Two VR Exercises

Meta XR SDK for Unity • Quest 3 • Step-by-step tutorial deck

Exercise 1: VR basics, grab and throw
Exercise 2: World-space VR UI

Individual Meta packages, not the All-in-One

Start from a clean Unity project

1 **Create project**

Use a 3D or URP 3D template.

2 **Switch platform**

File → Build Settings → Meta → Switch Platform.

3 **Enable XR**

Say yes to OpenXR installation popup

Install only the Meta packages you need from Package manager

This path avoids the All-in-One wrapper while keeping the project small and focused.

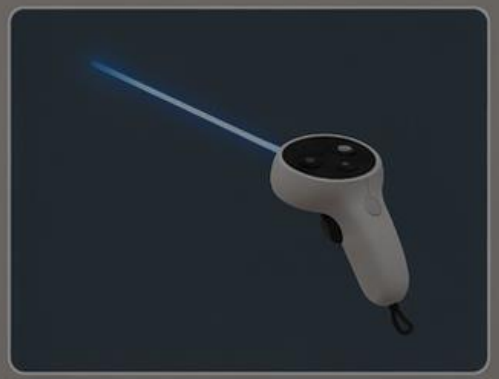
Meta XR Core SDK	`com.meta.xr.sdk.core`	Base Quest support: rendering, tracking, OVR camera rig, OVRInput, project setup tools.
Meta XR Interaction SDK	`com.meta.xr.sdk.interaction`	Grab, ray, poke, hand/controller interaction building blocks.
Meta XR Interaction SDK OVR Integration	`com.meta.xr.sdk.interaction.ovr`	Connects Interaction SDK to OVRPlugin / Utilities; useful with OVR-based rigs and prefabs.

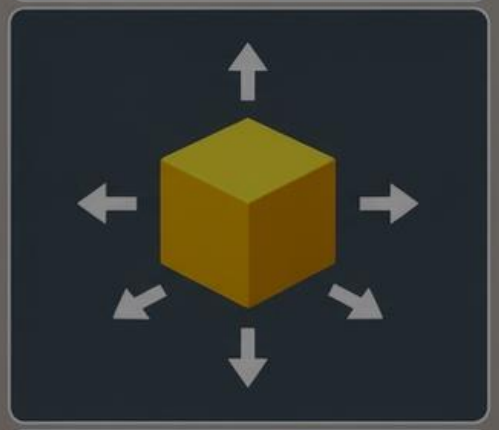
Meta will ask to Fix Player settings, say yes and fix all settings

EXERCISE 1

Pick Up Cubes in VR

- Move your head in VR
- See controllers / hands
- Grab the cube and release





Tutorial from:

https://developers.meta.com/horizon/documentation/unity/unity-tutorial-hello-vr/

Step 8 — Build and run on Quest 3

Test on-device early, especially for input and performance.

1 Connect headset

USB cable → accept USB debugging and data access prompts in the headset.

2 Add scene

File → Build Settings → make sure Exercise_01_BasicVR is in Scenes In Build.

3 Build And Run

Platform: Android → Build And Run. First build may take longer.

Goal: you should stand in the room, see the cube, grab it.

