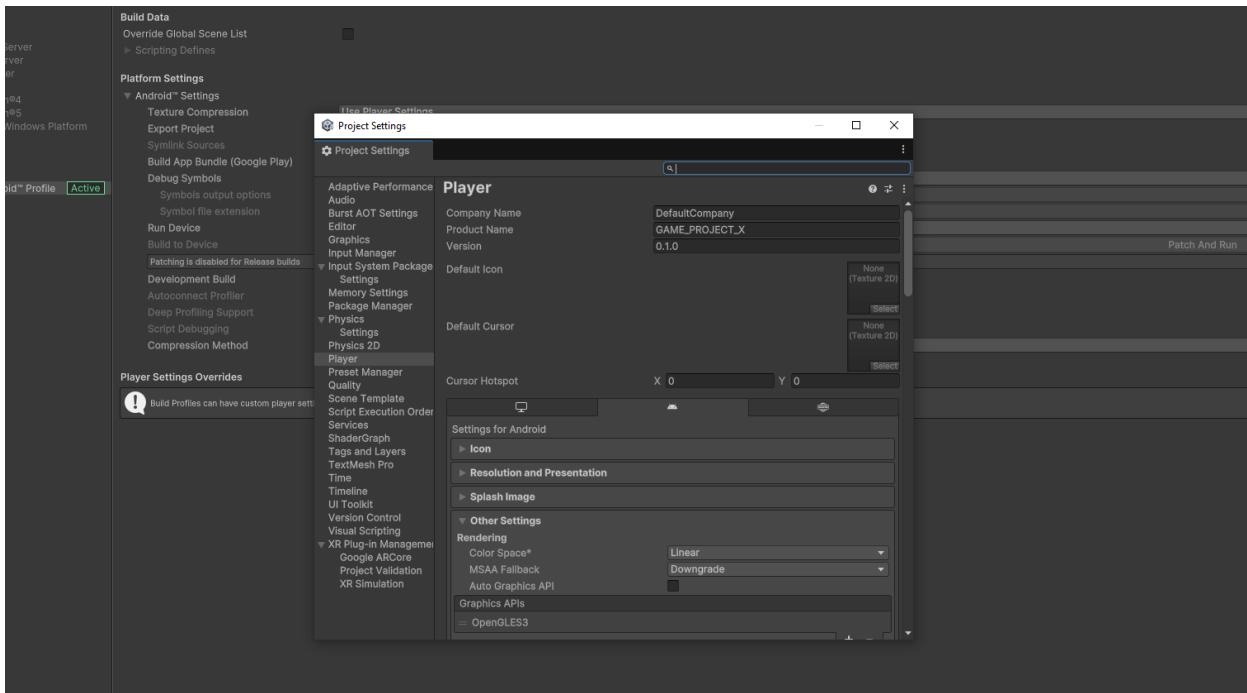


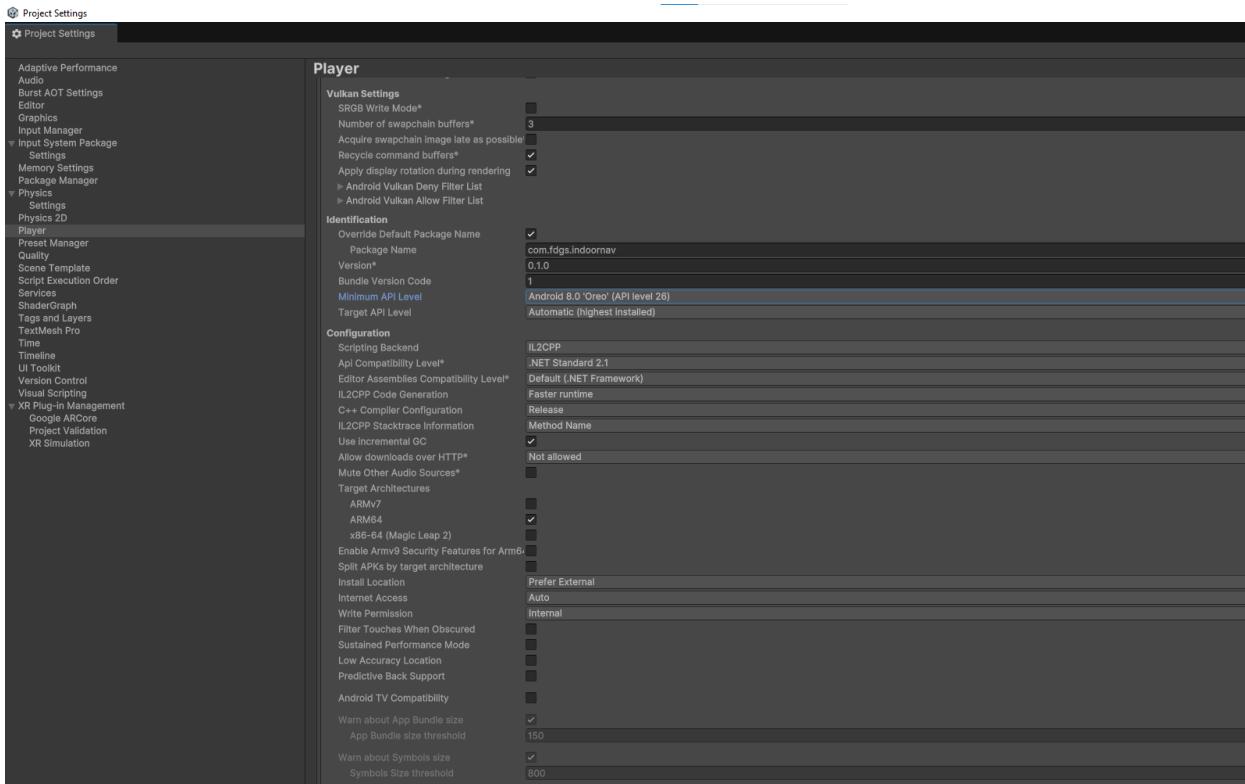
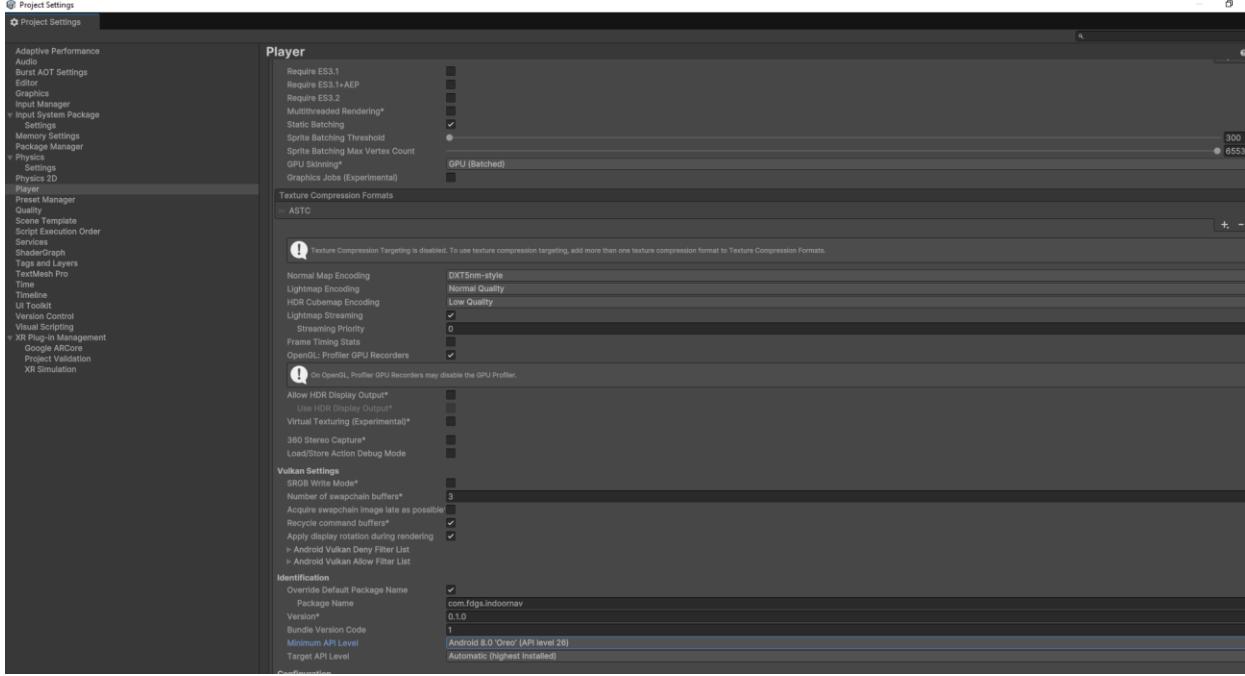
Configure Player Settings for XR Plugin Management

In Player Settings, enabling XR Plugin Management allows the app to connect to compatible AR devices. Select the appropriate plugins for ARCore or ARKit based on the target platform (Android or iOS).

