Schedule Semester

DAY 1 (07.11.)

- Introduction
- Examples
- Overview History Immersive Technologies
- Remediation, Immediacy, Hypermediacy (Bolter, Grusin)
- Unity Introduction

DAY 2 (10.11.)

- Video Game Spaces (Nitsche)
- Case Studies XR
- Design Challenges AR/VR
- Personal Projects (Find Group & Device)
- Unity Introduction

DAY 3 (17.11.)

- Mini-Pitches & Discussion
- Project Setup (AR/VR)
- Interaction Examples for AR / VR in Unity
- Work on Projects

DAY 4 (28.11.)

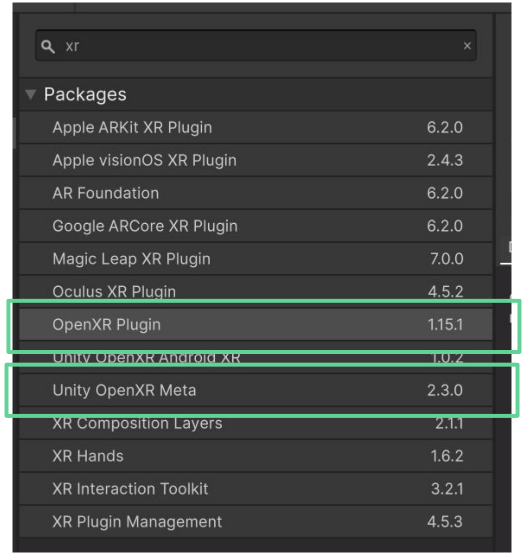
- requested topics, specific questions concerning technical and dramaturgical questions of your projects

DAY 5 (15.12.)

- Project work and final problem solving

Unity Setup for Virtual Reality

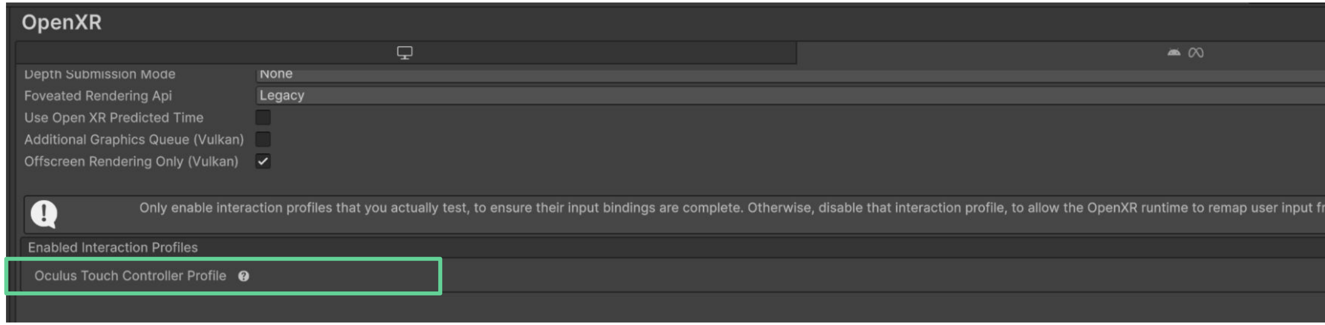
Installations & Settings



xr

Packages

Apple ARKit XR Plugin	6.2.0
Apple visionOS XR Plugin	2.4.3
AR Foundation	6.2.0
Google ARCore XR Plugin	6.2.0
Magic Leap XR Plugin	7.0.0
Oculus XR Plugin	4.5.2
OpenXR Plugin	1.15.1
Unity OpenXR Android XR	1.0.2
Unity OpenXR Meta	2.3.0
XR Composition Layers	2.1.1
XR Hands	1.6.2
XR Interaction Toolkit	3.2.1
XR Plugin Management	4.5.3



OpenXR

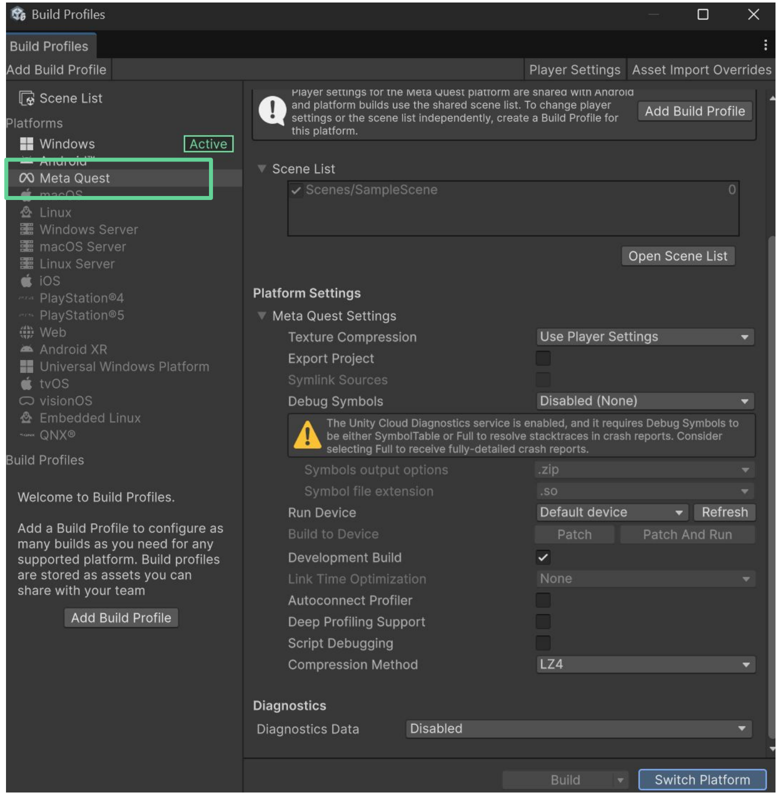
Depth Submission Mode	None
Foveated Rendering Api	Legacy
Use Open XR Predicted Time	☐
Additional Graphics Queue (Vulkan)	☐
Offscreen Rendering Only (Vulkan)	✓

Only enable interaction profiles that you actually test, to ensure their input bindings are complete. Otherwise, disable that interaction profile, to allow the OpenXR runtime to remap user input fr

Enabled Interaction Profiles

Oculus Touch Controller Profile

Installations & Settings



Welcome to Build Profiles.

Add a Build Profile to configure as many builds as you need for any supported platform. Build profiles are stored as assets you can share with your team

Player settings for the Meta Quest platform are shared with Android and platform builds use the shared scene list. To change player settings or the scene list independently, create a Build Profile for this platform.

Scenes/SampleScene 0

Platform Settings

Meta Quest Settings

Texture Compression	Use Player Settings
Export Project	
Symlink Sources	
Debug Symbols	Disabled (None)

The Unity Cloud Diagnostics service is enabled, and it requires Debug Symbols to be either SymbolTable or Full to resolve stacktraces in crash reports. Consider selecting Full to receive fully-detailed crash reports.

Symbols output options	.zip
Symbol file extension	.so
Run Device	Default device / Refresh
Build to Device	Patch / Patch And Run
Development Build	✓
Link Time Optimization	None
Autoconnect Profiler	
Deep Profiling Support	
Script Debugging	
Compression Method	LZ4

