[Unity] Set Up MRTK2 For VIVE WAVE

Oct, 2023

Revision History

Revision	Date	Description
0.1	Aug 2023	Alpha release
0.2	Sept 2023	Add Eye tracking
0.3	Oct 2023	Add passthrough, change model and troubleshooting

Contents

1. Introduction

VIVE Wave Unity plugin supports the **MRTK Support** feature. By following this guide you can build an application with MRTK and run on VIVE mobile devices.

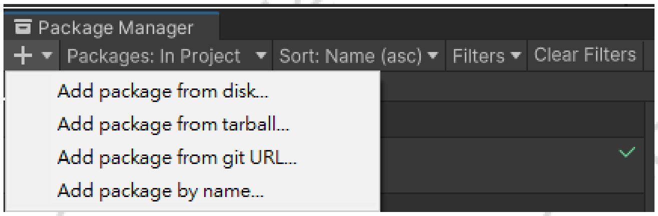
The feature of **MRTK Support** :

- Controller
- Hand Tracking
- Eye Tracking

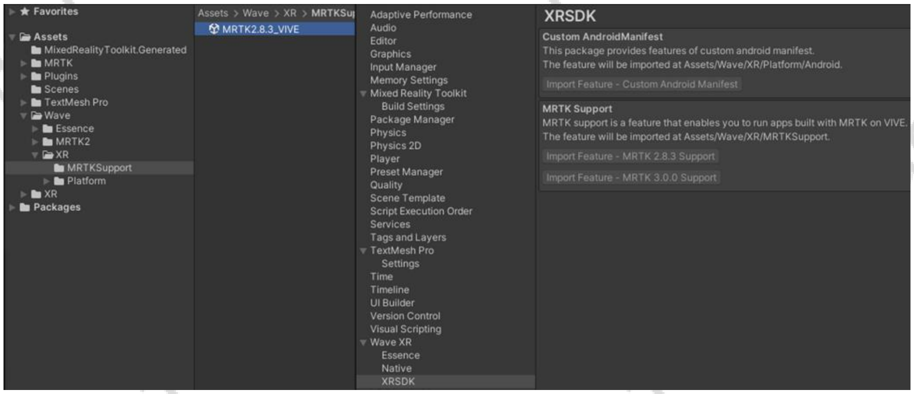
2. Development Environment

2.1. Download Packages

1. Download version 2.8.3 of MRTK Foundation, MRTK Tools and MRTK Examples (optional) from MRTK GitHub.
2. Import these packages into your project and skip MRTK configurator setup.
3. Import WAVE SDK into your project from menu **Window > Package manager > Add package from tarball**.



4. Import MRTK Support feature from menu **Edit > Project Settings > Wave XR > XRSDK**.
5. Import MRTK2.8.3_VIVE package from Assets/Wave/XR/MRTKSupport/.



XRSDK

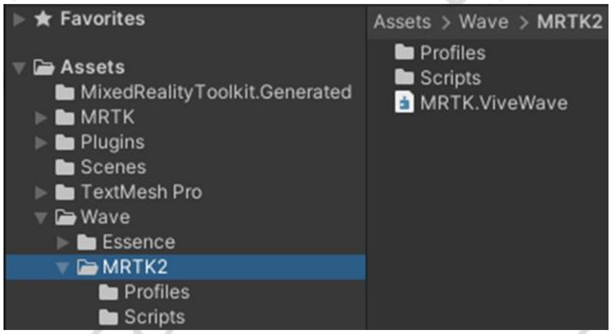
Custom AndroidManifest

This package provides features of custom android manifest.
The feature will be imported at Assets/Wave/XR/Platform/Android.

MRTK Support

MRTK support is a feature that enables you to run apps built with MRTK on VIVE.
The feature will be imported at Assets/Wave/XR/MRTKSupport.

