

Designing Educational VR Experiences in Unity

A Beginner's Guide to Building Interactive VR Learning Environments

Fatemeh Ashrafabadi

University of Nebraska–Lincoln

2026



Table of Contents

Introduction

- Introduction
- Who Is This Guide For?
- Learning Outcomes
- How This Guide Is Organized

Chapter 1 – Foundations of Educational VR Design

1.1 Research Foundation

1.2 VR Activity Planning

1.3 Example Scene Design

Chapter 2 – Building the Experience in Unity

2.1 Unity Overview

- What We Are Building
- Tools Checklist
- Unity Navigation Shortcuts

2.2 Setting Up the VR Project

- Install Unity Hub
- Install a Unity Editor Version
- Create a New Project
- Install VR Packages

2.3 Player Movement

- Teleportation Provider
- Teleportation Area
- Configure the Controller
- Configure Input Actions
- Add the Teleport Ray
- Configure Locomotion
- Test Teleportation

2.4 Importing 3D Assets

- Supported File Formats

- Import Assets
- Place Assets in the Scene

2.5 Creating Interactive Objects

- Prepare the Object
 - Configure the Collider
 - Configure the Rigidbody
 - Add an XR Grab Interactable
 - Make the Grab Feel Natural (Attach Transform)
 - Test the Interaction
-

How This Guide Is Organized

This guide is divided into two parts:

Part I introduces the educational principles that guide effective VR design.

Part II demonstrates how to implement these principles using Unity and the XR Interaction Toolkit through step-by-step tutorials.

Introduction

Virtual reality has become an increasingly powerful tool for education, allowing learners to interact with digital environments in ways that traditional learning materials cannot provide. Creating educational VR experiences, however, requires both instructional design knowledge and technical development skills.

This guide introduces the fundamental concepts needed to design and build educational VR applications using Unity and the XR Interaction Toolkit. It is written for beginners with little or no previous experience in VR development.

Throughout this guide, you will learn how to:

- Design educational VR activities
- Set up a Unity VR project
- Import 3D models
- Configure player movement
- Create interactive objects
- Build complete educational VR scenes

Who is this guide for?

This guide is intended for:

- Teachers
- Educational researchers
- Graduate students
- Instructional designers
- Beginning Unity developers

- Anyone interested in educational VR

Learning Outcomes

By the end of this guide, you will be able to:

- ✓ Plan an educational VR activity
- ✓ Configure Unity for VR
- ✓ Create a VR environment
- ✓ Add teleportation
- ✓ Import 3D assets
- ✓ Make objects interactive
- ✓ Test the experience on Meta Quest

Chapter 1 –Foundations of Educational VR Design

1.1 Research Foundation

Before we build anything in Unity, we need to ground our VR design in research so the experience is purposeful, usable, and effective—not just visually impressive.

1.1.1 Use research to guide design choices

We need to pay attention to what studies show about engagement, cognitive load (not overwhelming the user), feedback, transfer to real tasks, and comfort in headsets.

1.1.2 Apply core principles that improve VR experiences

- Users should always know what they are trying to do and what “done” looks like.
- VR works best when users make choices and perform actions, not just watch.
- Actions should happen in a scenario that makes sense and explains “why this matters.”
- The system should show results right away and allow safe retries.
- Include close options so users must think (not guess randomly).
- Start simple, then increase challenge once users understand the basics.
- Revisit key actions/ideas later in a slightly new way to strengthen learning.
- Reduce motion sickness triggers and include options (teleport vs smooth move, readable text, audio support).

1.1.3 Turn principles into a repeatable activity structure

Every scene should follow a simple loop:

Introduce the goal → let the user act → give feedback → check understanding → move forward

1.2 VR Activity Planning

Use this template to plan any VR activity before you build it:

- What does the user accomplish?
- Where are they and why does it matter?
- What do they physically do in VR?
- What options do they compare (include close alternatives)?
- What happens for correct/incorrect actions?
- How does the idea/action reappear later in a new form?
- How do we reduce motion sickness and overload?

1.3 Example Scene Design

This is an example of applying the design principles and vocabulary plan to one scene. The scene is a bright croquet court with hedge walls and red roses, card-soldier guards, and the Queen seated on a pavilion throne.

The activity targets Tier 2 vocabulary: command, protect, prevent, and uses Task-Induced Involvement (Need → Search → Evaluation) + active engagement, based on Laufer & Hulstijn's Involvement Load Hypothesis.

1.3.1 Student experience (steps)

1. Need (goal/mission): The Queen gives an audio command (“Make the roses safe. Protect them before the game!”) with kid-friendly subtitles (command = tell with power; protect = keep safe).



2. Search + Evaluation (meaning-based choice): Near the roses, students choose which action really means protect: place shields (correct), water the roses (decoy: care $\neq$ protect), or put up a “KEEP AWAY” sign (decoy: command $\neq$ protect). Each option includes a short hint bubble and immediate feedback (success locks shields; incorrect prompts a retry).

3. Reinforcement (extend to prevent): A short follow-up step highlights prevent by emphasizing stopping harm before it happens (vs. fixing after), then students finish with a brief sentence choice/say-back using the target words.

Chapter 2 – Building the Experience in Unity

In this chapter, you will configure a Unity project for virtual reality and learn the core development skills needed to build an educational VR experience, from setting up the project to creating interactive objects.

2.1 Unity Overview

2.1.1 What we are building

We are creating an interactive virtual reality experience where users explore a 3D environment and complete simple tasks by moving around and interacting with objects.

- Open Unity and an existing VR project correctly
- Enter VR (with a headset) and move/teleport around the environment
- Build a simple environment (terrain/ground + basic textures)
- Import 3D models and organize them in the scene
- Make objects interactive (grab, place, click)
- Understand why prefabs/instances and scene organization matter for performance and smooth VR gameplay

2.1.2 Tools Checklist

Make sure your computer has everything required to open and run a Unity VR project.

What you need (hardware):

- A computer that can run Unity (recommended: at least 16GB RAM if possible)
- A VR headset (Meta Quest)
- A USB-C cable for Link Air Link (Wi-Fi)
- Enough storage space (Unity projects can be large)

2.1.3 Unity Navigation Shortcuts

These are the most important controls:

- Right click + WASD: fly around the Scene view
- Mouse wheel: zoom
- Alt + Left mouse: orbit around a point
- Alt + Middle mouse: pan
- F: frame selected object (jump to it)
- Ctrl/Cmd + Z: Undo
- Ctrl + Y / Cmd + Shift + Z: Redo

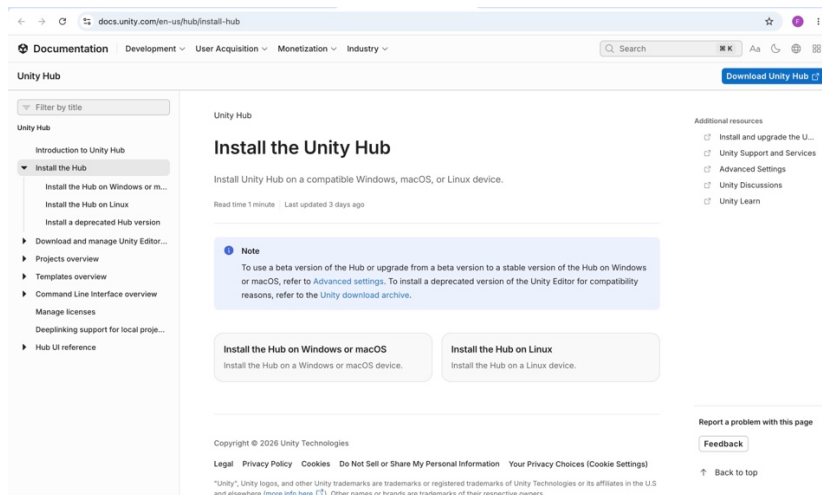
If you “lose” everything:

Select a known object like Terrain → press F.