Diagnostics

Diagnostics Data	Disabled

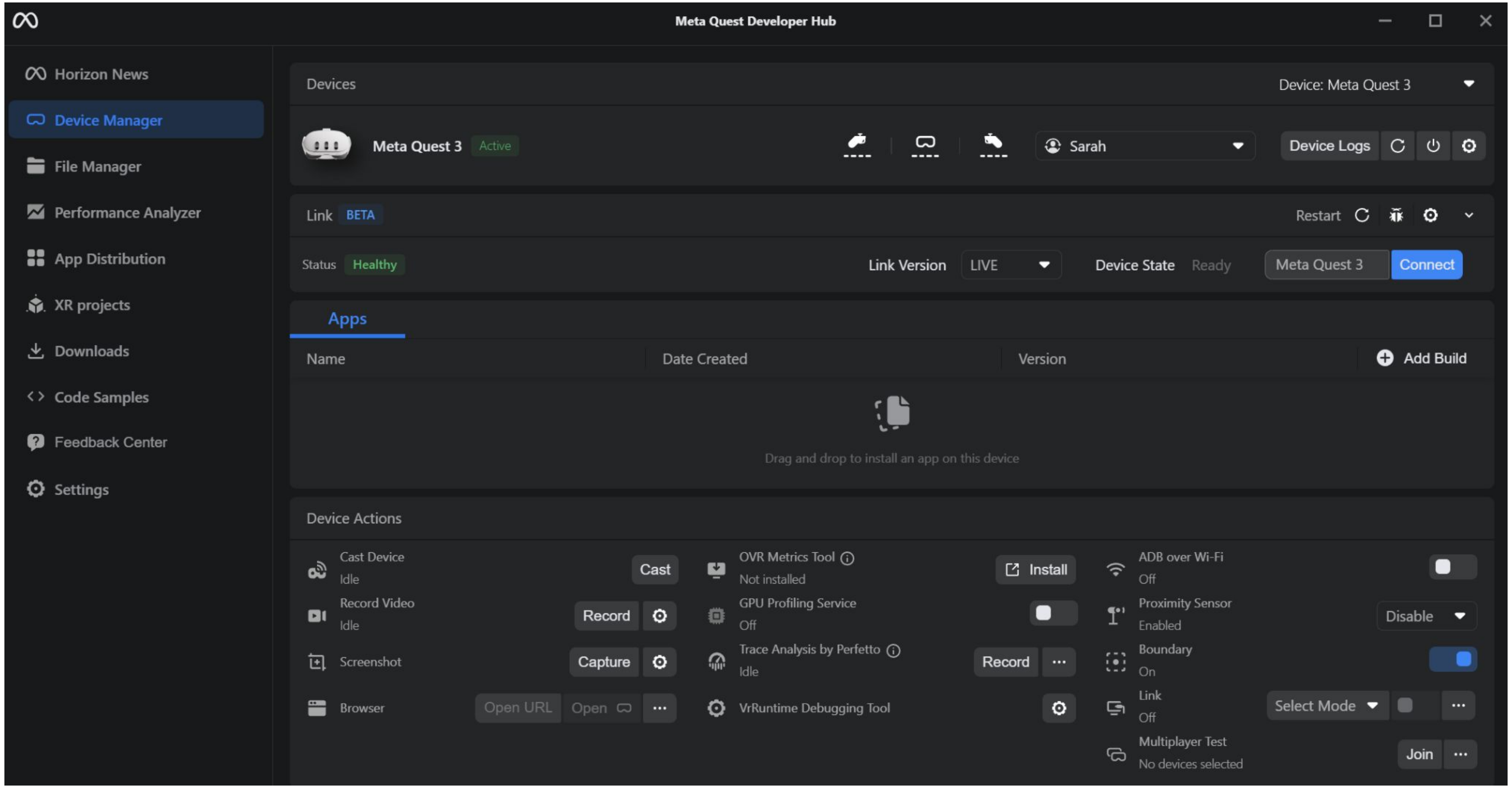
Setup Quest 3 Glasses

You will need a meta developer account

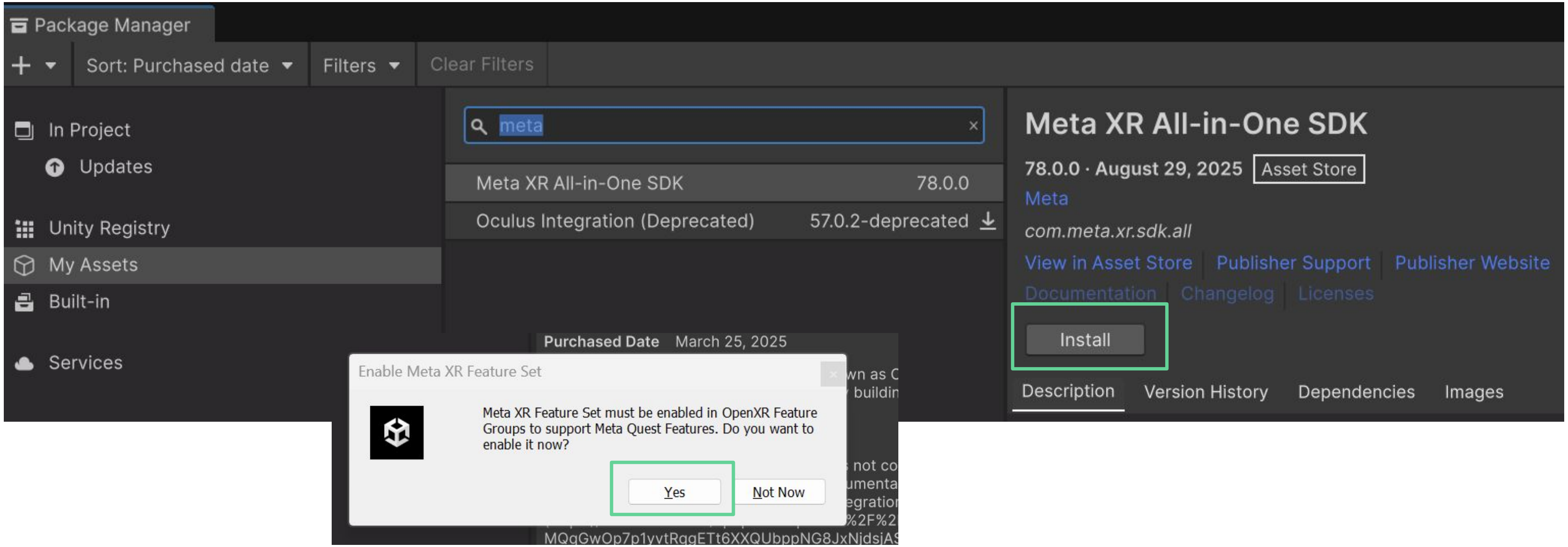
Enable development settings on glasses:
https://developers.meta.com/horizon/documentation/unity/unity-env-device-setup

Download Meta Quest Developer Hub on your Computer..

Meta Quest Developer Hub



Basic Scene Setup in Unity



Package Manager

Sort: Purchased date | Filters | Clear Filters

In Project
Updates
Unity Registry
My Assets
Built-in
Services

meta

Meta XR All-in-One SDK	78.0.0
Oculus Integration (Deprecated)	57.0.2-deprecated

Purchased Date March 25, 2025

wn as C
buildin
not co
umenta
egration
%2F%2
MQgGwOp7p1yvtRggETt6XXQUbppNG8JxNjdsjAS

Meta XR All-in-One SDK

78.0.0 · August 29, 2025 Asset Store

Meta

com.meta.xr.sdk.all

View in Asset Store | Publisher Support | Publisher Website

Documentation | Changelog | Licenses

Install

Description | Version History | Dependencies | Images

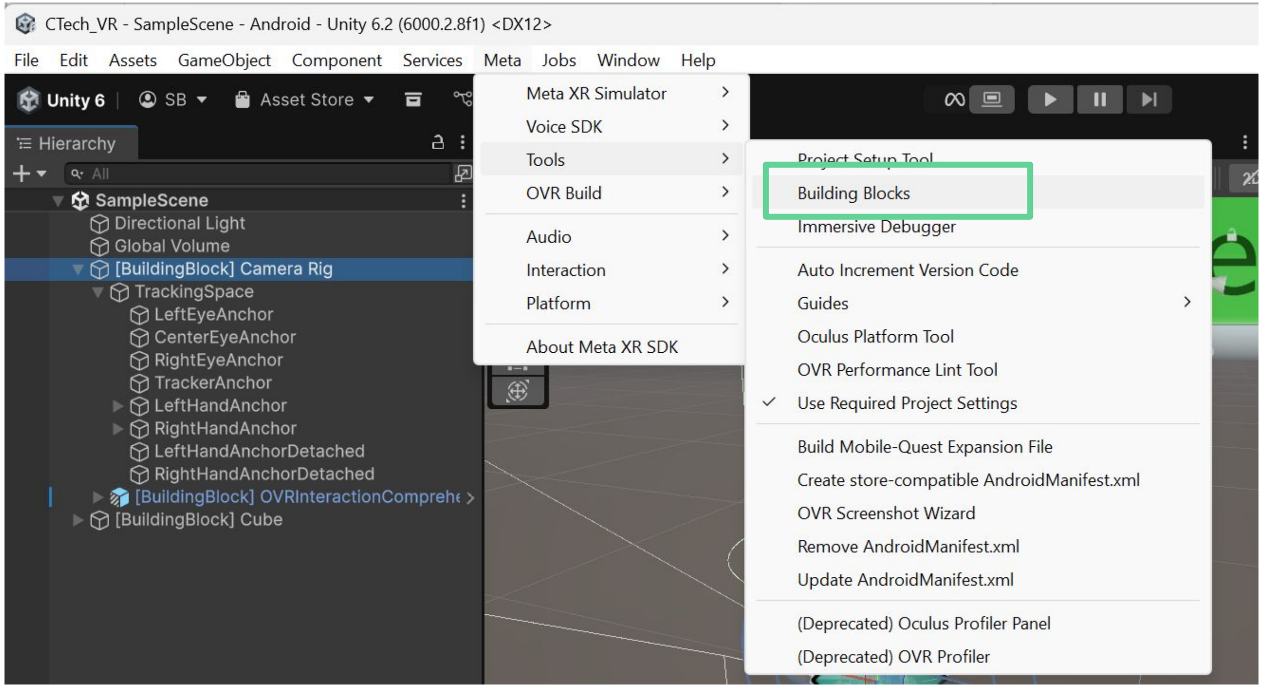
Enable Meta XR Feature Set

Meta XR Feature Set must be enabled in OpenXR Feature Groups to support Meta Quest Features. Do you want to enable it now?

Yes | Not Now

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

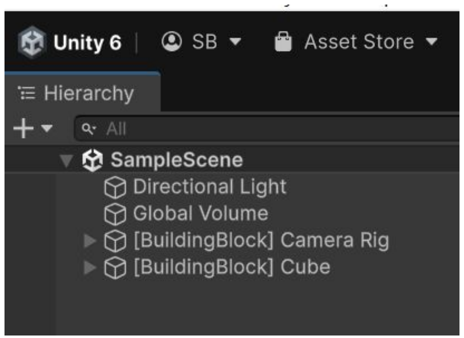
Basic Scene Setup in Unity


CTech_VR - SampleScene - Android - Unity 6.2 (6000.2.8f1) <DX12>
File Edit Assets GameObject Component Services Meta Jobs Window Help
Unity 6
SB
Asset Store
Hierarchy
All
SampleScene
Directional Light
Global Volume
[BuildingBlock] Camera Rig
TrackingSpace
LeftEyeAnchor
CenterEyeAnchor
RightEyeAnchor
TrackerAnchor
LeftHandAnchor
RightHandAnchor
LeftHandAnchorDetached
RightHandAnchorDetached
[BuildingBlock] OVRInteractionComprehe
[BuildingBlock] Cube
Meta XR Simulator
Voice SDK
Tools
OVR Build
Audio
Interaction
Platform
About Meta XR SDK
Project Setup Tool
Building Blocks
Immersive Debugger
Auto Increment Version Code
Guides
Oculus Platform Tool
OVR Performance Lint Tool
✓ Use Required Project Settings
Build Mobile-Quest Expansion File
Create store-compatible AndroidManifest.xml
OVR Screenshot Wizard
Remove AndroidManifest.xml
Update AndroidManifest.xml
(Deprecated) Oculus Profiler Panel
(Deprecated) OVR Profiler