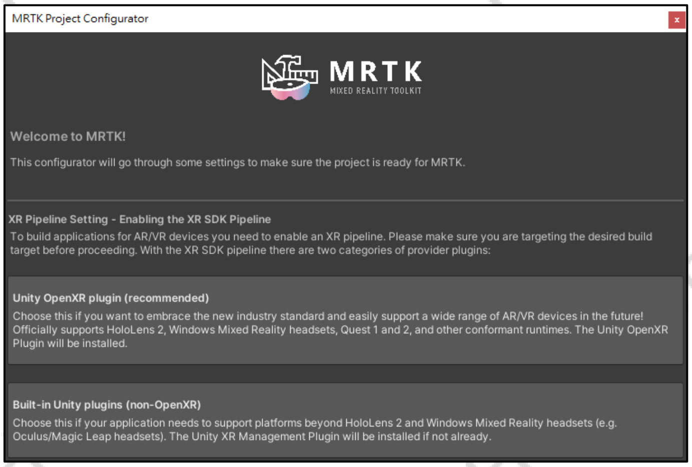
6. Check MRTK2 folder be added to Assets/ Wave/.



2.2. Project Settings

1. Switch to Android platform.
2. Setup MRTK Configurator from menu **Mixed Reality > Toolkit > Utilities > Configure Project for MRTK**.

2.1. Select non-OpenXR if you wouldn't use OpenXR in your project.



MRTK Project Configurator

Welcome to MRTK!

This configurator will go through some settings to make sure the project is ready for MRTK.

XR Pipeline Setting - Enabling the XR SDK Pipeline

To build applications for AR/VR devices you need to enable an XR pipeline. Please make sure you are targeting the desired build target before proceeding. With the XR SDK pipeline there are two categories of provider plugins:

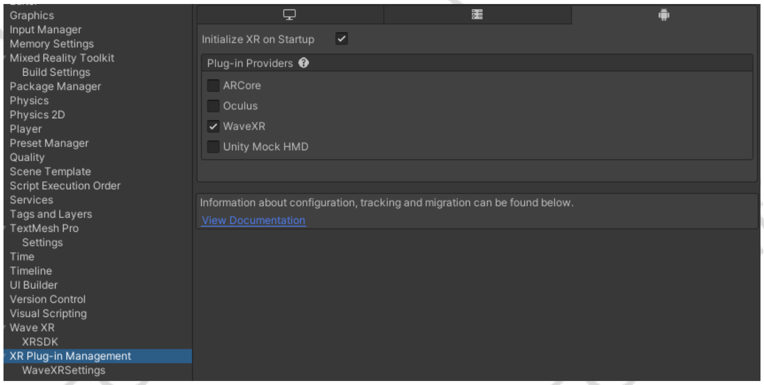
Unity OpenXR plugin (recommended)

Choose this if you want to embrace the new industry standard and easily support a wide range of AR/VR devices in the future! Officially supports HoloLens 2, Windows Mixed Reality headsets, Quest 1 and 2, and other conformant runtimes. The Unity OpenXR Plugin will be installed.

Built-in Unity plugins (non-OpenXR)

Choose this if your application needs to support platforms beyond HoloLens 2 and Windows Mixed Reality headsets (e.g. Oculus/Magic Leap headsets). The Unity XR Management Plugin will be installed if not already.

2.2. Show Settings and enable Wave XR in XR Plug-in Management.



Initialize XR on Startup ☑

Plug-in Providers

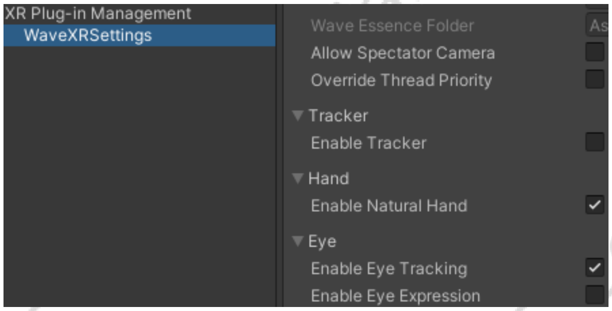
☐ ARCore
☐ Oculus
☑ WaveXR
☐ Unity Mock HMD

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

View Documentation

2.3. Apply and import the recommended settings.

3. Enable Natural Hand and Eye Tracking(optional) from menu **Edit > Project Settings > XR Plug-in Management > WaveXRSettings**.