2.2 Setting Up the VR Project

2.2.1 Install Unity Hub

- Download and install Unity Hub from the Unity website: <https://docs.unity.com/en-us/hub/install-hub>

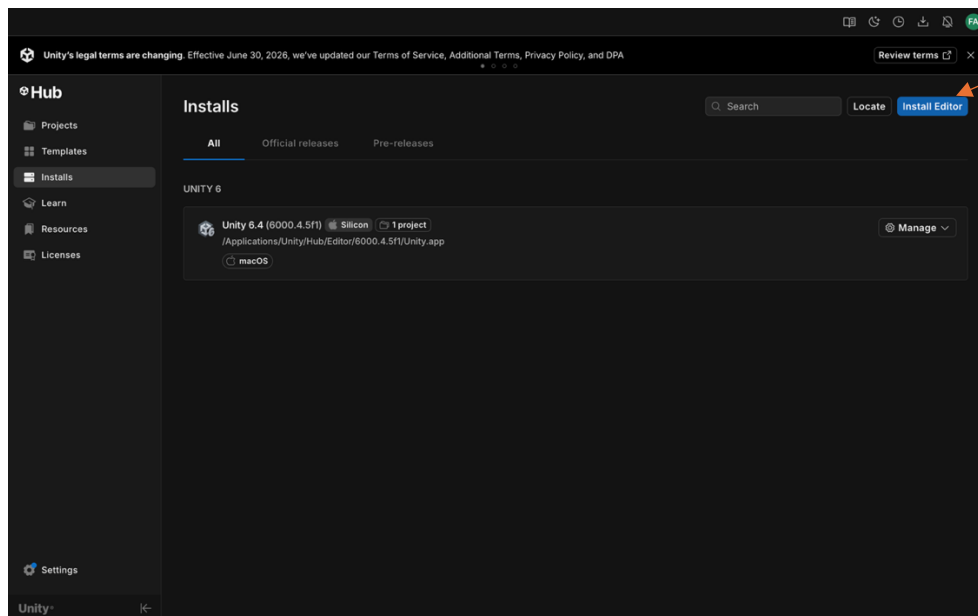


- Sign in or create a Unity account (free is fine): <https://login.unity.com/en/sign-up>

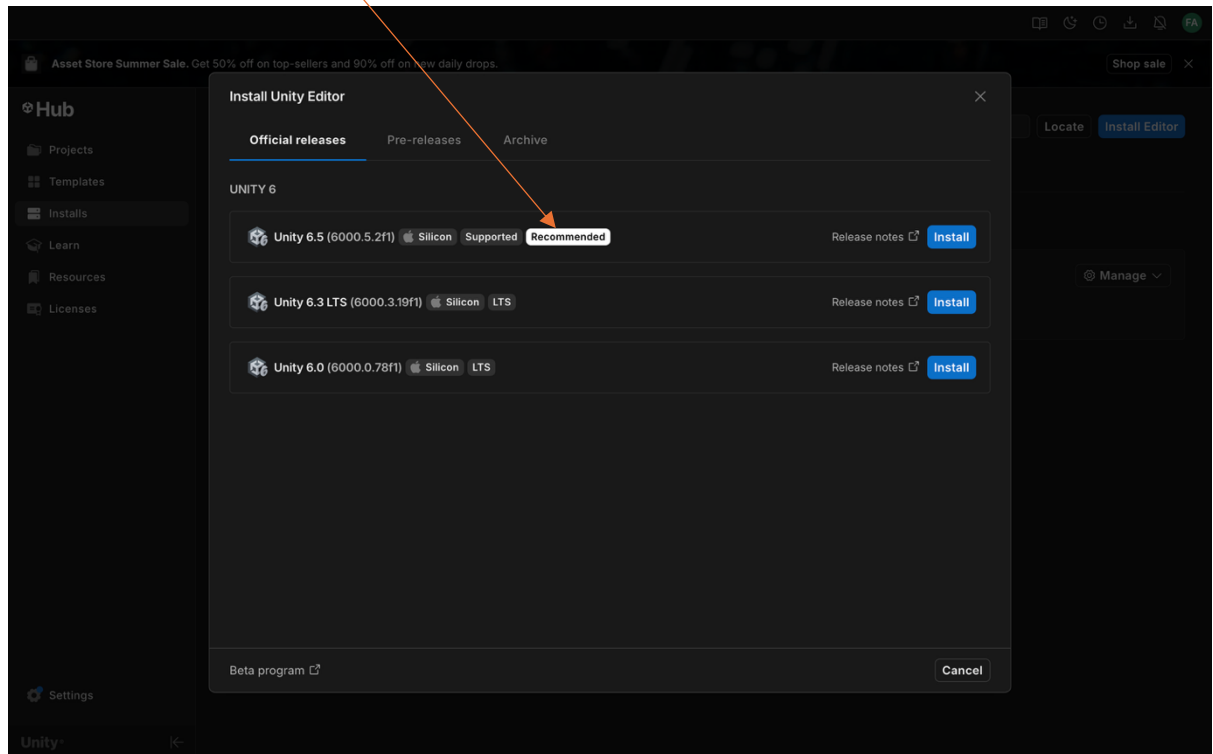
2.2.2 Install a Unity Editor version

In Unity Hub:

- Go to Installs.
- Click Install Editor.



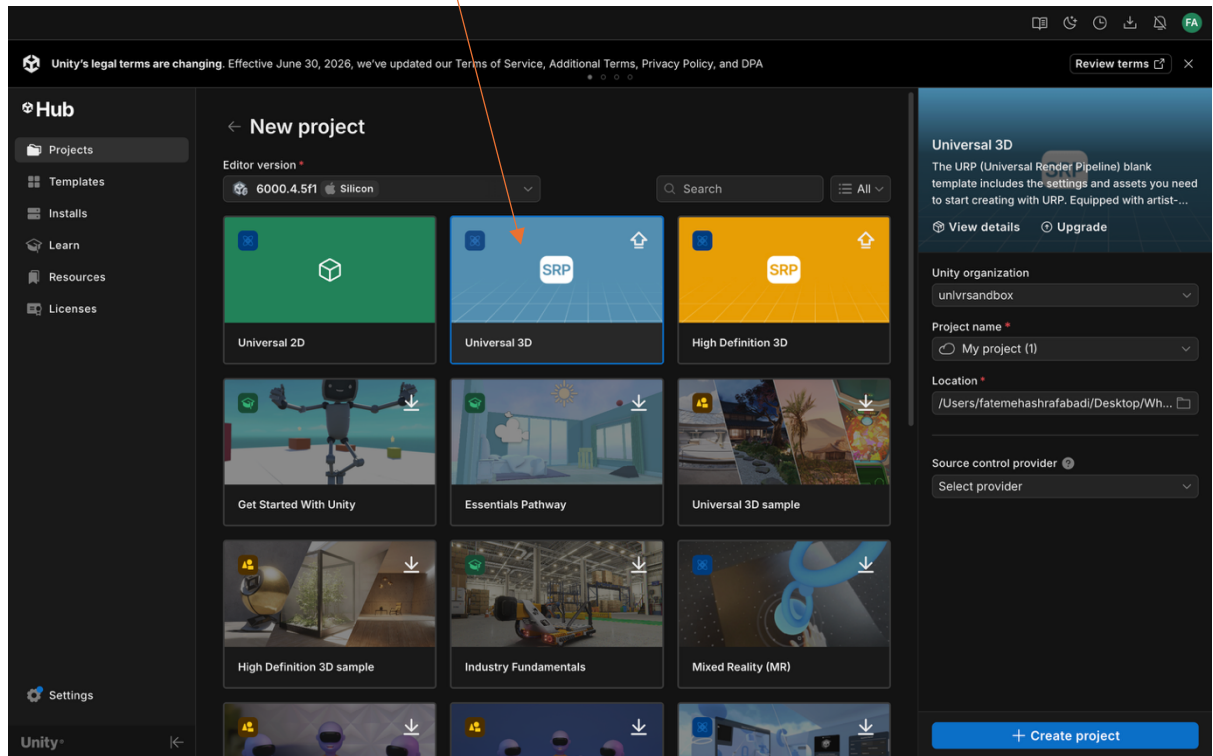
- Choose the recommended **Long-Term Support (LTS)** version).



2.2.3 Create a new project

- Go to projects
- Click **New Project**.
- Choose **3D (Core)** or **Universal 3D** (whichever your tutorial uses).
- Name the project.
- Click **Create Project**.