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

Basic Scene Setup in Unity

- chose [BuildingBlock] Camera Rig
- chose [BuildingBlock] Cube

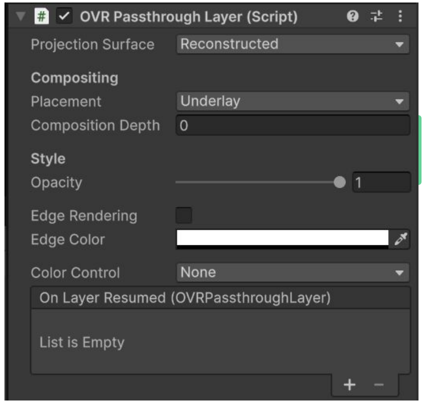


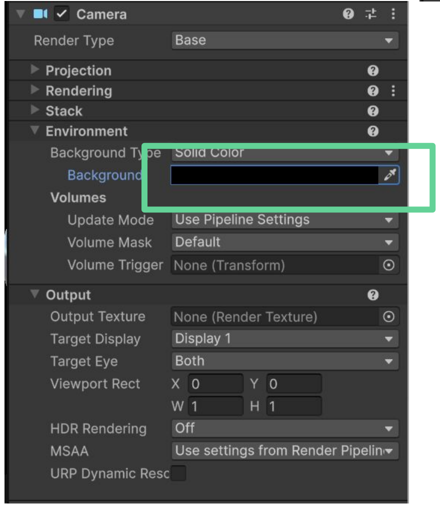
SampleScene
Directional Light
Global Volume
[BuildingBlock] Camera Rig
[BuildingBlock] Cube

Passthrough

if you want the passthrough enabled:

- chose [BuildingBlock] Passthrough
- in CenterEyeAnchor > Environment > BackgroundType > Solid Color black + alpha =0
- in PC_RP Asset (Universal Render Pipeline Asset) uncheck Terrain Holes and in Quality > uncheck HDR
- in Universal Render Data > disable Post-Processing

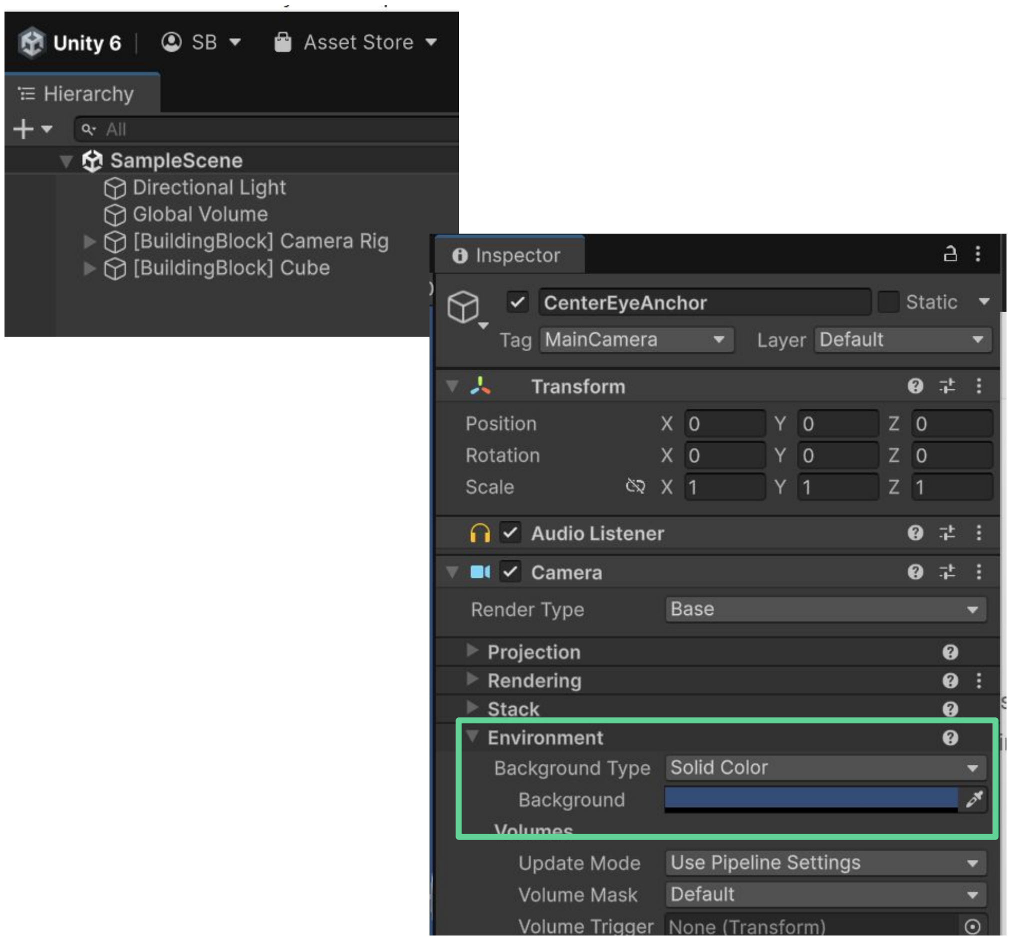

OVR Passthrough Layer (Script)
Projection Surface: Reconstructed
Compositing
Placement: Underlay
Composition Depth: 0
Style
Opacity: 1
Edge Rendering
Edge Color
Color Control: None
On Layer Resumed (OVRPassthroughLayer)
List is Empty


Camera
Render Type: Base
Projection
Rendering
Stack
Environment
Background Type: Solid Color
Background
Volumes
Update Mode: Use Pipeline Settings
Volume Mask: Default
Volume Trigger: None (Transform)
Output
Output Texture: None (Render Texture)
Target Display: Display 1
Target Eye: Both
Viewport Rect: X 0 Y 0 W 1 H 1
HDR Rendering: Off
MSAA: Use settings from Render Pipelin
URP Dynamic Resc

Disable Boundaries

On the glasses, go to Settings >Advanced > Developer > enable Developer Settings disable “Physical space features”

in the Meta Quest Developer Hub, you should now be able to disable Boundary



CenterEyeAnchor Static
Tag MainCamera Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Audio Listener

Camera
Render Type Base
Projection
Rendering
Stack
Environment
Background Type Solid Color
Background
Volumes
Update Mode Use Pipeline Settings
Volume Mask Default
Volume Trigger None (Transform)

Unity Setup for Augmented Reality

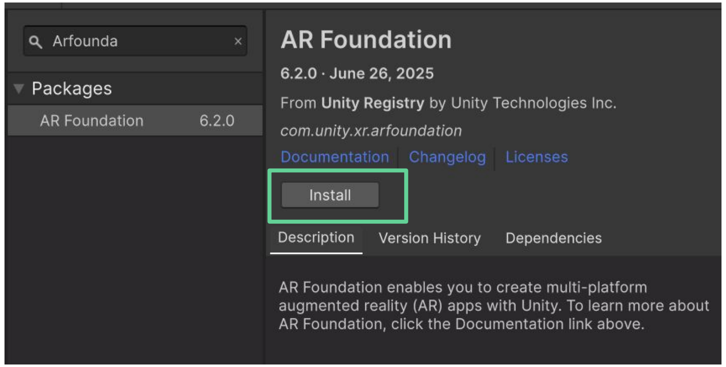
Advice

If you want to test both Setups, for AR and VR, I would recommend creating two separate Unity Projects. One for AR with ARFoundation and one for VR with Open XR Meta. This way you can keep the build and XR Settings without having to switch.

Installations & Settings

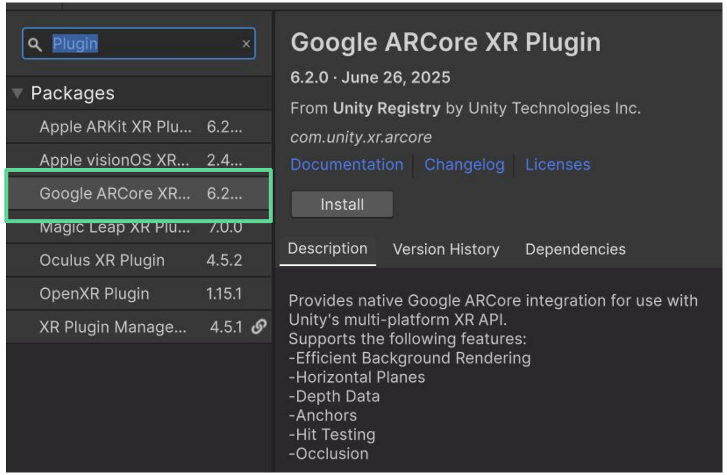
via package manager install:

- ARFoundation
- AR_Provider Plugin (depending on the platform you are developing for) as I will be developing for Android, I chose: Google ARCore XR Plugin



AR Foundation
6.2.0 · June 26, 2025
From Unity Registry by Unity Technologies Inc.
com.unity.xr.arfoundation

AR Foundation enables you to create multi-platform augmented reality (AR) apps with Unity. To learn more about AR Foundation, click the Documentation link above.