Wave Essence Folder

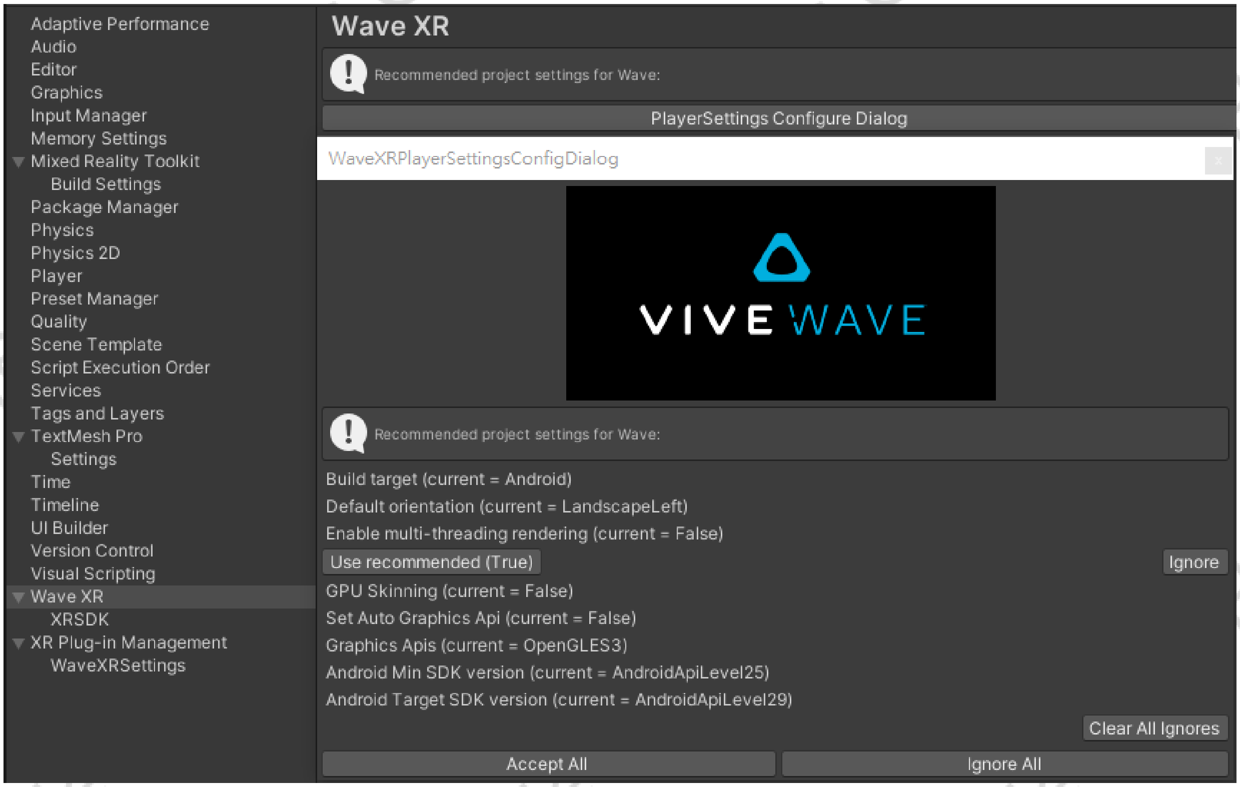
Allow Spectator Camera ☐
Override Thread Priority ☐

Tracker
Enable Tracker ☐

Hand
Enable Natural Hand ☑

Eye
Enable Eye Tracking ☑
Enable Eye Expression ☐

4. Check WaveXRPlayerSettingsConfigDialog from menu **Edit > Project Settings > Wave XR**.
5. Disable multi-thread rendering and click ignore to avoid warning.



Wave XR

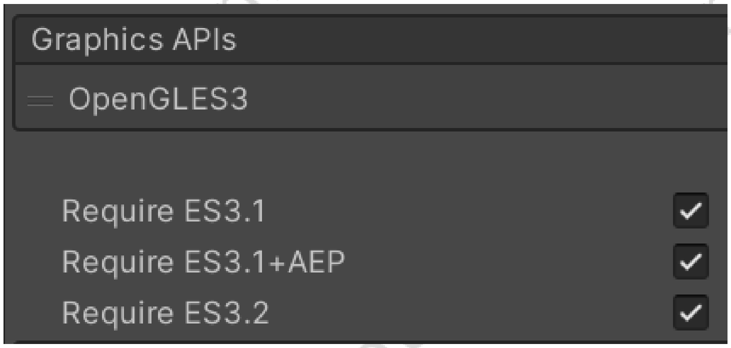
Recommended project settings for Wave:

WaveXRPlayerSettingsConfigDialog

Recommended project settings for Wave:

Build target (current = Android)
Default orientation (current = LandscapeLeft)
Enable multi-threading rendering (current = False)
GPU Skinning (current = False)
Set Auto Graphics Api (current = False)
Graphics Apis (current = OpenGLES3)
Android Min SDK version (current = AndroidApiLevel25)
Android Target SDK version (current = AndroidApiLevel29)

6. Check Graphics APIs from menu **Edit > Project Settings > Player > Other Settings** only select OpenGLES3.



Graphics APIs

OpenGLES3

Require ES3.1 ☑
Require ES3.1+AEP ☑
Require ES3.2 ☑

2.3. Fixed MRTK Shader

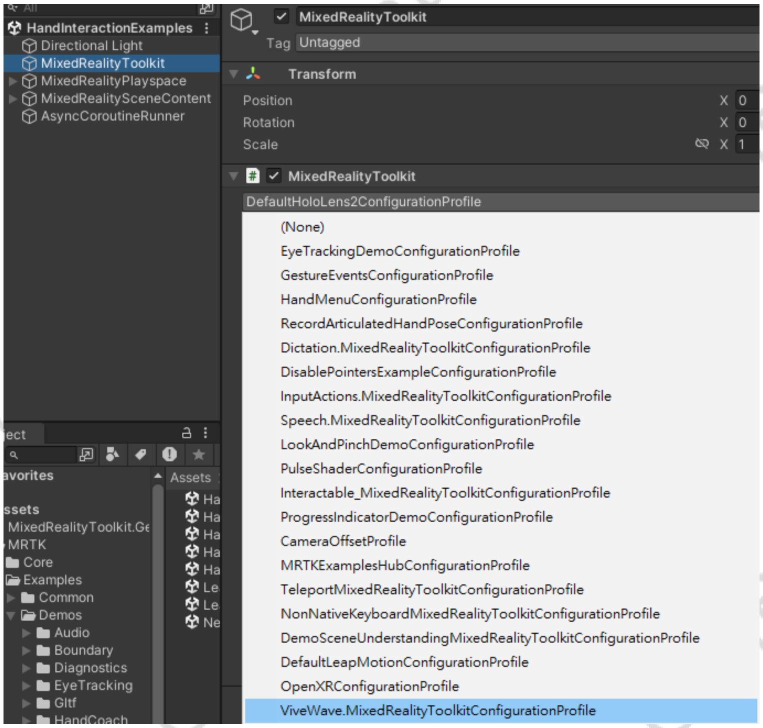
The MixedRealityStandard.shader from **Assets/MRTK/StandardAssets/Shaders/** may give an error in Unity 2022 LTS when building. Make these edits in the shader to fix this.

1. On line 775, fixed4 frag(v2f i, fixed facing : VFACE) : SV_Target change to fixed4 frag(v2f i) : SV_Target.
2. On lines 956 and 959, remove * facing.

2.4. Hand Interaction Example

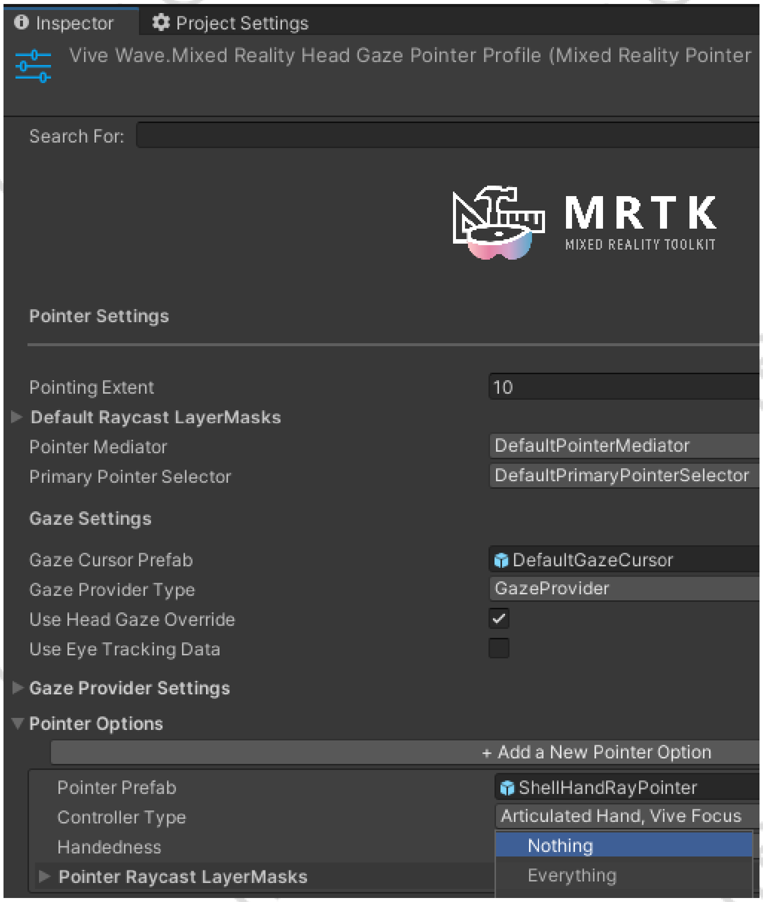
You can build an example which imported by MRTK Examples to verify your environment.