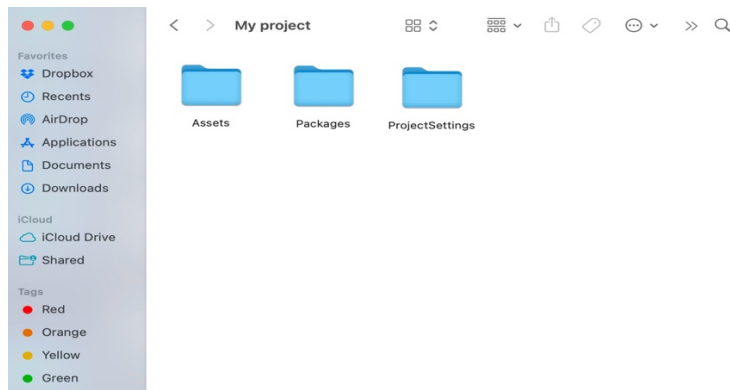
Universal 3D



After creating the project a project file will be created automatically on your computer

it must contain:

- Assets
- Packages
- ProjectSettings



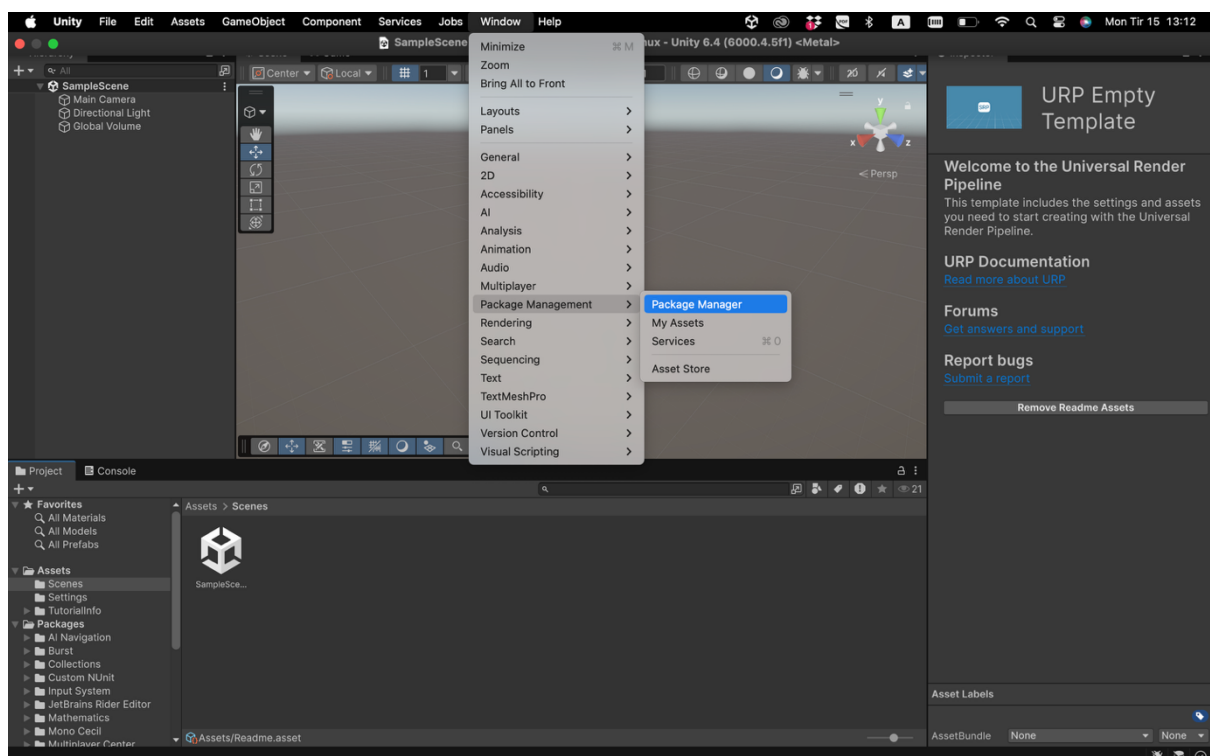
Important:

Signing into Unity with the same account does not automatically bring your project to a new computer. Unity projects are stored as folders on the computer, so you must copy or sync the project folder (USB/OneDrive/Drive/Git) and then open it through Unity Hub.

2.2.4 Install VR packages

Once the project opens:

Window → Package Manager



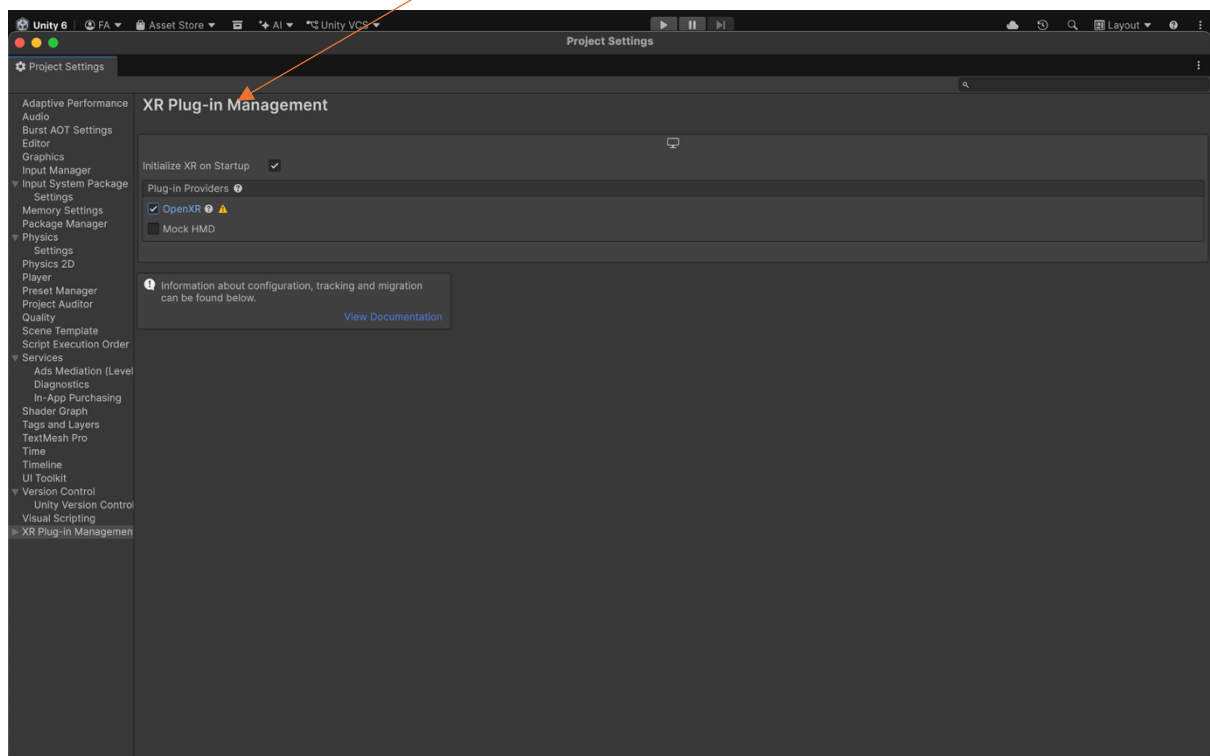
And install:

1) XR Plug-in (the “VR connection layer”)

Click on Unity Registry

Search for **XR Plug-in Management** and install it.

Go to Edit → Project Settings → **XR Plug-in Management** → Under plugin providers enable Open_XR.