Plugin
Packages
Apple ARKit XR Plu... 6.2...
Apple visionOS XR... 2.4...
Google ARCore XR... 6.2...
Magic Leap XR Plu... 7.0.0
Oculus XR Plugin 4.5.2
OpenXR Plugin 1.15.1
XR Plugin Manage... 4.5.1
Documentation
Changelog
Licenses
Install
Description
Version History
Dependencies

Google ARCore XR Plugin
6.2.0 · June 26, 2025
From Unity Registry by Unity Technologies Inc.
com.unity.xr.arcore

Provides native Google ARCore integration for use with Unity's multi-platform XR API.
Supports the following features:
-Efficient Background Rendering
-Horizontal Planes
-Depth Data
-Anchors
-Hit Testing
-Occlusion

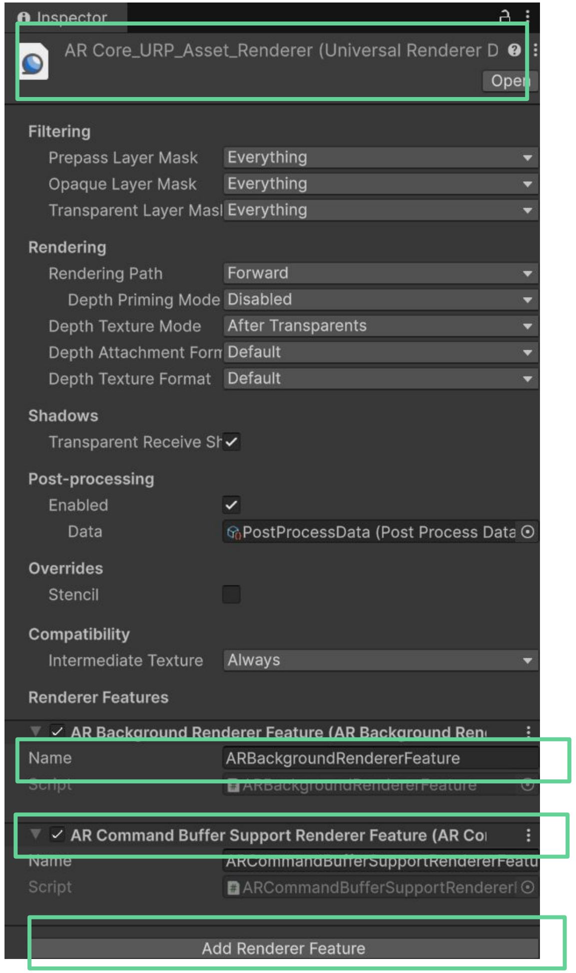
https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.3/manual/project-setup/install-arfoundation.html

URP Render Pipeline

if you are using URP (Universal Render Pipeline) you need an additional tweak:

- create a RenderPiplineAsset and its Renderer: in Assets > create > Rendering > “URP Asset (with Universal Renderer)
- add AR Background Renderer Feature
- add AR Command Buffer Support Renderer Feature

And do the same for the already existing “Mobile_Renderer” in your Settings folder



AR Core_URP_Asset_Renderer (Universal Renderer D

Filtering

Prepass Layer Mask	Everything
Opaque Layer Mask	Everything
Transparent Layer Masl	Everything

Rendering

Rendering Path	Forward
Depth Priming Mode	Disabled
Depth Texture Mode	After Transparents
Depth Attachment Forn	Default
Depth Texture Format	Default

Shadows

Transparent Receive Sh	✓

Post-processing

Enabled	✓
Data	PostProcessData (Post Process Data

Overrides

Stencil	

Compatibility

Intermediate Texture	Always

Renderer Features

✓ **AR Background Renderer Feature (AR Background Ren**

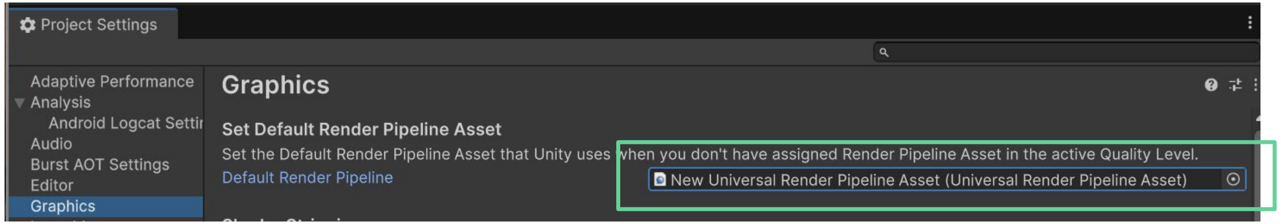
Name	ARBackgroundRendererFeature
Script	ARBackgroundRendererFeature

✓ **AR Command Buffer Support Renderer Feature (AR Co**

Name	ARCommandBufferSupportRendererFeatu
Script	ARCommandBufferSupportRendererI

https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.3/manual/project-setup/universal-render-pipeline.html

AR Settings



Graphics

Set Default Render Pipeline Asset

Set the Default Render Pipeline Asset that Unity uses when you don't have assigned Render Pipeline Asset in the active Quality Level.

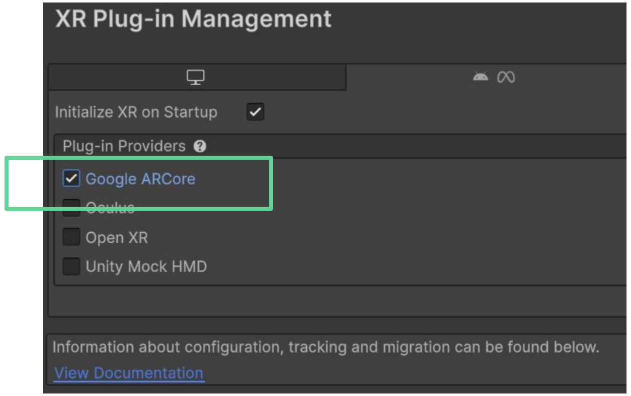
Default Render Pipeline: New Universal Render Pipeline Asset (Universal Render Pipeline Asset)

Graphics Settings

- in Project Settings > Graphics > Reference the Renderer you just Created

XR-Plugin Management Settings

- in Project Settings > XR-Plugin Management > for Android > Google AR Core
- in Google AR Core > make Depth Optional



XR Plug-in Management

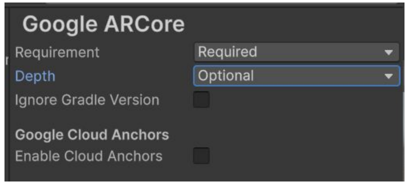
Initialize XR on Startup ✓

Plug-in Providers

- ✓ Google ARCore
- Oculus
- Open XR
- Unity Mock HMD

Information about configuration, tracking and migration can be found below.

View Documentation

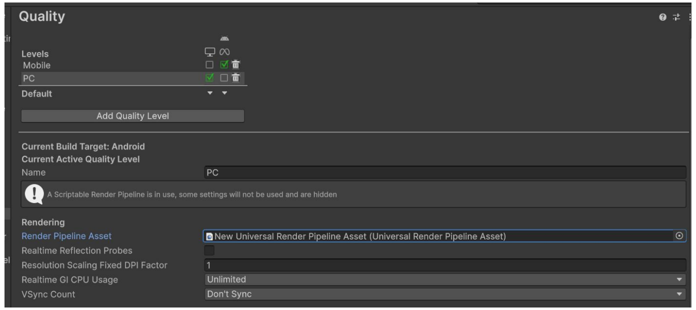


Google ARCore

Requirement	Required
Depth	Optional
Ignore Gradle Version	

Google Cloud Anchors

Enable Cloud Anchors	



Quality

Levels		
Mobile		✓
PC	✓	

Default

Add Quality Level

Current Build Target: Android

Current Active Quality Level

Name: PC

A Scriptable Render Pipeline is in use, some settings will not be used and are hidden

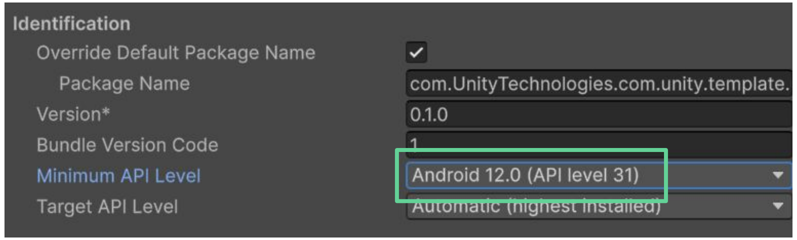
Rendering

Render Pipeline Asset	New Universal Render Pipeline Asset (Universal Render Pipeline Asset)
Realtime Reflection Probes	
Resolution Scaling Fixed DPI Factor	1
Realtime GI CPU Usage	Unlimited
VSync Count	Don't Sync

AR Settings

Player Settings

- in Project Settings > Player > Other Settings > in “Identification” > “Minimum API Level” > Android 12.0



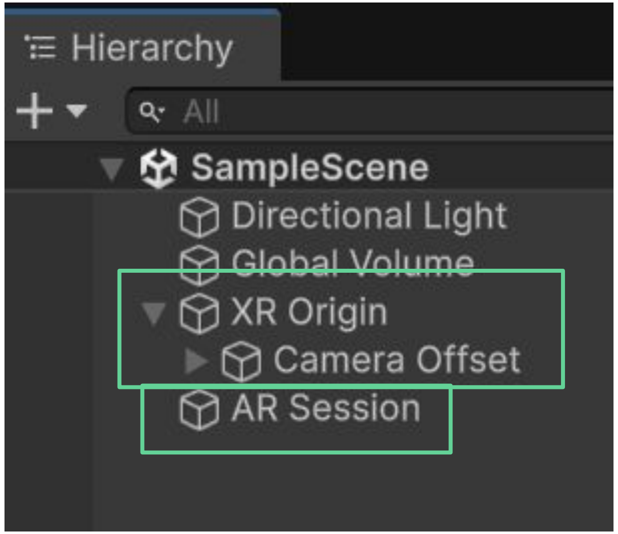
Identification

Override Default Package Name	✓
Package Name	com.UnityTechnologies.com.unity.template.
Version*	0.1.0
Bundle Version Code	1
Minimum API Level	Android 12.0 (API level 31)
Target API Level	Automatic (highest installed)

AR Scene Setup

in Hierarchy:

- delete Main Camera
- create > XR > XR Origin (Mobile AR)
- create > XR > AR Session

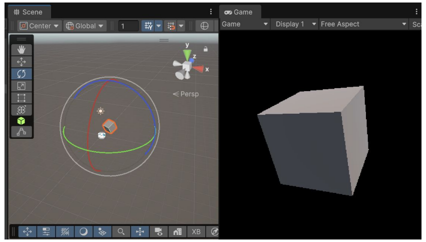


https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.3/manual/project-setup/scene-setup.html