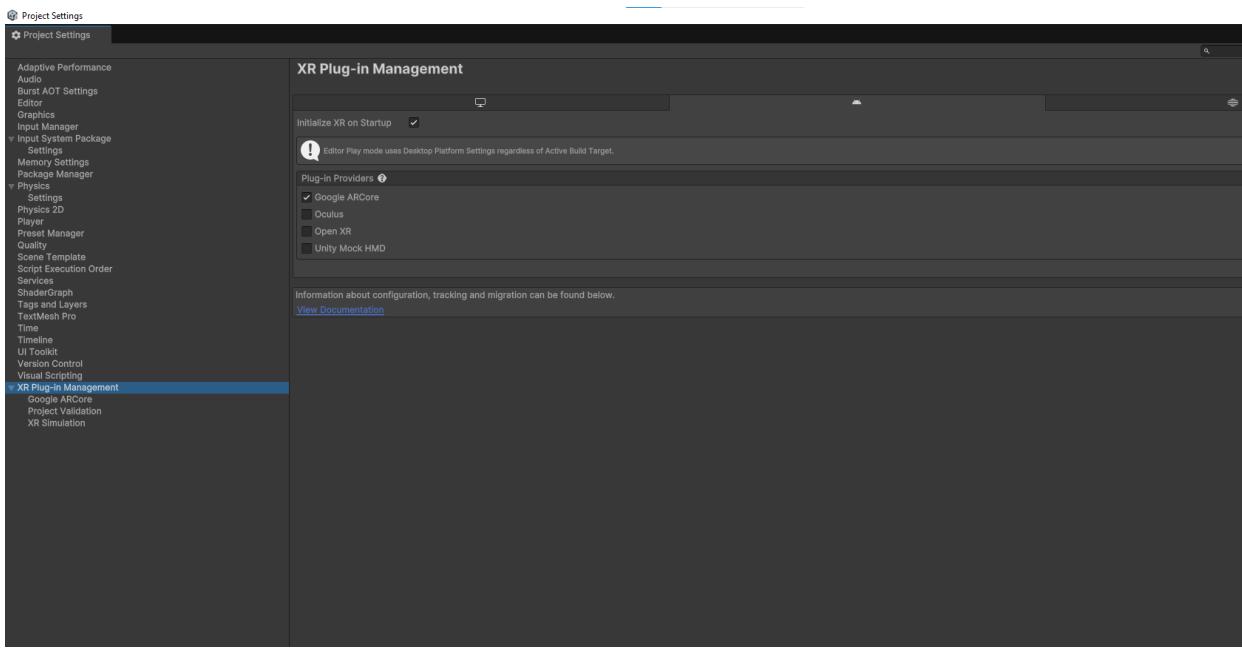


Go to Player Settings:



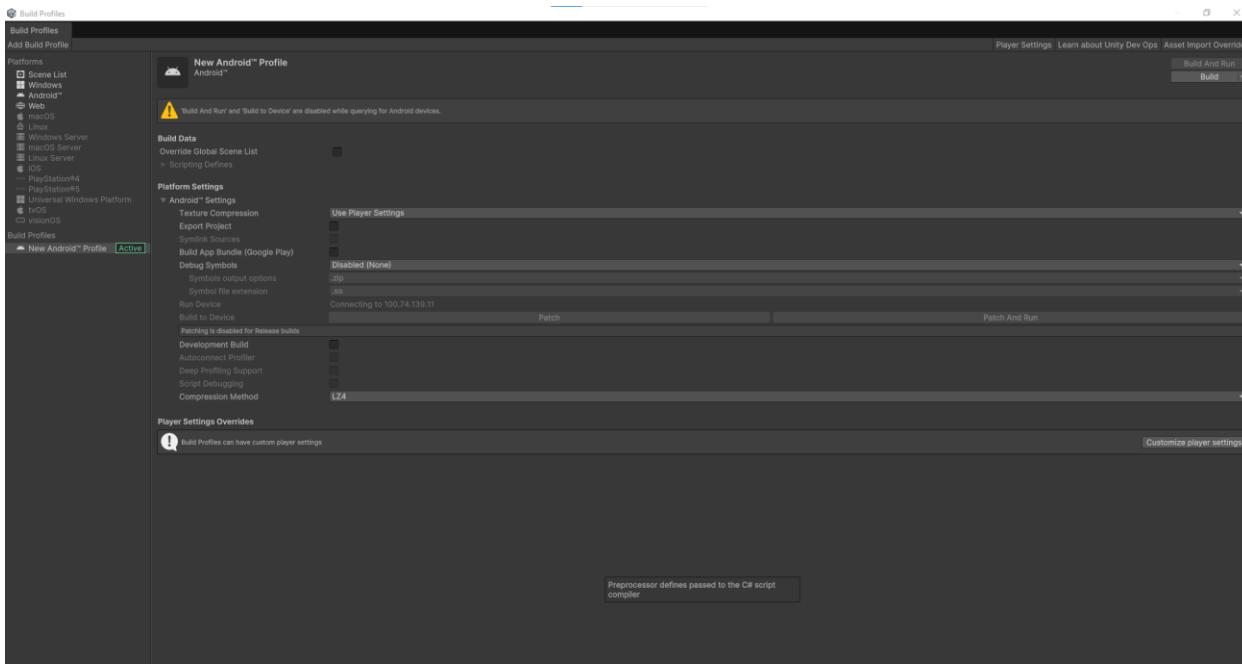


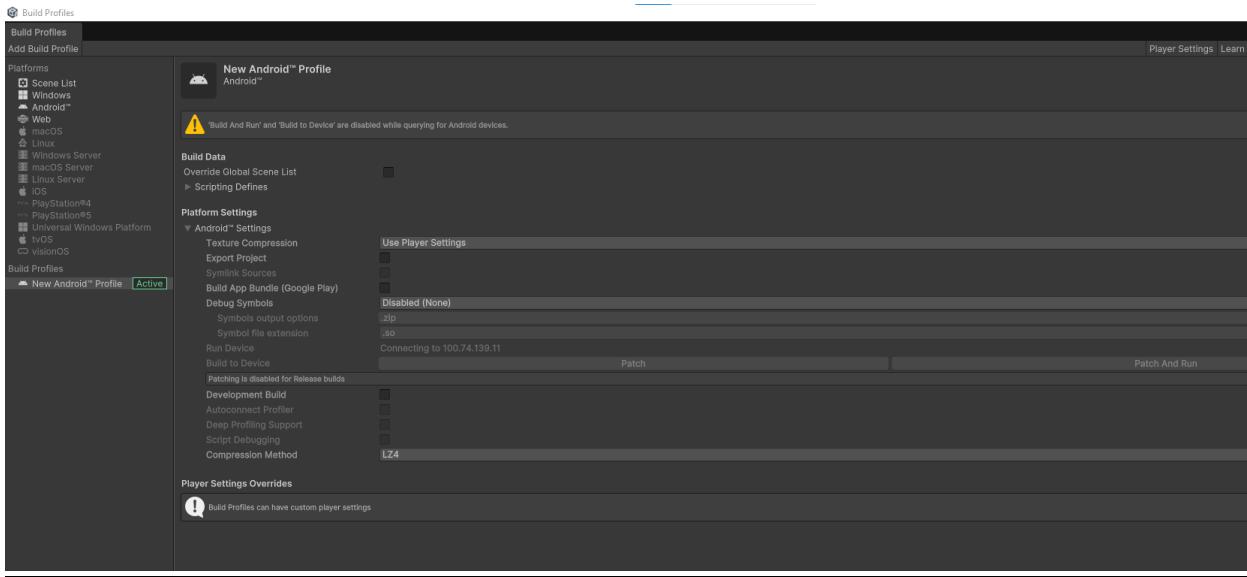
In player settings-> XR Plugin Management



Establish Connection with Secondary Device via IP Address

For real-time testing and debugging, connect to a secondary device using its IP address. This allows you to simulate the campus navigation experience and validate the AR interactions before final deployment.





Conclusion

The AR Campus Navigation project showcases how augmented reality can transform the campus navigation experience, offering users an intuitive way to find locations using their mobile devices. The careful setup of AR components, efficient scene management, and thoughtful UI considerations ensure a seamless and engaging user experience. This project is not only a technical achievement but also a significant step towards modernizing campus accessibility through interactive technology.