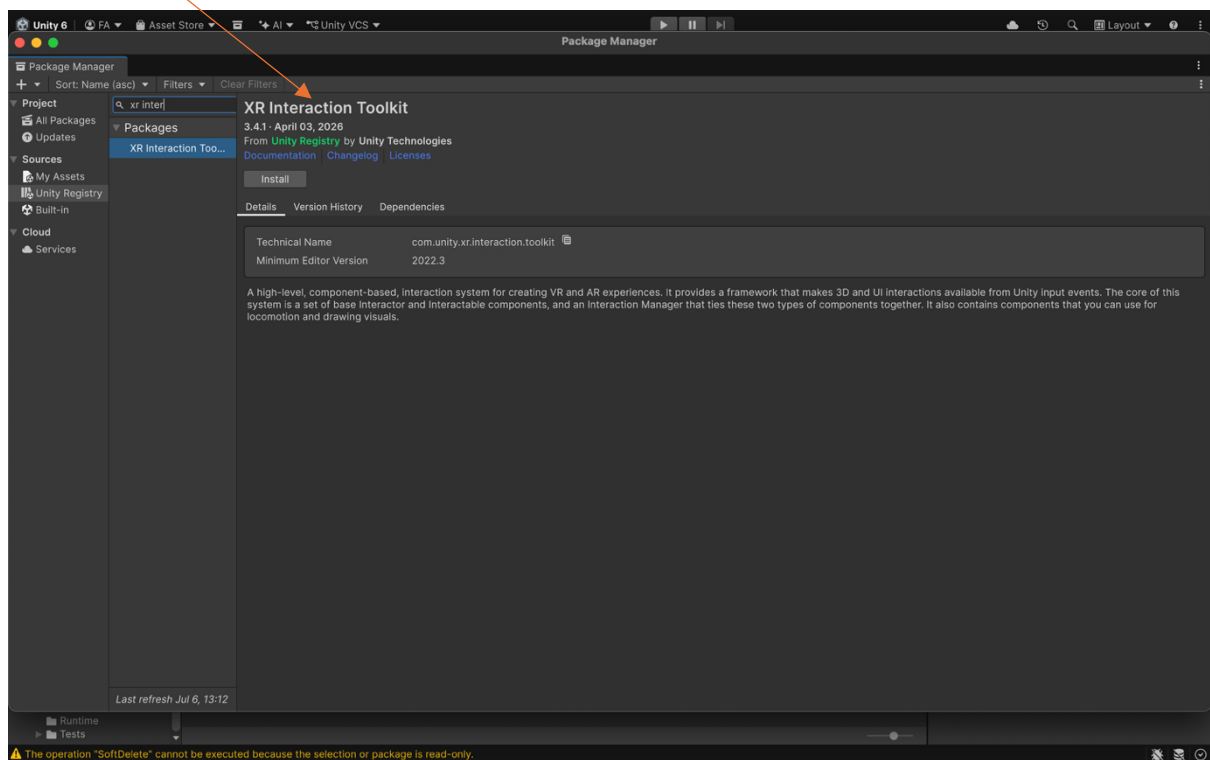
Unity needs an XR plug-in so it can “talk” to the headset. This is what lets Unity send the VR image to the headset and receive head/controller tracking data.

What it enables:

- Head tracking (look around naturally)
- Controller tracking (hands move in VR)
- VR rendering (stereo view in the headset)

2) XR Interaction Toolkit (the “VR interactions layer”)

go to Window → Package Management → Package Manager → Search for **XR Interaction Toolkit** and install it.



This is Unity’s toolkit for common VR interactions. It

provides ready-made systems for:

- Teleportation (comfort-friendly movement)
- Ray pointing (laser pointer for selecting objects or UI)
- Grabbing objects (pick up, drop, throw)
- Interactable objects (press, select, activate)

Why we use it:

It saves time and ensures interactions work consistently across devices.

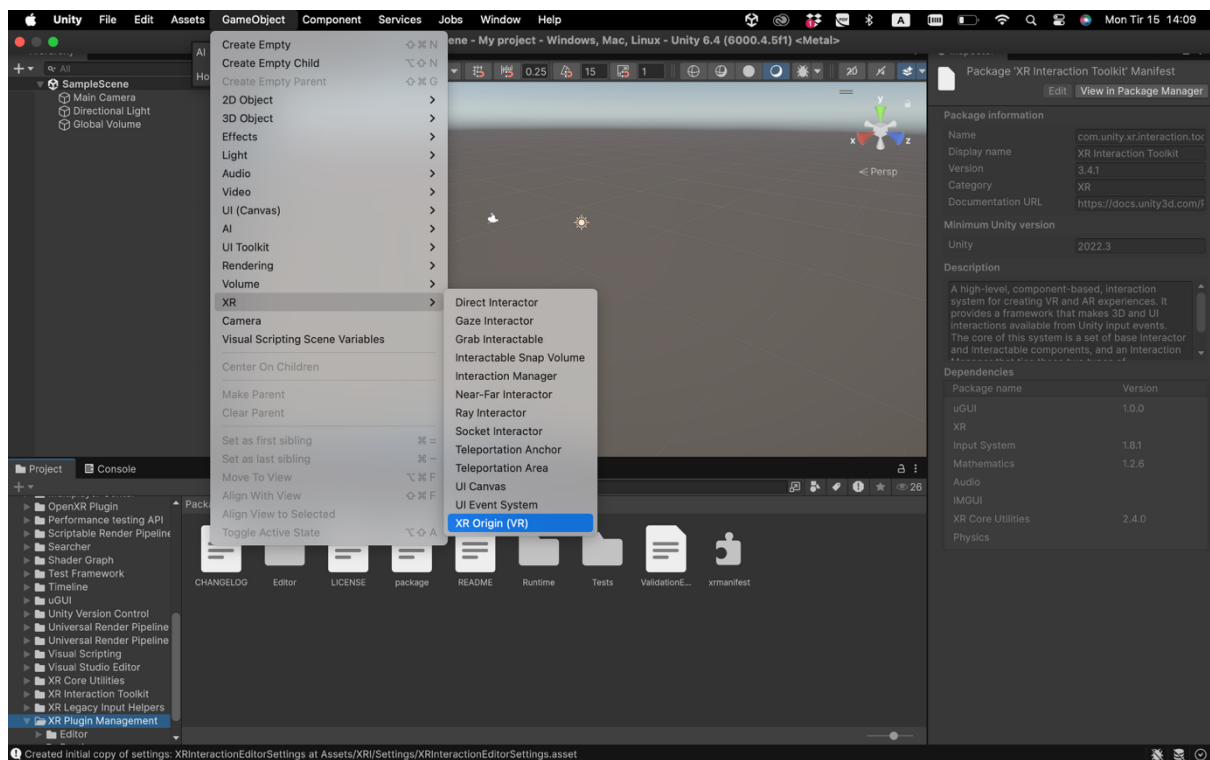
3) XR Origin / XR Rig (the “player body”)

The XR Rig is the player’s representation inside the scene.

It usually contains:

- Main Camera → the user’s head/eyes in VR
- Left Controller and Right Controller → the user’s hands
- A “floor offset” object (often called Camera Offset) that helps set the correct player height

Go to GameObject → XR → click on **XR Origin (VR)**



What it controls:

- Where the player starts in the world
- How high the player is (floor vs seated mode)
- How the camera and controllers move with tracking

2.3 Player Movement

Goal:

Be able to move around the VR environment comfortably using the controller.

Teleportation in Unity requires three main components:

1. Teleportation Provider
2. Teleportation Area
3. Configure the Controller

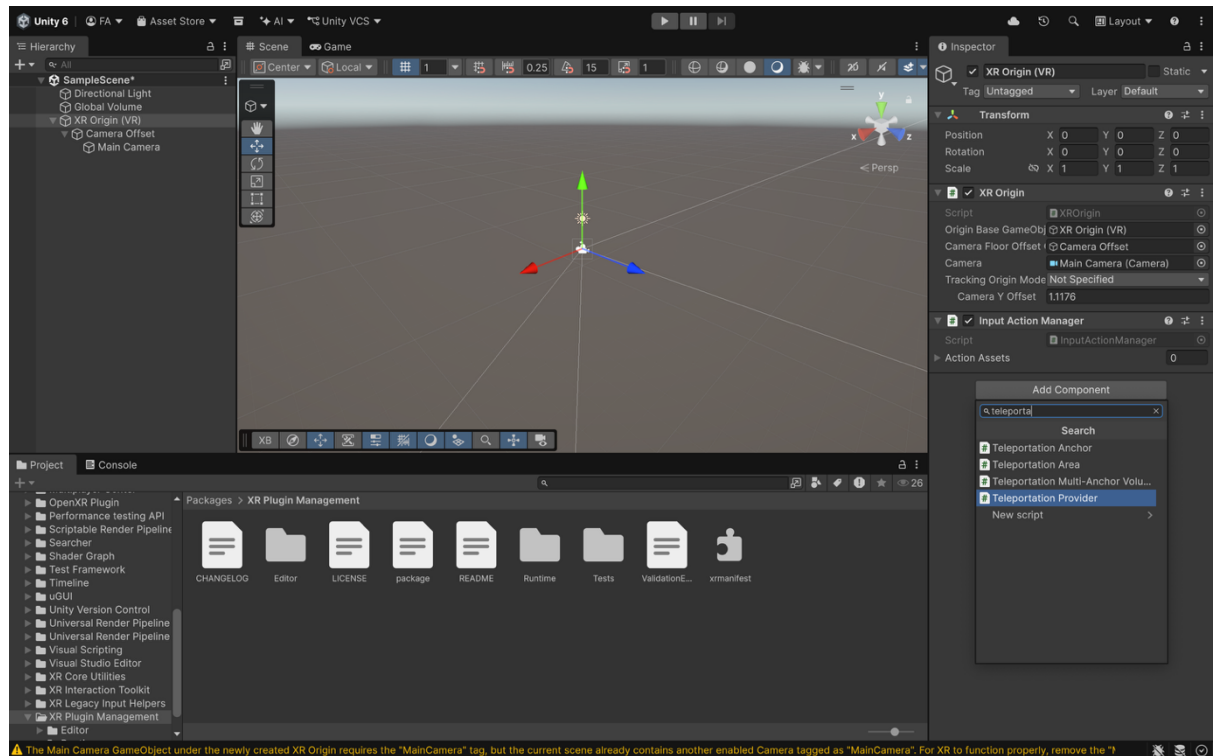
2.3.1 Teleportation Provider

The Teleportation Provider is responsible for moving the XR Origin from one location to another after the user selects a valid teleport destination.