Augment for Test

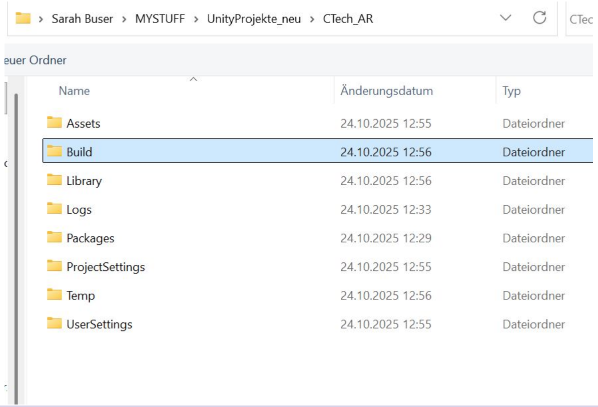
in Hierarchy:

- create > 3D > Cube



build your apk, to see if the pipeline works

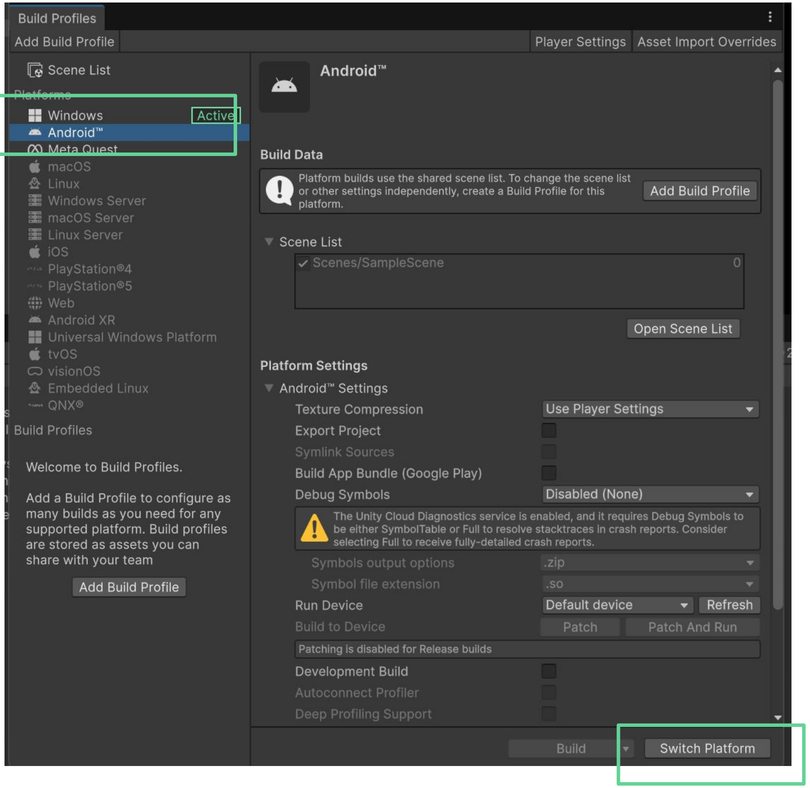
save apk to an extra folder, in the project folder structure, but NOT in the Assets folder:



Sarah Buser > MYSTUFF > UnityProjekte_neu > CTech_AR

euer Ordner

Name	Änderungsdatum	Typ
Assets	24.10.2025 12:55	Dateiordner
Build	24.10.2025 12:56	Dateiordner
Library	24.10.2025 12:56	Dateiordner
Logs	24.10.2025 12:33	Dateiordner
Packages	24.10.2025 12:29	Dateiordner
ProjectSettings	24.10.2025 12:55	Dateiordner
Temp	24.10.2025 12:56	Dateiordner
UserSettings	24.10.2025 12:55	Dateiordner



Build Profiles

Add Build Profile | Player Settings | Asset Import Overrides

Scene List

Platforms
- Windows — Active
- Android™
- Meta Quest
- macOS
- Linux
- Windows Server
- macOS Server
- Linux Server
- iOS
- PlayStation®4
- PlayStation®5
- Web
- Android XR
- Universal Windows Platform
- tvOS
- visionOS
- Embedded Linux
- QNX®

Build Profiles

Welcome to Build Profiles.

Add a Build Profile to configure as many builds as you need for any supported platform. Build profiles are stored as assets you can share with your team

Add Build Profile

Android™

Build Data

Platform builds use the shared scene list. To change the scene list or other settings independently, create a Build Profile for this platform. — Add Build Profile

Scene List
- ✓ Scenes/SampleScene 0

Open Scene List

Platform Settings

Android™ Settings

Texture Compression	Use Player Settings
Export Project	
Symlink Sources	
Build App Bundle (Google Play)	
Debug Symbols	Disabled (None)

The Unity Cloud Diagnostics service is enabled, and it requires Debug Symbols to be either SymbolTable or Full to resolve stacktraces in crash reports. Consider selecting Full to receive fully-detailed crash reports.

Symbols output options	.zip
Symbol file extension	.so
Run Device	Default device — Refresh
Build to Device	Patch — Patch And Run

Patching is disabled for Release builds

Development Build	
Autoconnect Profiler	
Deep Profiling Support	

Build | Switch Platform

Examples Interaction for Augmented Reality

Examples of Interaction in AR

Collision

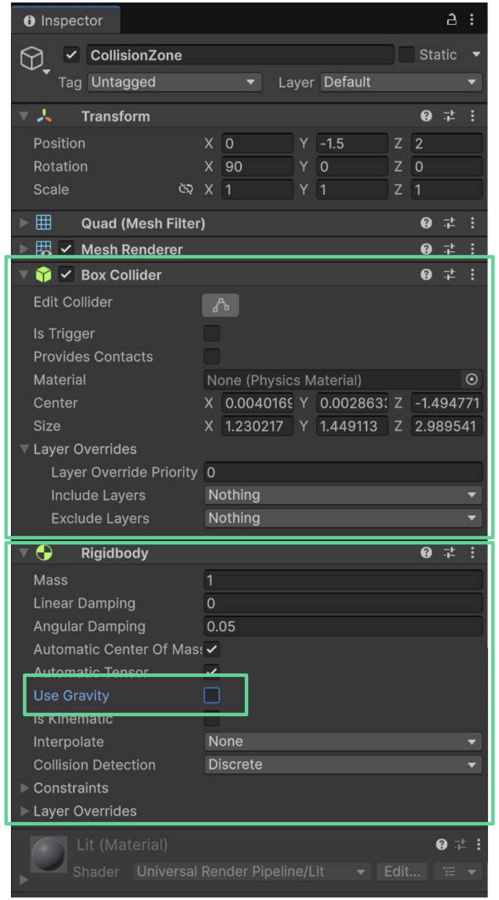
Use case: when you walk into a zone (collide with it) something happens.

- in scene: create a zone with a collider and rigidbody component
- on AR-Camera: add Collider and Rigidbody
- create C# Script, that detects the collision. You can chose where to place it, the logic we code will depend on it. For the example we will be placing it as a component on the Collision Zone

Examples of Interaction in AR

Collision - Create Zone

- create > 3DObject > Quad > remove "Mesh Collider" > add "Box Collider" Component > scale it
- to Quad > add "Rigidbody 3D" Component > uncheck "Use Gravity" > check: "Is Kinematic"



Inspector

✓ CollisionZone   Static
Tag Untagged   Layer Default

Transform
Position X 0 Y -1.5 Z 2
Rotation X 90 Y 0 Z 0
Scale X 1 Y 1 Z 1

Quad (Mesh Filter)

✓ Mesh Renderer

✓ Box Collider
Edit Collider
Is Trigger
Provides Contacts
Material None (Physics Material)
Center X 0.0040169 Y 0.002863 Z -1.494771
Size X 1.230217 Y 1.449113 Z 2.989541
Layer Overrides
Layer Override Priority 0
Include Layers Nothing
Exclude Layers Nothing

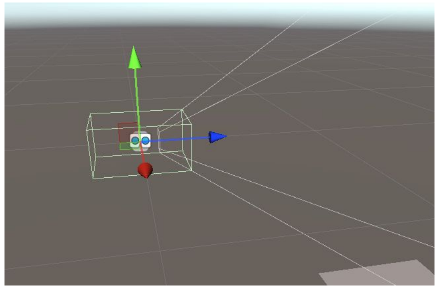
Rigidbody
Mass 1
Linear Damping 0
Angular Damping 0.05
Automatic Center Of Mass ✓
Automatic Tensor ✓
Use Gravity ☐
Is Kinematic
Interpolate None
Collision Detection Discrete
Constraints
Layer Overrides

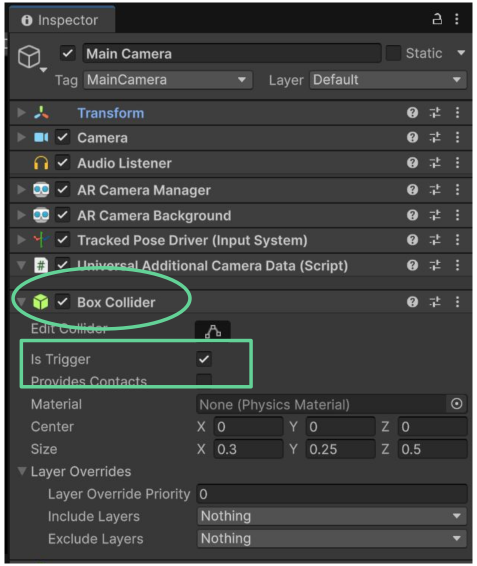
Lit (Material)
Shader Universal Render Pipeline/Lit   Edit...