1. Open demo scene HandInteractionExamples.unity from Assets/MRTK/Examples/Demos/HandTracking /Scenes/.
2. Change the MixedRealityToolkit Profile to ViveWave.MixedRealityToolkitConfigurationProfile.


HandInteractionExamples
Directional Light
MixedRealityToolkit
MixedRealityPlayspace
MixedRealitySceneContent
AsyncCoroutineRunner
MixedRealityToolkit
Tag Untagged
Transform
Position
Rotation
Scale
X 0
X 0
X 1
MixedRealityToolkit
DefaultHoloLens2ConfigurationProfile
(None)
EyeTrackingDemoConfigurationProfile
GestureEventsConfigurationProfile
HandMenuConfigurationProfile
RecordArticulatedHandPoseConfigurationProfile
Dictation.MixedRealityToolkitConfigurationProfile
DisablePointersExampleConfigurationProfile
InputActions.MixedRealityToolkitConfigurationProfile
Speech.MixedRealityToolkitConfigurationProfile
LookAndPinchDemoConfigurationProfile
PulseShaderConfigurationProfile
Interactable_MixedRealityToolkitConfigurationProfile
ProgressIndicatorDemoConfigurationProfile
CameraOffsetProfile
MRTKExamplesHubConfigurationProfile
TeleportMixedRealityToolkitConfigurationProfile
NonNativeKeyboardMixedRealityToolkitConfigurationProfile
DemoSceneUnderstandingMixedRealityToolkitConfigurationProfile
DefaultLeapMotionConfigurationProfile
OpenXRConfigurationProfile
ViveWave.MixedRealityToolkitConfigurationProfile
ect
avorites
Assets
ssets
MixedRealityToolkit.Ge
MRTK
Core
Examples
Common
Demos
Audio
Boundary
Diagnostics
EyeTracking
Gltf
HandCoach
Ha
Ha
Ha
Ha
Ha
Le
Le
Ne

3. Build and Run from menu **File > Build Settings**.
4. In the example, you can test the functionality of the controller, hand tracking, and eye tracking and also interact with UGUI and objects using hand ray or controller ray.
   Controller : Trigger Press(Select), Grab Press(Grab)
   Hand Tracking : Pinch(Select), Grab(Grab)
   Note: Please refer to ViveWave.MixedRealityControllerMappingProfile from Assets/MRTK/Providers/Vive/Profiles/ for more input actions.

5. If you want to replace hand ray or controller ray to head gaze or eye gaze(it need to switch eye tracking profile), set controller type to nothing in ShellHandRayPointer of pointer options.