1. Select **XR Origin (VR)**.
2. In the **Inspector**, look for components such as:
 - **Teleportation Provider**

If you **don't** see a teleportation provider, then:

- Click **Add Component**.
- Search for **Teleportation Provider**.
- Add it.



2. 3.2 Teleportation Area

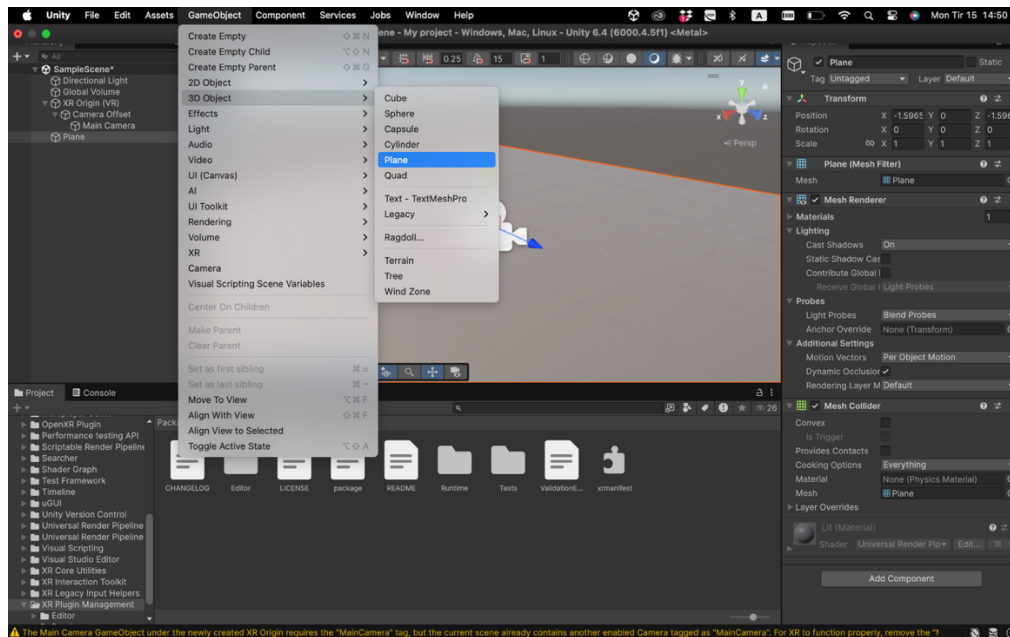
A **Teleportation Area** defines the surfaces where users are allowed to teleport.

Usually this is applied to the floor.

Step 1 – Create a Floor

Create a floor by selecting:

GameObject → 3D Object → Plane

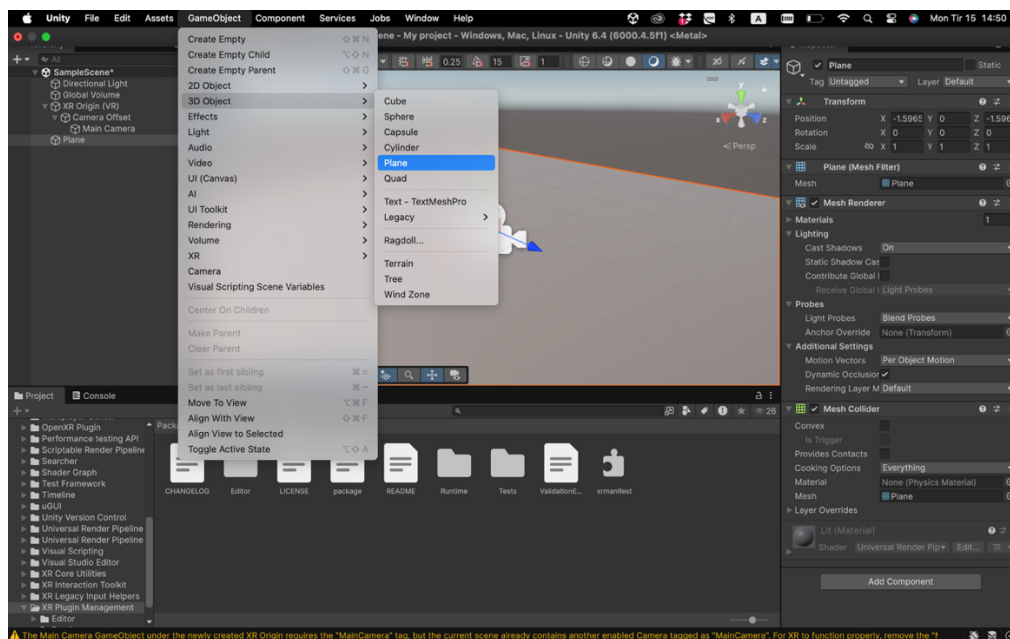


Step 2 – Add the Teleportation Area

Select the Plane. Click: **Add Component**

Search for: **Teleportation Area**

Add the component.



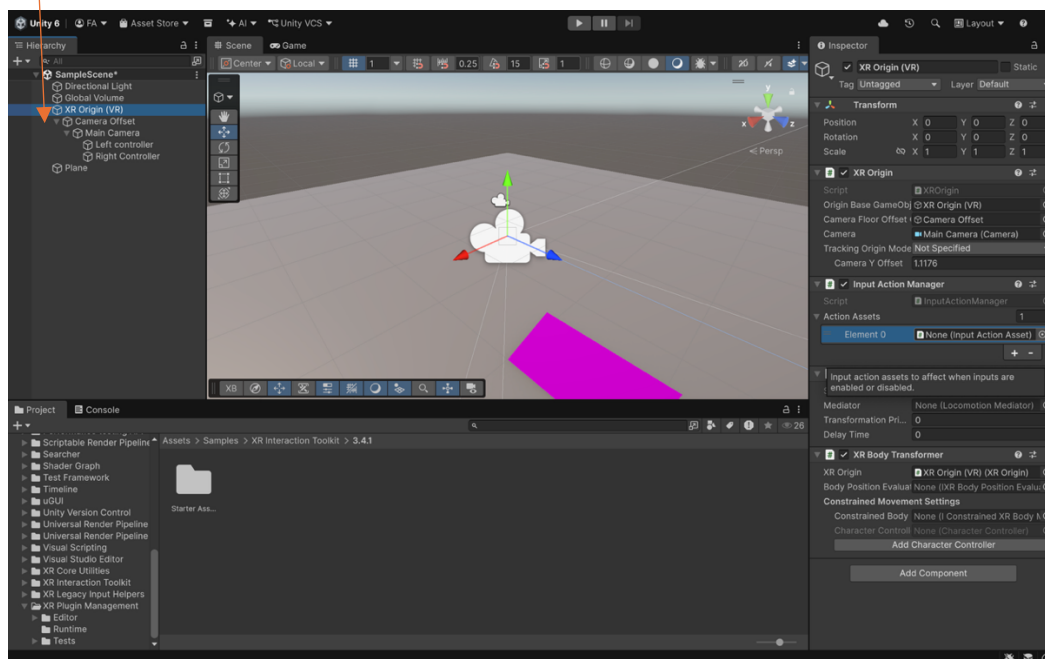
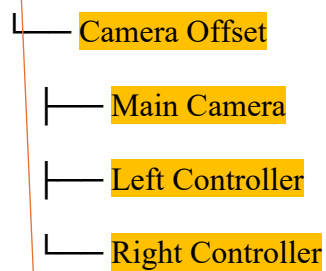
2.3.3 Configure the Controller

Expand the XR Origin. Create controllers:

1. Right-click **Camera Offset**
2. Choose **Create Empty**
3. Name it **Left Controller**
4. Create another empty object
5. Name it **Right Controller**