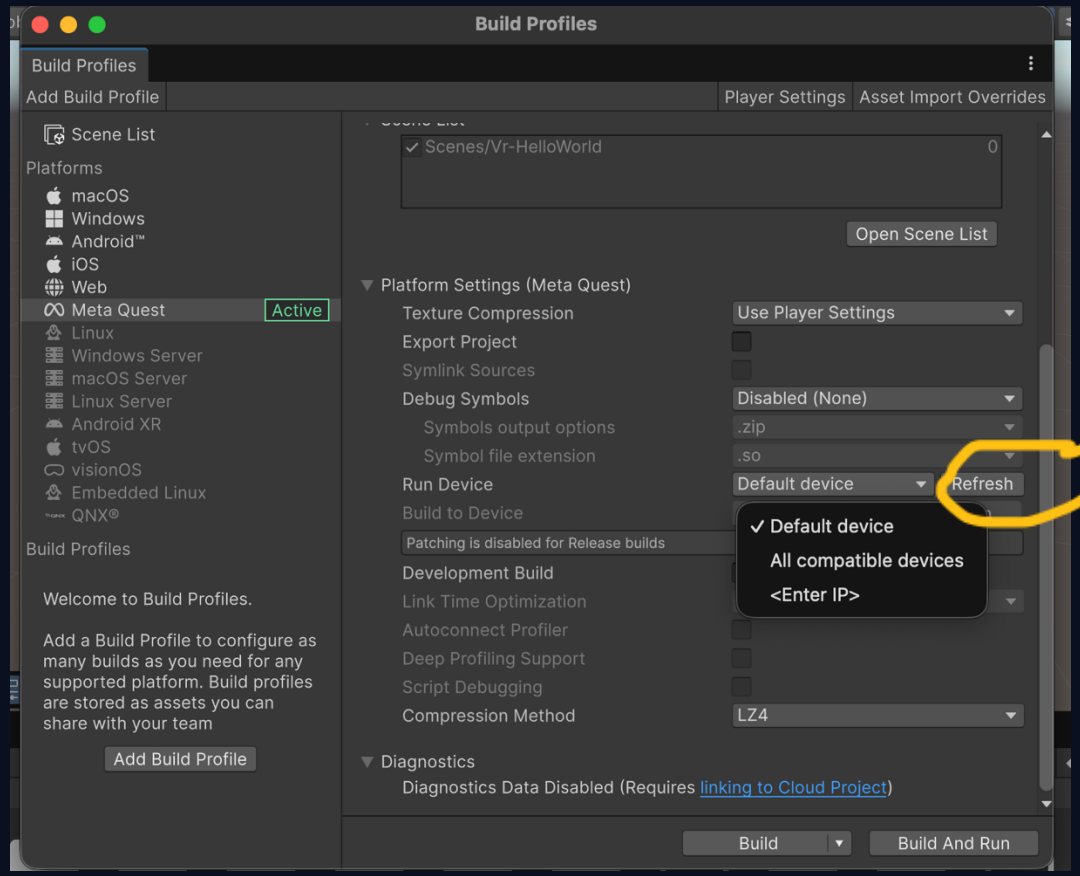
Exercise 1 completion criteria

- The scene launches in VR on Quest 3.
- The controller or hand model is tracked.
- Head tracking changes the view naturally.
- The cube can be grabbed and released.

Install Meta Horizon Link to run on the Headset from Unity Editor: https://www.meta.com/it-it/help/quest/1517439565442928/

Or Build it on the Headset:

1. Plug in the Headset on the PC, allow permissions on the headset
2. Go to File → Buil Profiles
3. In the section Run Device choose your headset name, it it does not appear click on Refresh and then select it
4. Click on Build and Run, this process will install the apk on the headset and will also save a copy of the apk in your PC



Some Useful Documentation

Ray Interactions: https://developers.meta.com/horizon/documentation/unity/unity-isdk-create-ray-interactions/

Throw an object: https://developers.meta.com/horizon/documentation/unity/unity-isdk-throw-object/

Poke interactions: https://developers.meta.com/horizon/documentation/unity/unity-isdk-create-poke-interactions/

Distance grab (magnet effect on controller with ray): https://developers.meta.com/horizon/documentation/unity/unity-isdk-create-distance-grab-interactions-legacy/

Hand interactions: https://developers.meta.com/horizon/documentation/unity/unity-isdk-hand-grab-interaction/

EXERCISE 2

Build a VR World-Space UI

Exercise 2 steps: Create a World Space Canvas and add it as cube child

- Create a Canvas
- Set it as World Canvas
- Set the size as 0, 0

- Add a text as children of the Canvas (import TextMeshPro if promped)

- Write something you like inside the text

- Move the canvas inside the cube that we did for the previous exercise --> the Canvas will become a child of the cube

- Try to run in on the Headset

Some Useful Documentation

UI Interactions: https://developers.meta.com/horizon/documentation/unity/unity-isdk-poke-interaction/

Ray interaction with UI: https://developers.meta.com/horizon/documentation/unity/unity-isdk-use-ray-with-ui/