Vive Wave.Mixed Reality Head Gaze Pointer Profile (Mixed Reality Pointer

Search For:

Pointer Settings

Pointing Extent: 10

Default Raycast LayerMasks

Pointer Mediator: DefaultPointerMediator

Primary Pointer Selector: DefaultPrimaryPointerSelector

Gaze Settings

Gaze Cursor Prefab: DefaultGazeCursor

Gaze Provider Type: GazeProvider

Use Head Gaze Override: ☑

Use Eye Tracking Data: ☐

Gaze Provider Settings

Pointer Options

Pointer Prefab: ShellHandRayPointer

Controller Type: Articulated Hand, Vive Focus

Handedness

Pointer Raycast LayerMasks

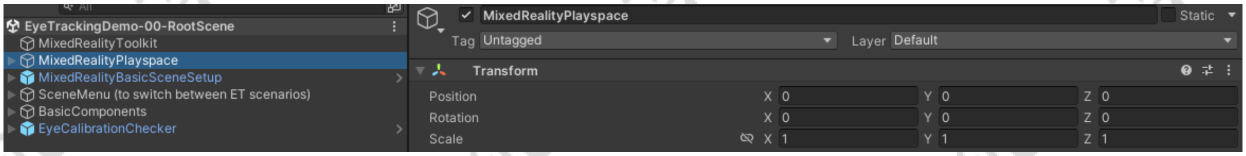
Nothing

Everything

2.5. Eye Tracking Example

You can build an example which imported by MRTK Examples to verify your environment.

1. Open demo scene EyeTrackingDemo-00-RootScene.unity from Assets/MRTK/Examples/Demos/EyeTracking/ Scenes/.
2. Fix bug with offset by setting position of MixedRealityPlayspace to (0, 0, 0).

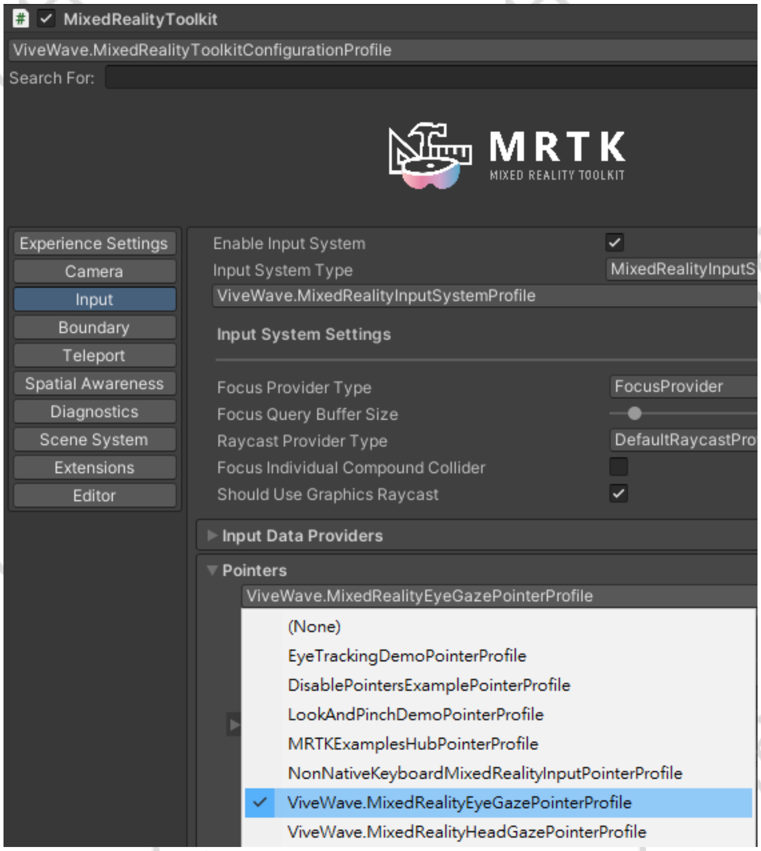


EyeTrackingDemo-00-RootScene
MixedRealityToolkit
MixedRealityPlayspace
MixedRealityBasicSceneSetup
SceneMenu (to switch between ET scenarios)
BasicComponents
EyeCalibrationChecker

☑ MixedRealityPlayspace □ Static
Tag Untagged Layer Default
Transform

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

3. Change the MixedRealityToolkit Profile to ViveWave.MixedRealityToolkitConfigurationProfile. (Follow 2.4. Hand Interaction Example step 2)
4. Change the profile from **MixedRealityToolkit > Input > Pointers** to ViveWave.MixedRealityEyeGazePointerProfile



☑ MixedRealityToolkit
ViveWave.MixedRealityToolkitConfigurationProfile
Search For:

Enable Input System ☑
Input System Type MixedRealityInputS
ViveWave.MixedRealityInputSystemProfile