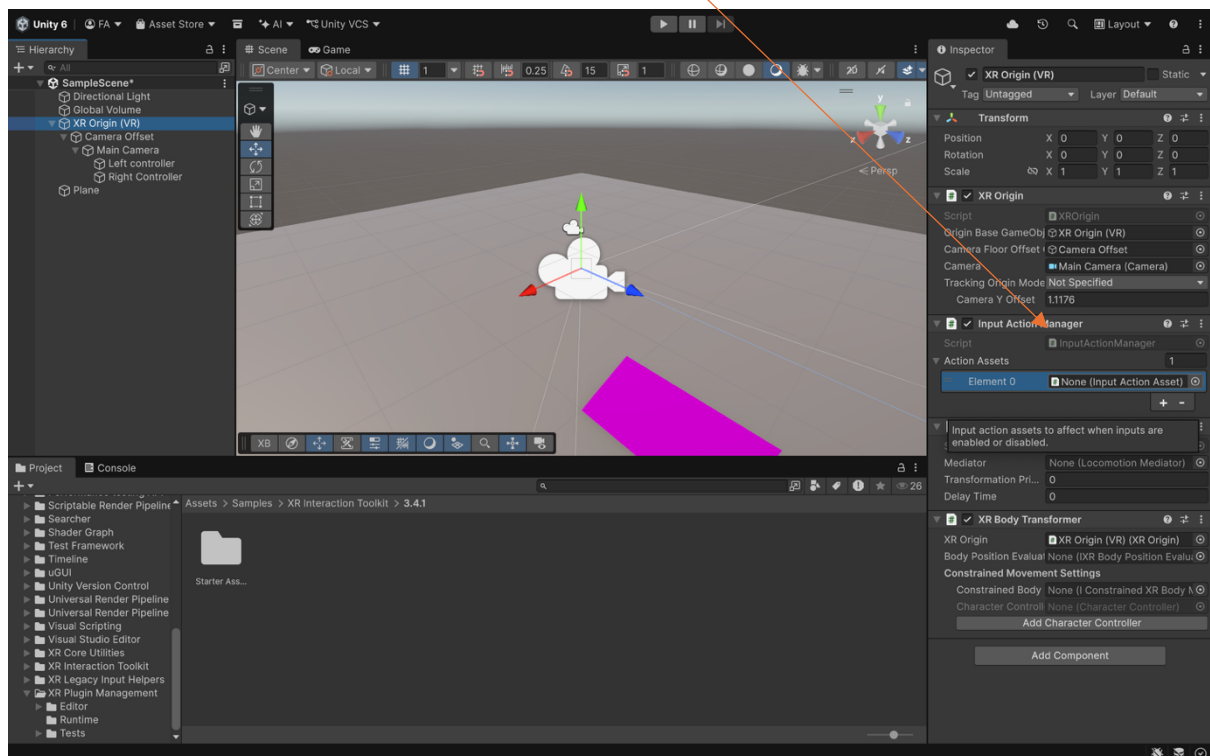
Your hierarchy should become:

XR Origin (VR)



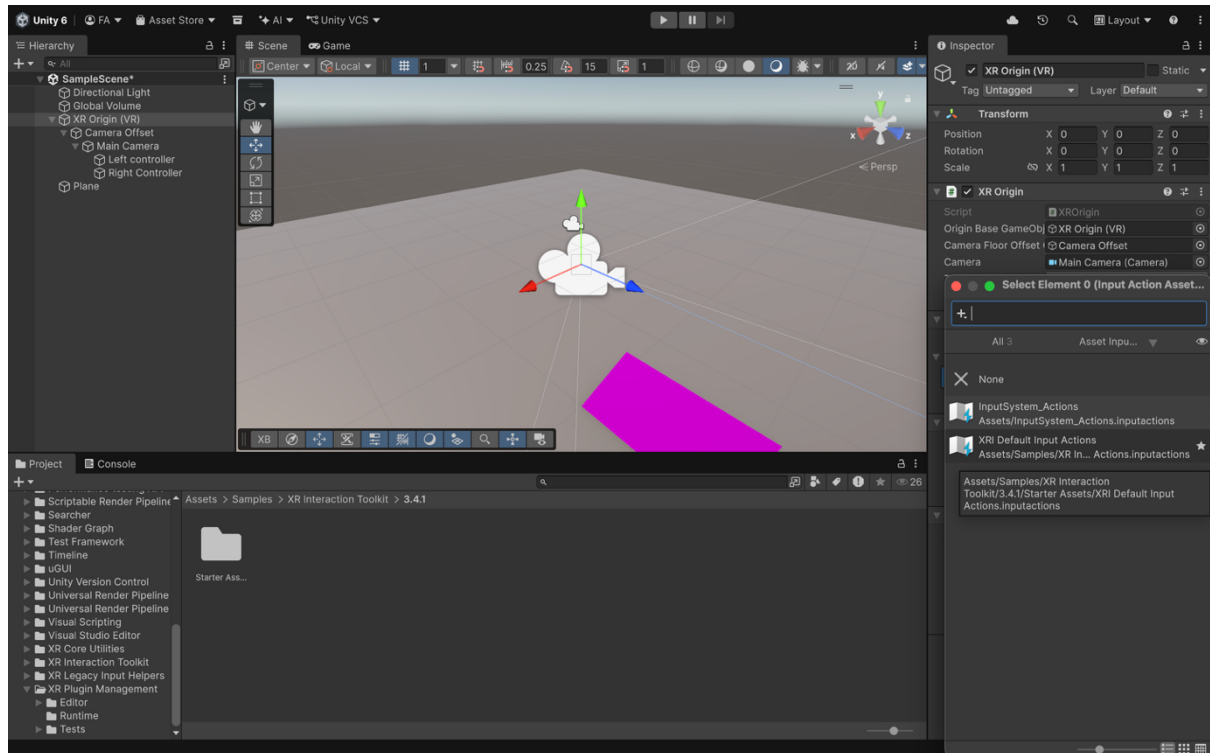
2.3.4 Configure Input Actions

1. Select **XR Origin (VR)** in the Hierarchy.
2. In the Inspector, find **Input Action Manager**.



Click the small **circle icon** next to **None** and select:

XRI Default Input Actions



2.3.5 Add the Teleport Ray

Select **Left Controller**, then add these components:

- **XR Ray Interactor**
- **XR Interactor Line Visual**

The left controller is commonly used for teleportation.

2.3.6 Configure Locomotion

Select **XR Origin (VR)** and add:

Locomotion Mediator

2.3.7 Test Teleportation

1. Enter Play Mode.
2. Activate the teleport ray using the assigned controller button.
3. Aim at the floor.
4. Release the button to teleport.

If configured correctly:

- A curved ray appears.
- Valid teleport locations are highlighted.
- Releasing the teleport button instantly moves the player.

2.4 Importing 3D Assets

Goal:

Import 3D models into Unity, place them in the scene, and make sure they look correct (scale, materials, organization).

2.4.1 Supported File Formats

- GLB / glTF: good for props and environments (often includes textures).
- FBX: best for characters/animation (Unity-native).
- OBJ: okay for simple static objects (materials may need manual setup).

2.4.2 Import Assets

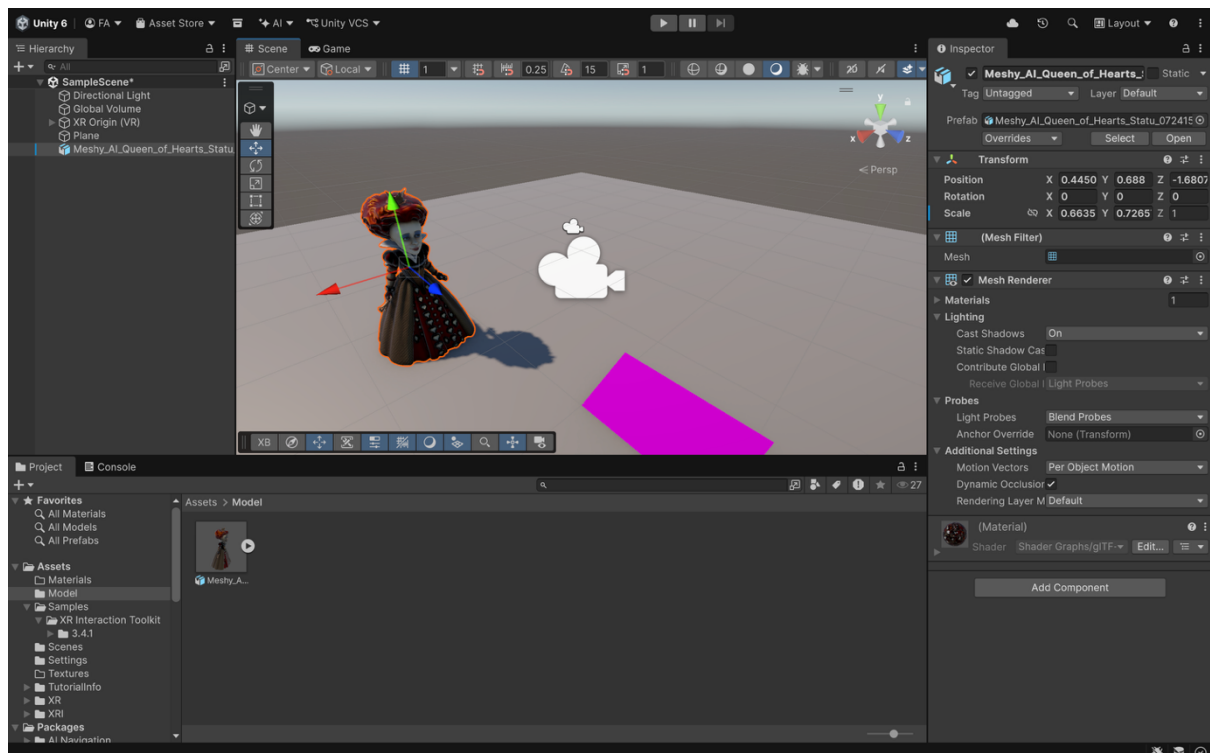
In Unity Project window, create folders:

- Assets/Models
- Assets/Textures
- Assets/Materials

Drag your model file (GLB/FBX/OBJ) into Assets/Models.

If using GLB:

Make sure you have glTFast installed in Unity. After importing, Unity creates a prefab-like asset you can drag into the scene.



2.4.3 Place Assets in the Scene

Drag the imported model from the Project into the Hierarchy or directly into the Scene.

Use W/E/R to position/rotate/scale:

- W move
- E rotate
- R scale

2.5 Creating Interactive Objects: grab, hold, and place

Goal:

Make a 3D object grabbable in VR so the user can pick it up, move it, and place it down.

2.5.1 Prepare the Object

Select the object you want to grab (example: teapot).

It must have:

- Collider (so Unity can detect touch/selection)
- Rigidbody (so physics works)
- XR Grab Interactable (so controllers can grab it)

2.5.2 Configure the Collider

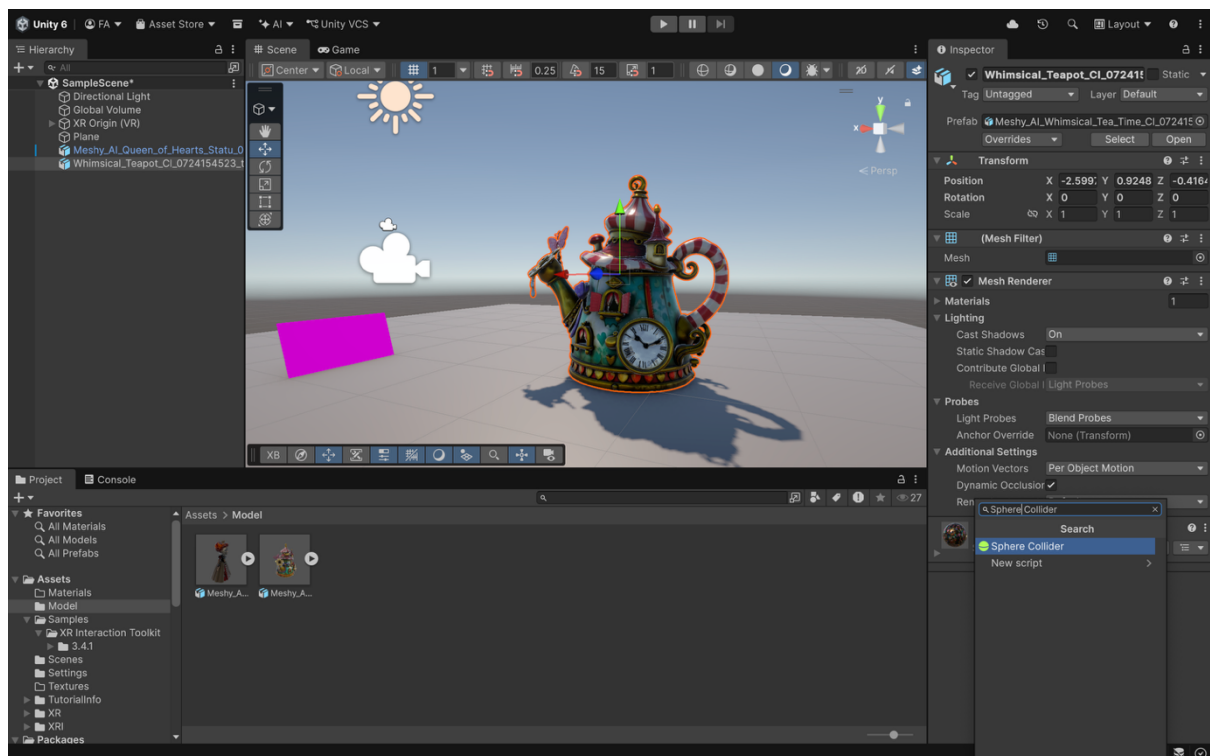
With the object selected:

Add Component → Box Collider (easy for most props) or Sphere/Capsule Collider
(good for rounded objects)

For irregular objects (like a teapot):

Use multiple simple colliders (compound colliders):

- body = sphere
- spout = capsule
- handle = 1–2 capsules



Put colliders on child objects, but keep only one Rigidbody on the root.

Important:

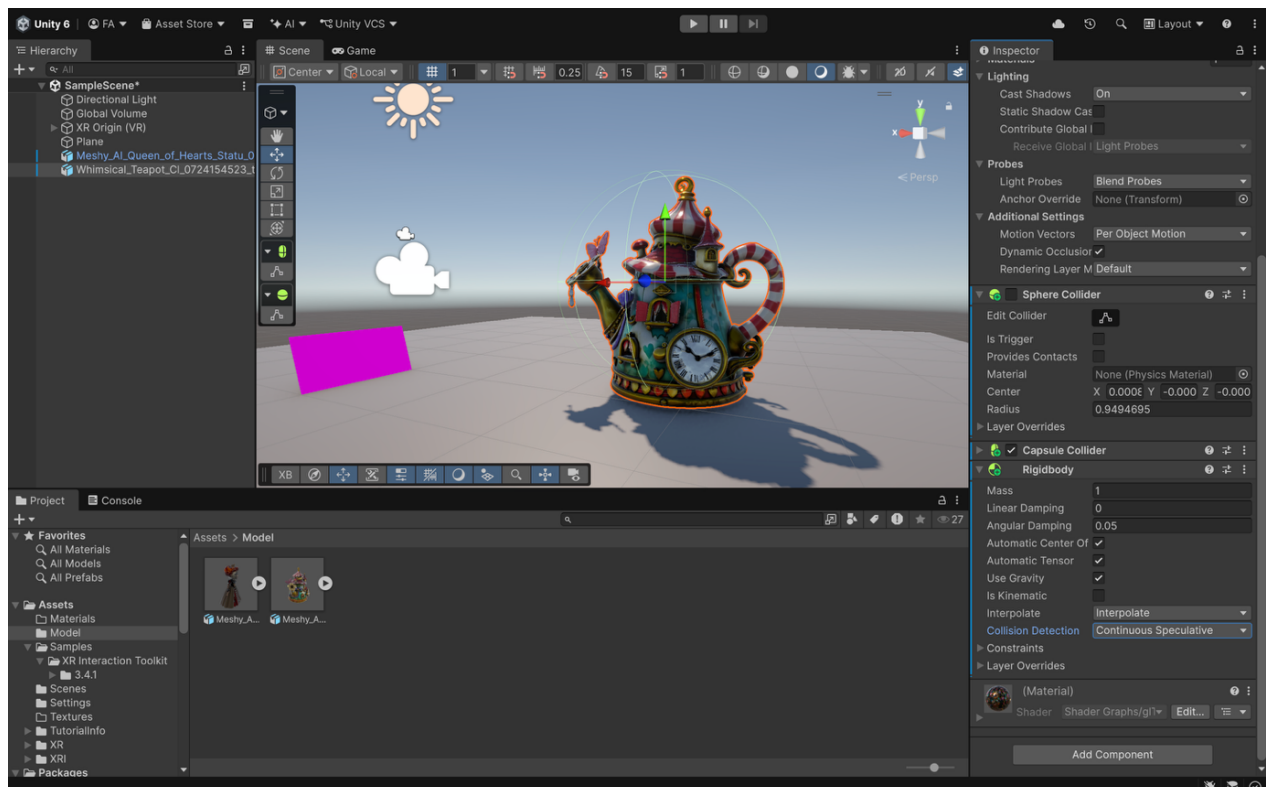
Colliders on grabbable props should be Is Trigger = OFF.