Examples of Interaction in AR

Collision - Add Physics Componets to AR Camera

- to the MainCamera (Child of XR ORigin > Camera Offset) > add Box Collider > resize it
- check “is Trigger”





Inspector

Main Camera Static

Tag MainCamera Layer Default

Transform
Camera
Audio Listener
AR Camera Manager
AR Camera Background
Tracked Pose Driver (Input System)
Universal Additional Camera Data (Script)
Box Collider

Edit Collider

Is Trigger	✓
Provides Contacts	
Material	None (Physics Material)
Center	X 0 Y 0 Z 0
Size	X 0.3 Y 0.25 Z 0.5

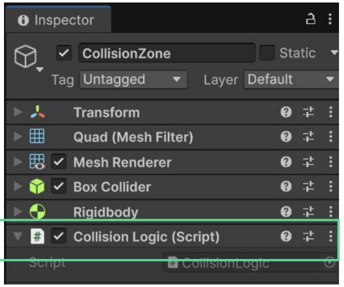
Layer Overrides

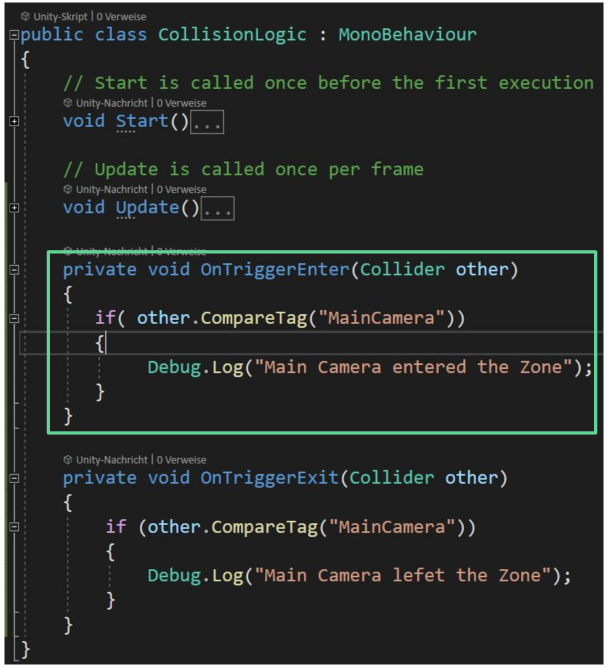
Layer Override Priority	0
Include Layers	Nothing
Exclude Layers	Nothing

Examples of Interaction in AR

Collision - Create Script to add logic

- create > NewMonoBehaviour (in Assetsfolder > Create new Folder “Scripts”) > name it CollisionLogic
- add it as Component to Collision Zone
- OnTrigger enter, check, what Object the Trigger belongs to, if it belongs to the Main Camera, do something.
- Test in Editor, read Console




```
Unity-Skript | 0 Verweise
public class CollisionLogic : MonoBehaviour
{
    // Start is called once before the first execution
    Unity-Nachricht | 0 Verweise
    void Start()...

    // Update is called once per frame
    Unity-Nachricht | 0 Verweise
    void Update()...

    Unity-Nachricht | 0 Verweise
    private void OnTriggerEnter(Collider other)
    {
        if( other.CompareTag("MainCamera"))
        {
            Debug.Log("Main Camera entered the Zone");
        }
    }

    Unity-Nachricht | 0 Verweise
    private void OnTriggerExit(Collider other)
    {
        if (other.CompareTag("MainCamera"))
        {
            Debug.Log("Main Camera lefet the Zone");
        }
    }
}
```

Examples of Interaction in AR

Collision - Exercise

- let something appear/disappear
- play/pause Audio
- play particle System

Examples of Interaction in AR

Touch

Use case: when you tap on an object, something happens

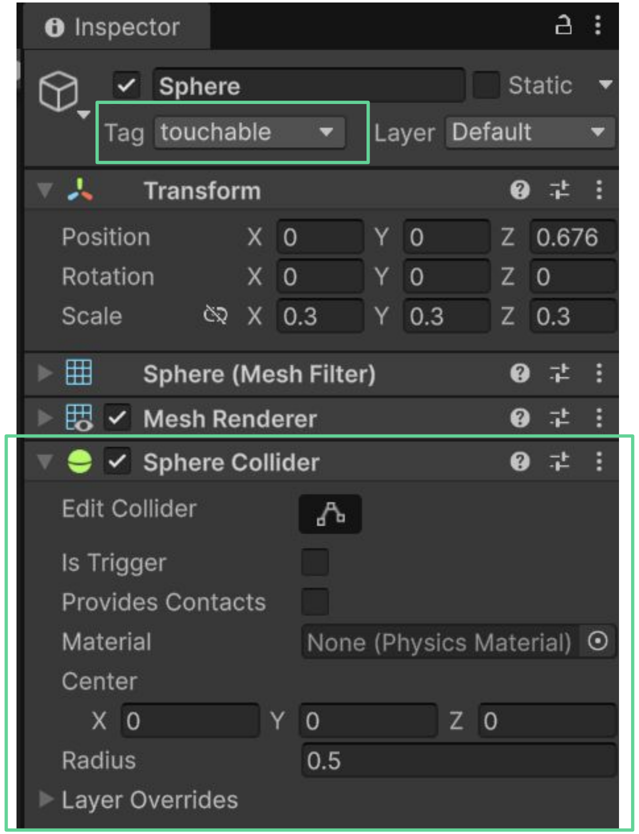
- in scene: create an object you want to touch
- in Scene: create a TouchManager (Empty Game Object) that contains the TouchLogic (MonoBehaviour Script) as a component

Examples of Interaction in AR

Touch - Create touchable Object

in Scene:

- create > 3D Object > Sphere
- on Sphere > Add Tag “touchable”



Inspector

Sphere — Static

Tag: touchable — Layer: Default

Transform

	X	Y	Z
Position	0	0	0.676
Rotation	0	0	0
Scale	0.3	0.3	0.3

Sphere (Mesh Filter)

Mesh Renderer

Sphere Collider

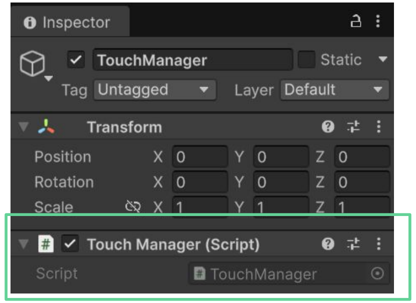
Edit Collider	
Is Trigger	☐
Provides Contacts	☐
Material	None (Physics Material)
Center	X 0 Y 0 Z 0
Radius	0.5

Layer Overrides

Examples of Interaction in AR

Touch - Create TouchManager

- in Scene: create > Empty GameObject > name it “TouchManager”
- in Scripts: Create > new MonoBehaviour Script > name it “TouchManager” (can be anything)
- add Script to Empty Object

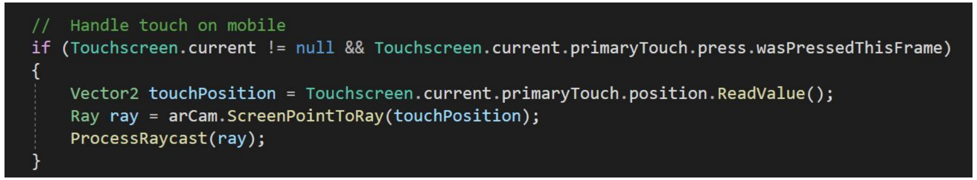


```
TouchManager                     Static
Tag Untagged          Layer Default
Position   X 0   Y 0   Z 0
Rotation   X 0   Y 0   Z 0
Scale      X 1   Y 1   Z 1
Script     TouchManager
```

new Input System

Examples of Interaction in AR

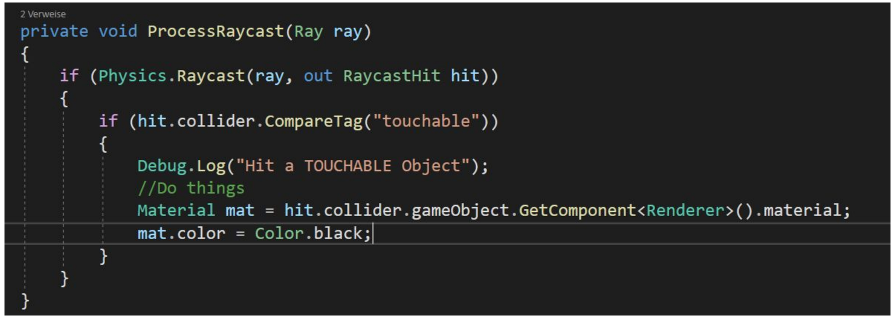
Touch - Write Logic



```
//  Handle touch on mobile
if (Touchscreen.current != null && Touchscreen.current.primaryTouch.press.wasPressedThisFrame)
{
    Vector2 touchPosition = Touchscreen.current.primaryTouch.position.ReadValue();
    Ray ray = arCam.ScreenPointToRay(touchPosition);
    ProcessRaycast(ray);
}
```

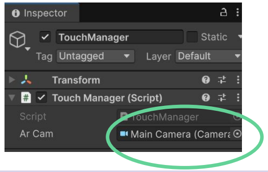
we are not using the ARRaycastManager provided by ARFoundation, but instead work with the Unity integrated Physics Raycast:

- in TouchManager Script > Reference the MainCamera
- in Update CheckForTouch
- if we have Touch, perform a Raycast
- check if the Raycast touches something
- check if the something is a “touchable” object



```
2 Verweise
private void ProcessRaycast(Ray ray)
{
    if (Physics.Raycast(ray, out RaycastHit hit))
    {
        if (hit.collider.CompareTag("touchable"))
        {
            Debug.Log("Hit a TOUCHABLE Object");
            //Do things
            Material mat = hit.collider.gameObject.GetComponent<Renderer>().material;
            mat.color = Color.black;
        }
    }
}
```

in Editor: don’t forget to reference the ARCamera!



```
Inspector
TouchManager          Static
Tag Untagged    Layer Default
Transform
Touch Manager (Script)
Script    TouchManager
Ar Cam    Main Camera (Camera)
```