Ray castable UIs: https://developers.meta.com/horizon/documentation/unity/unity-isdk-create-raycast-ui/

Curved and flat UIs: https://developers.meta.com/horizon/documentation/unity/unity-isdk-create-ui/

Curved UI: https://developers.meta.com/horizon/documentation/unity/unity-isdk-curved-canvases/

VR UI constraints and guidelines: think spatial, not screen-based

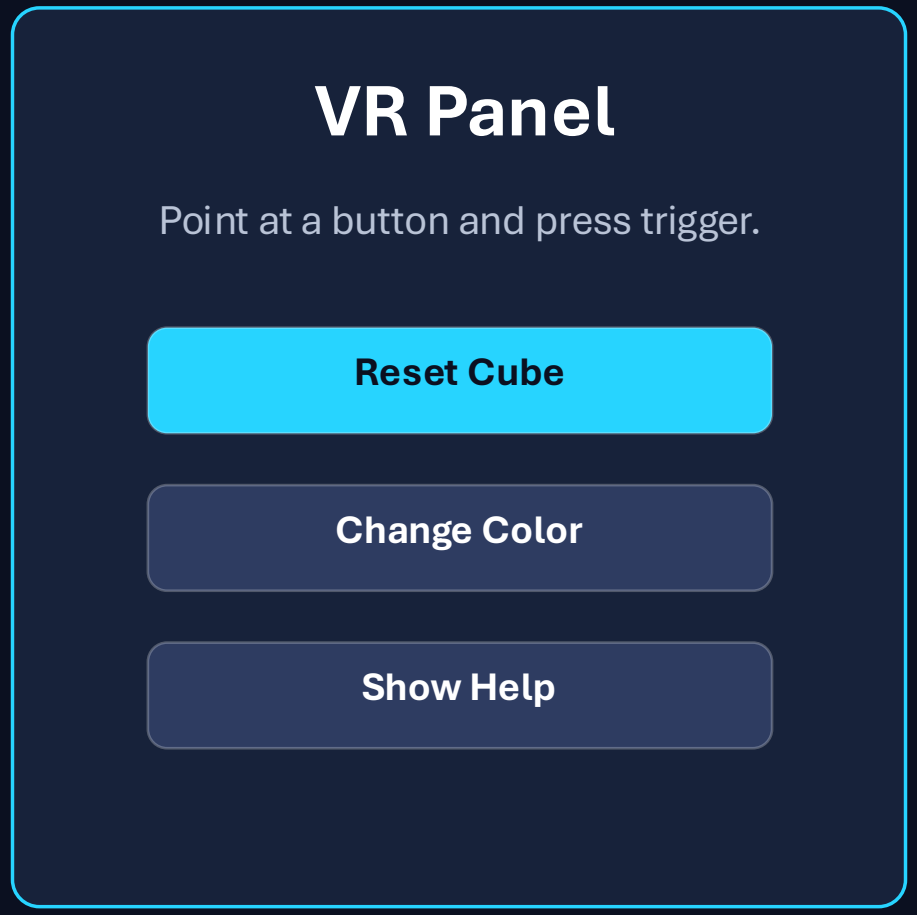
A VR menu is a 3D object. Design it as something the user can approach, read and point at.

- Use World Space Canvas for VR menus.
- Avoid tiny text and dense paragraphs.
- Place the panel about 1.2–2 m away.
- Do not attach big menus permanently to the head.
- Give hover, press and state-change feedback.

Source: Unity XR docs describe world-space Canvas setup and XR UI input modules for XR interaction.

Curved and Plane example of UI

UI can be **Plane** such as an UI canvas on a wall or on a plane object; or **Curved**, such as curved menu or settings panels



Design rules

- Use high contrast.
- Avoid transparent-only panels.
- Leave generous spacing.
- Label buttons with verbs.
- Keep one idea per line.

Curved UI documentation: https://developers.meta.com/horizon/documentation/unity/unity-isdk-curved-canvases/
Curved UI tutorial: https://developers.meta.com/horizon/documentation/unity/unity-isdk-create-ui/