2.5.3 Configure the Rigidbody

Add Component → Rigidbody

Recommended settings:

- Use Gravity = ON
- Is Kinematic = OFF
- Interpolation = Interpolate
- Collision Detection = Continuous Speculative



2.5.4 Add an XR Grab Interactable

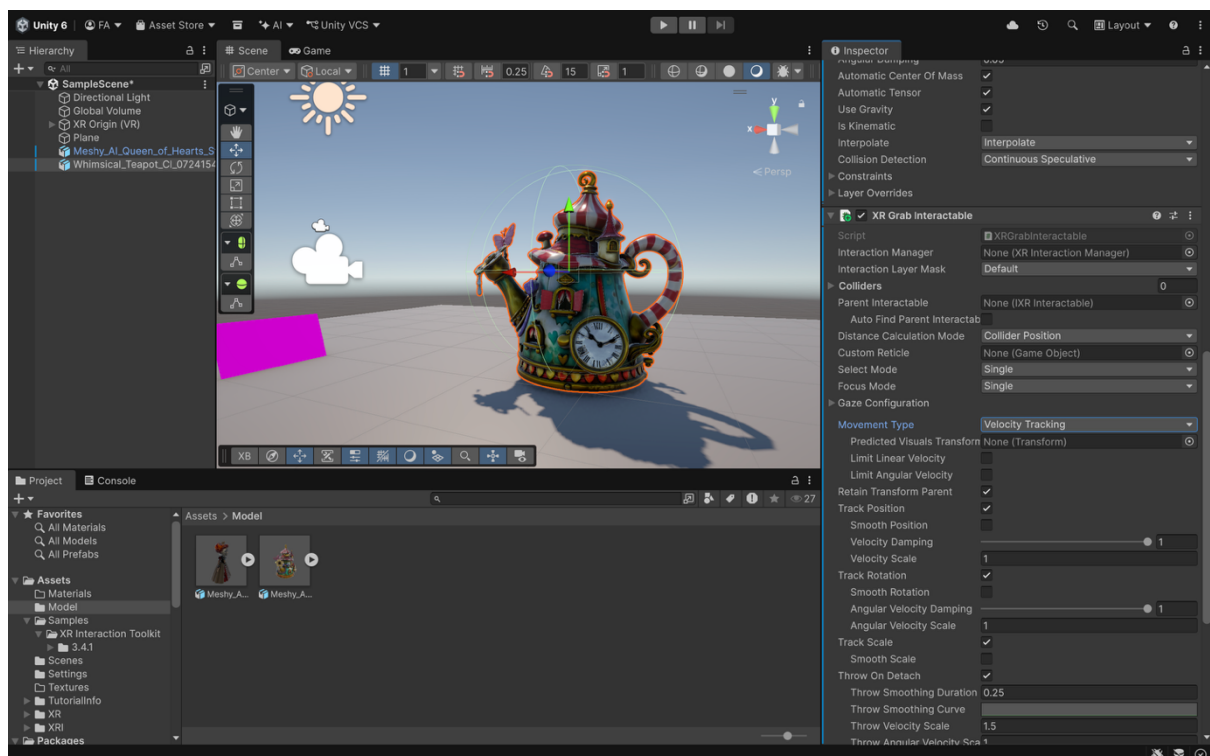
Add Component → XR Grab Interactable (Action-based)

Recommended settings:

Movement Type: start with Velocity Tracking (often drops/places more naturally)

(If jittery, try Kinematic instead.)

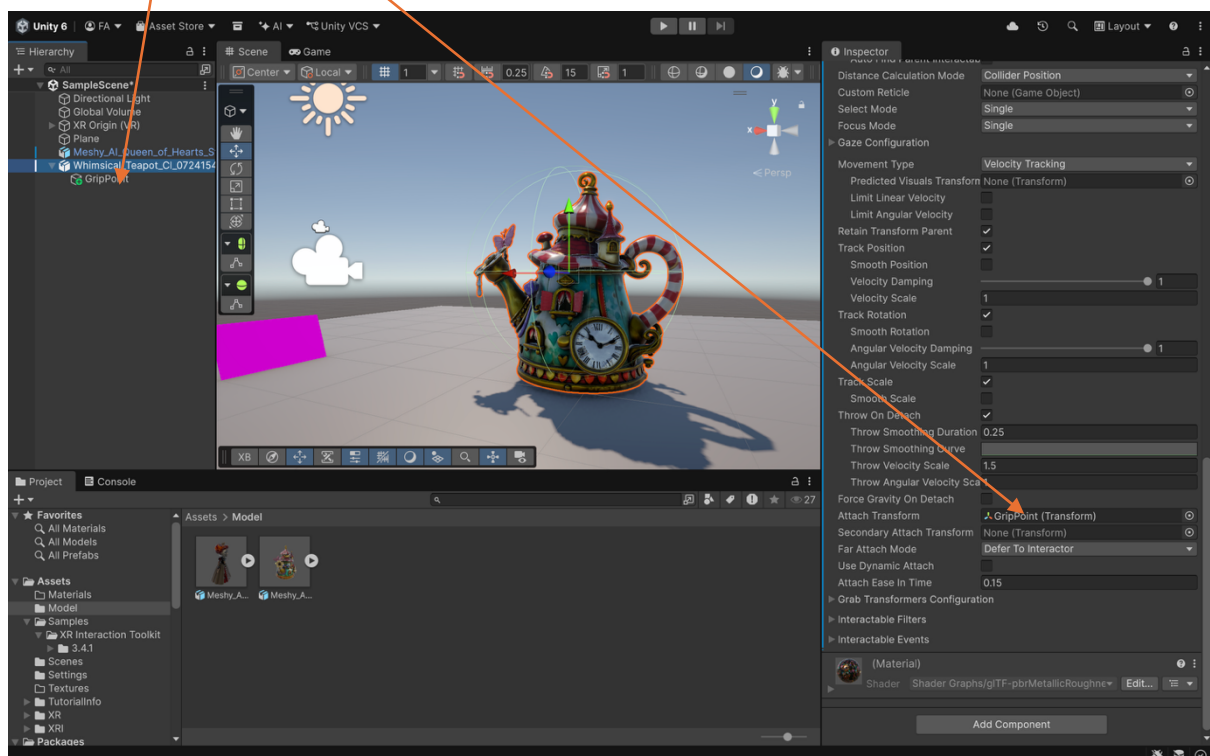
Optional: Throw On Detach = ON (if you want throwing)



2.5.5 Make the grab feel natural (Attach Transform)

- Right-click the object → Create Empty → name it GripPoint
- Move GripPoint where the hand should hold the object (e.g., near the handle)
- In XR Grab Interactable → Attach Transform, drag in

GripPoint



2.5.6 Test the Interaction

1. Enter Play Mode.
2. Move your controller close to the object.
3. Press the Grab button.
4. The object should attach to your hand.
5. Release the Grab button to drop it.

If the object does not grab, check that:

- The object has a Collider.
 - The object has a Rigidbody.
 - The object has an XR Grab Interactable.
 - The controller has an XR Direct Interactor.
-

Congratulations! You have completed the foundations of designing and developing educational virtual reality experiences in Unity.

Throughout this guide, you explored the key principles of educational VR design, learned how to plan effective learning activities, and gained practical experience configuring a Unity project for virtual reality. You also learned how to set up player movement, import 3D assets, and create interactive objects that support engaging learning experiences.

Developing educational VR applications is an iterative process. As you continue building your own projects, remember that effective VR experiences are created by combining sound instructional design with intuitive interaction, meaningful feedback, and user-centered design. Regular testing with learners and refining your activities based on their feedback will help create more engaging and effective immersive learning experiences.

The skills introduced in this guide provide a strong foundation for developing educational VR applications. From here, you can continue exploring more advanced Unity features such as animations, user interfaces, audio, scripting, multiplayer interactions, and learning analytics to create richer immersive learning environments.

We hope this guide serves as a useful starting point for your journey into educational virtual reality development.