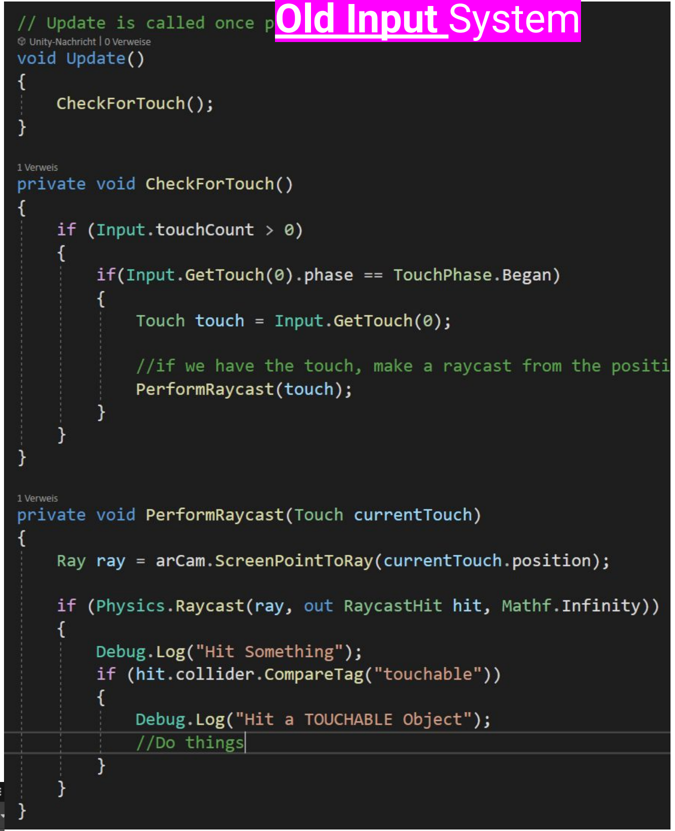
Examples of Interaction in AR

Touch - Write Logic

we are not using the ARRaycastManager provided by ARFoundation, but instead work with the Unity integrated Physics Raycast:

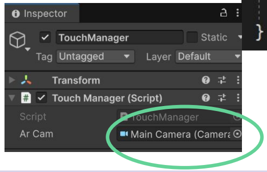
- in TouchManager Script > Reference the MainCamera
- in Update CheckForTouch
- if we have Touch, perform a Raycast
- check if the Raycast touches something
- check if the something is a “touchable” object

this is the code for the **Old Input** System



```
// Update is called once p
Unity-Nachricht | 0 Verweise
void Update()
{
    CheckForTouch();
}

1 Verweis
private void CheckForTouch()
{
    if (Input.touchCount > 0)
    {
        if(Input.GetTouch(0).phase == TouchPhase.Began)
        {
            Touch touch = Input.GetTouch(0);

            //if we have the touch, make a raycast from the positi
            PerformRaycast(touch);
        }
    }
}

1 Verweis
private void PerformRaycast(Touch currentTouch)
{
    Ray ray = arCam.ScreenPointToRay(currentTouch.position);

    if (Physics.Raycast(ray, out RaycastHit hit, Mathf.Infinity))
    {
        Debug.Log("Hit Something");
        if (hit.collider.CompareTag("touchable"))
        {
            Debug.Log("Hit a TOUCHABLE Object");
            //Do things
        }
    }
}
```

in Editor: don’t forget to reference the ARCamera!

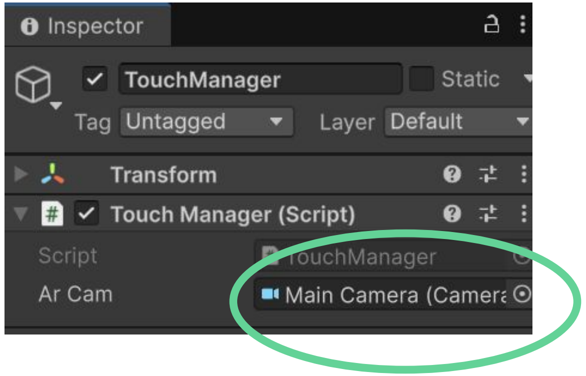


Script: TouchManager
Ar Cam: Main Camera (Camera

Examples of Interaction in AR

Touch - Final Adjustments

- in Editor: reference the ARCamera in the TouchManager script



Inspector

TouchManager | Static

Tag Untagged | Layer Default

Transform

Touch Manager (Script)

Script: TouchManager

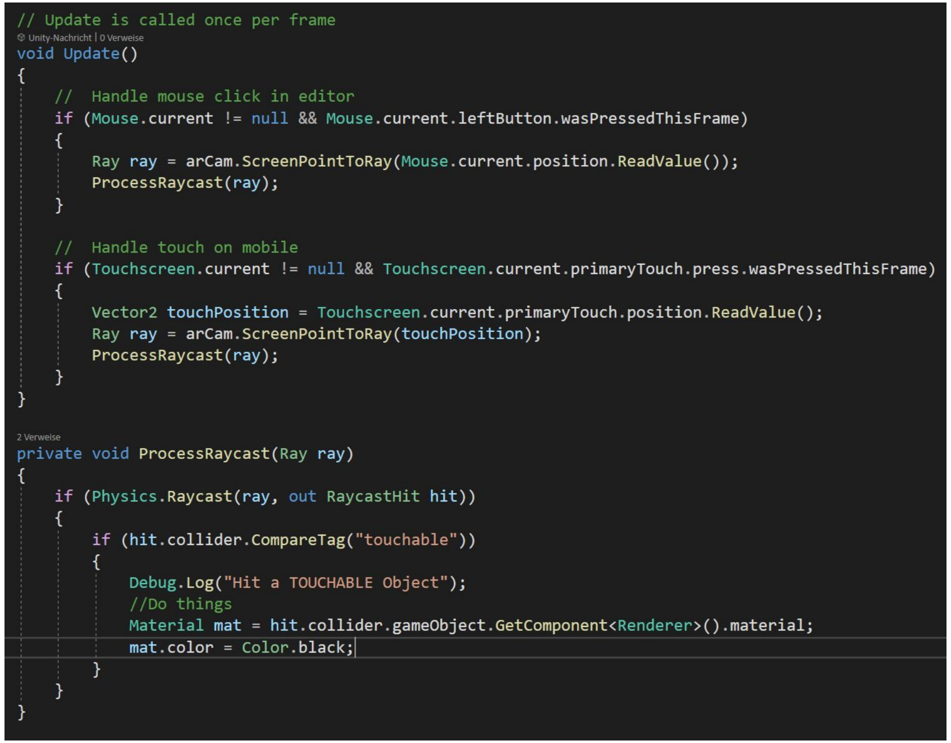
Ar Cam: Main Camera (Camera

Examples of Interaction in AR

Touch - Exercise

atm there is nothing detectable happening, add some lines of code to get a feedback, when the “touchable” Object is touched:

- play sound
- change color
- make the object disappear
- play particle system
- your idea



```
// Update is called once per frame
Unity-Nachricht | 0 Verweise
void Update()
{
    //  Handle mouse click in editor
    if (Mouse.current != null && Mouse.current.leftButton.wasPressedThisFrame)
    {
        Ray ray = arCam.ScreenPointToRay(Mouse.current.position.ReadValue());
        ProcessRaycast(ray);
    }

    //  Handle touch on mobile
    if (Touchscreen.current != null && Touchscreen.current.primaryTouch.press.wasPressedThisFrame)
    {
        Vector2 touchPosition = Touchscreen.current.primaryTouch.position.ReadValue();
        Ray ray = arCam.ScreenPointToRay(touchPosition);
        ProcessRaycast(ray);
    }
}

2 Verweise
private void ProcessRaycast(Ray ray)
{
    if (Physics.Raycast(ray, out RaycastHit hit))
    {
        if (hit.collider.CompareTag("touchable"))
        {
            Debug.Log("Hit a TOUCHABLE Object");
            //Do things
            Material mat = hit.collider.gameObject.GetComponent<Renderer>().material;
            mat.color = Color.black;
        }
    }
}
```

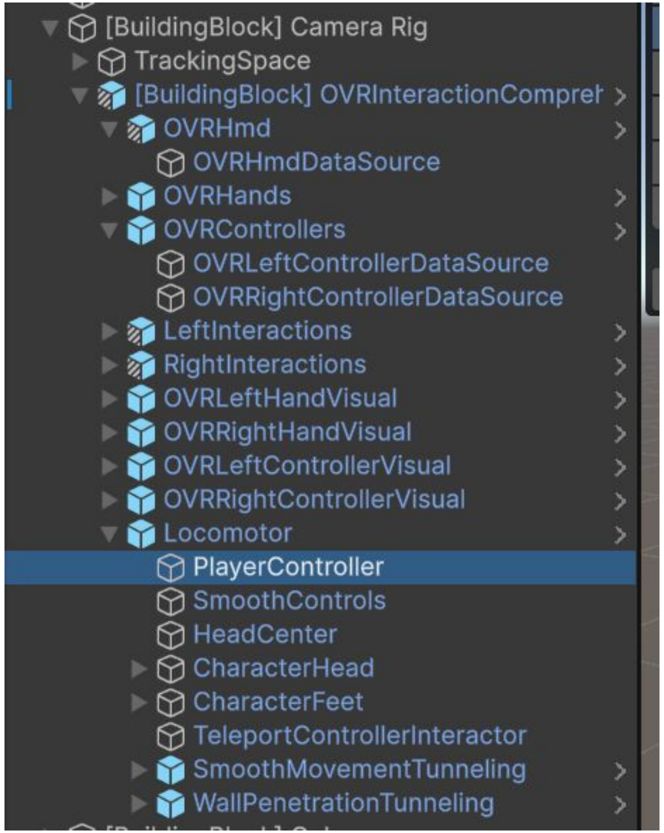
Examples Interaction for Virtual Reality

Examples of Interaction in VR

Collision - Create Trigger

Use case: when you walk into a zone (collide with it) something happens.

- on Camera-Rig > Center Eye Anchor: add Collider and Rigidbody
- OR: On CameraRig > OVRINteractionComprehensive > Locomotor > Player Controller > Check “is Trigger”



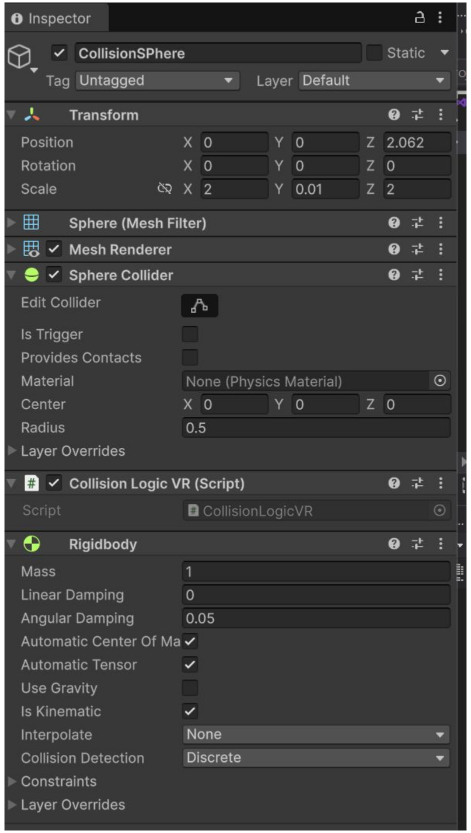
```
▼ [BuildingBlock] Camera Rig
    ▶ TrackingSpace
    ▼ [BuildingBlock] OVRInteractionCompreh
        ▼ OVRHmd
            OVRHmdDataSource
        ▶ OVRHands
        ▼ OVRControllers
            OVRLeftControllerDataSource
            OVRRightControllerDataSource
        ▶ LeftInteractions
        ▶ RightInteractions
        ▶ OVRLeftHandVisual
        ▶ OVRRightHandVisual
        ▶ OVRLeftControllerVisual
        ▶ OVRRightControllerVisual
        ▼ Locomotor
            PlayerController
            SmoothControls
            HeadCenter
          ▶ CharacterHead
          ▶ CharacterFeet
            TeleportControllerInteractor
          ▶ SmoothMovementTunneling
          ▶ WallPenetrationTunneling
```

Examples of Interaction in VR

Collision - Create Zone

Use case: when you walk into a zone (collide with it) something happens.

- in scene: create a zone with a collider and rigidbody component > uncheck “use Gravity” > Check “Is Kinematic”



Inspector

✓ CollisionSPhere Static
Tag Untagged Layer Default

Transform

Position	X	0	Y	0	Z	2.062
Rotation	X	0	Y	0	Z	0
Scale	X	2	Y	0.01	Z	2

Sphere (Mesh Filter)

✓ **Mesh Renderer**

✓ **Sphere Collider**

Edit Collider	
Is Trigger	☐
Provides Contacts	☐
Material	None (Physics Material)
Center	X 0 Y 0 Z 0
Radius	0.5
Layer Overrides	

✓ **Collision Logic VR (Script)**

Script CollisionLogicVR

Rigidbody

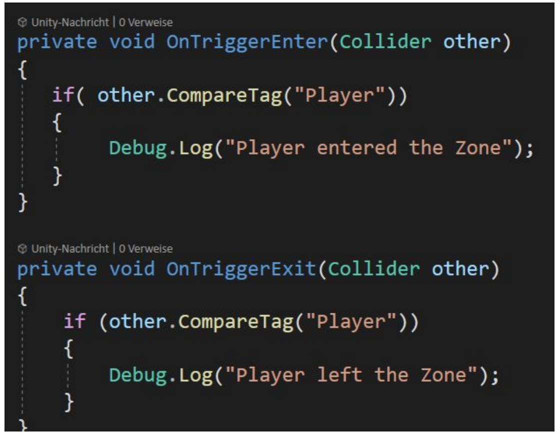
Mass	1
Linear Damping	0
Angular Damping	0.05
Automatic Center Of Ma	✓
Automatic Tensor	✓
Use Gravity	☐
Is Kinematic	✓
Interpolate	None
Collision Detection	Discrete
Constraints	
Layer Overrides	

Examples of Interaction in VR

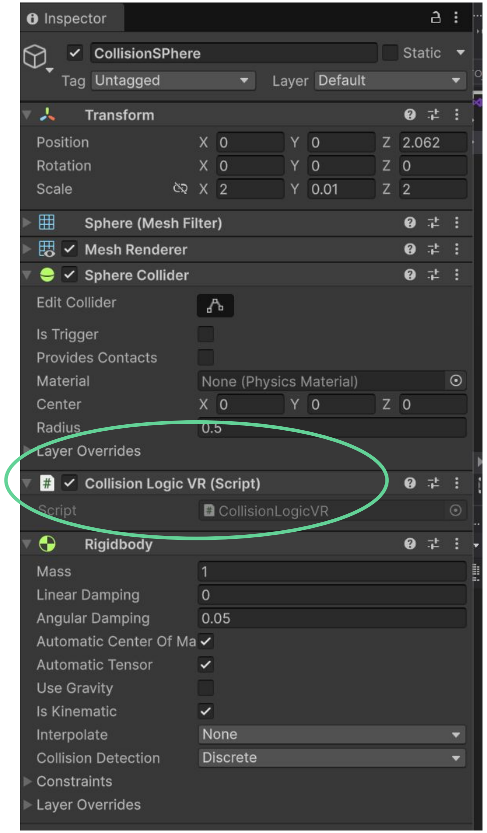
Collision - Logic

Use case: when you walk into a zone (collide with it) something happens.

- create C# Script, that detects the collision. You can chose where to place it, the logic we code will depend on it. For the example we will be placing it as a component on the Collision Zone



```
Unity-Nachricht | 0 Verweise
private void OnTriggerEnter(Collider other)
{
   if( other.CompareTag("Player"))
   {
       Debug.Log("Player entered the Zone");
   }
}

Unity-Nachricht | 0 Verweise
private void OnTriggerExit(Collider other)
{
    if (other.CompareTag("Player"))
    {
        Debug.Log("Player left the Zone");
    }
```



```
Inspector
✓ CollisionSPhere                      Static
Tag Untagged          Layer Default
Transform
Position        X 0    Y 0     Z 2.062
Rotation        X 0    Y 0     Z 0
Scale           X 2    Y 0.01  Z 2
Sphere (Mesh Filter)
✓ Mesh Renderer
✓ Sphere Collider
Edit Collider
Is Trigger
Provides Contacts
Material        None (Physics Material)
Center          X 0    Y 0     Z 0
Radius          0.5
Layer Overrides
✓ Collision Logic VR (Script)
Script          CollisionLogicVR
Rigidbody
Mass                     1
Linear Damping           0
Angular Damping          0.05
Automatic Center Of Ma   ✓
Automatic Tensor         ✓
Use Gravity
Is Kinematic             ✓
Interpolate              None
Collision Detection      Discrete
Constraints
Layer Overrides
```

Examples of Interaction in VR

Touch

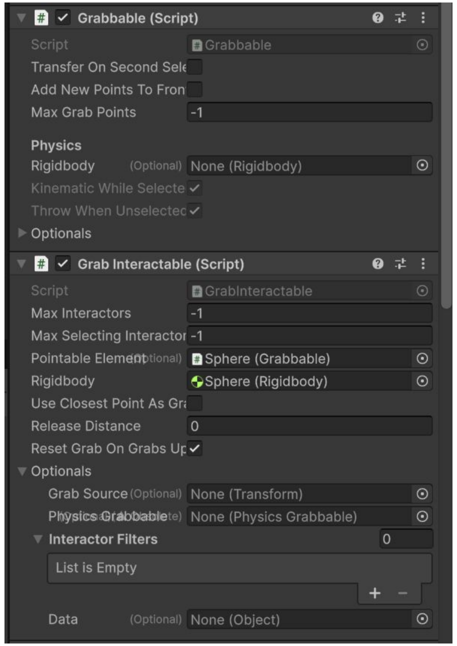
Use case: when you tap on an object, something happens

- in scene: create an object you want to touch
- in Scene: create a ConsequenceManager (Empty Game Object) that contains the Consequences (MonoBehaviour Script) as a component

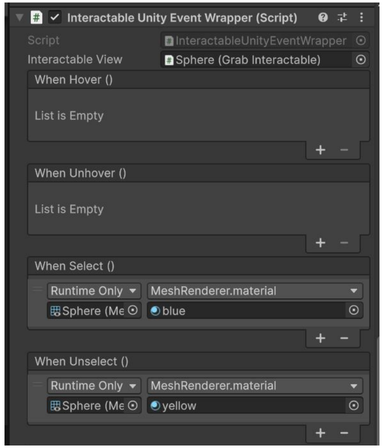
Examples of Interaction in VR

Touch - Create Grabbable Objects

- in Editor: Create > 3D Object > Sphere
- on Sphere > add Rigidbody Component
- on Sphere > add Grabbable.cs
- on Sphere > add GrabInteractable.cs
- on Sphere > add InteractableUnityEventWrapper.cs
- on InteractableUnityEventWrapper > in event System > define Consequences of touching the object


Grabbable (Script)
Script: Grabbable
Transfer On Second Sele
Add New Points To Fron
Max Grab Points: -1
Physics
Rigidbody (Optional): None (Rigidbody)
Kinematic While Selecte: ✓
Throw When Unselectec: ✓
Optionals

Grab Interactable (Script)
Script: GrabInteractable
Max Interactors: -1
Max Selecting Interactor: -1
Pointable Element (Optional): Sphere (Grabbable)
Rigidbody: Sphere (Rigidbody)
Use Closest Point As Gr
Release Distance: 0
Reset Grab On Grabs Up: ✓
Optionals
Grab Source (Optional): None (Transform)
Physics Grabbable (Optional): None (Physics Grabbable)
Interactor Filters: 0
List is Empty
Data (Optional): None (Object)


Interactable Unity Event Wrapper (Script)
Script: InteractableUnityEventWrapper
Interactable View: Sphere (Grab Interactable)
When Hover ()
List is Empty
When Unhover ()
List is Empty
When Select ()
Runtime Only | MeshRenderer.material
Sphere (Me | blue
When Unselect ()
Runtime Only | MeshRenderer.material
Sphere (Me | yellow