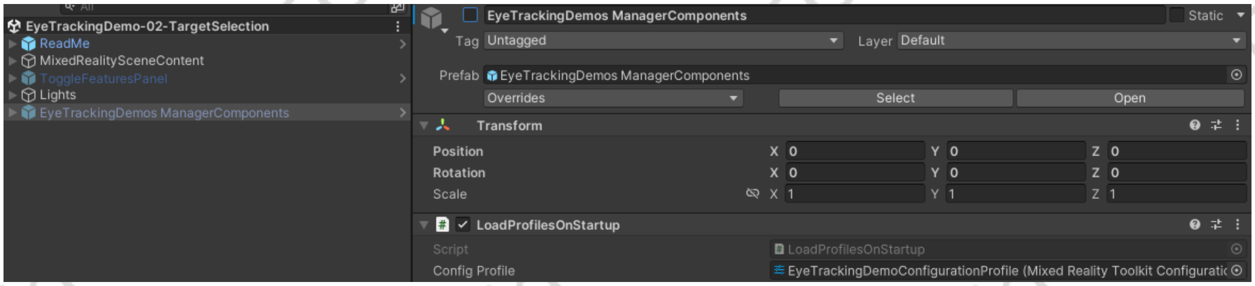
Input System Settings

Focus Provider Type FocusProvider
Focus Query Buffer Size
Raycast Provider Type DefaultRaycastPro
Focus Individual Compound Collider □
Should Use Graphics Raycast ☑

Input Data Providers

Pointers
ViveWave.MixedRealityEyeGazePointerProfile
(None)
EyeTrackingDemoPointerProfile
DisablePointersExamplePointerProfile
LookAndPinchDemoPointerProfile
MRTKExamplesHubPointerProfile
NonNativeKeyboardMixedRealityInputPointerProfile
✓ ViveWave.MixedRealityEyeGazePointerProfile
ViveWave.MixedRealityHeadGazePointerProfile

5. Open EyeTrackingDemo-(02~05) demo scene from Assets/MRTK/Examples/Demos/EyeTracking/ Scenes/
6. Disable EyeTrackingDemos ManagerComponents to avoid loading other profiles.



EyeTrackingDemo-02-TargetSelection
- ReadMe
- MixedRealitySceneContent
- ToggleFeaturesPanel
- Lights
- EyeTrackingDemos ManagerComponents

□ EyeTrackingDemos ManagerComponents — Static

Tag: Untagged — Layer: Default

Prefab: EyeTrackingDemos ManagerComponents

Overrides — Select — Open

Transform

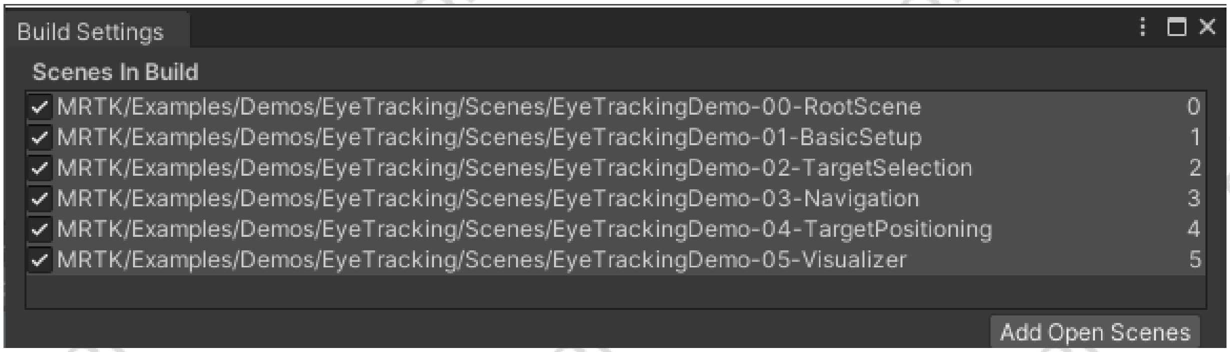
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ LoadProfilesOnStartup

Script: LoadProfilesOnStartup

Config Profile: EyeTrackingDemoConfigurationProfile (Mixed Reality Toolkit Configuratic

7. Add all demo scenes in Assets/MRTK/Examples/Demos/EyeTracking/Scenes/ to Scenes In Build from menu **File > Build Settings**.



Build Settings

Scenes In Build

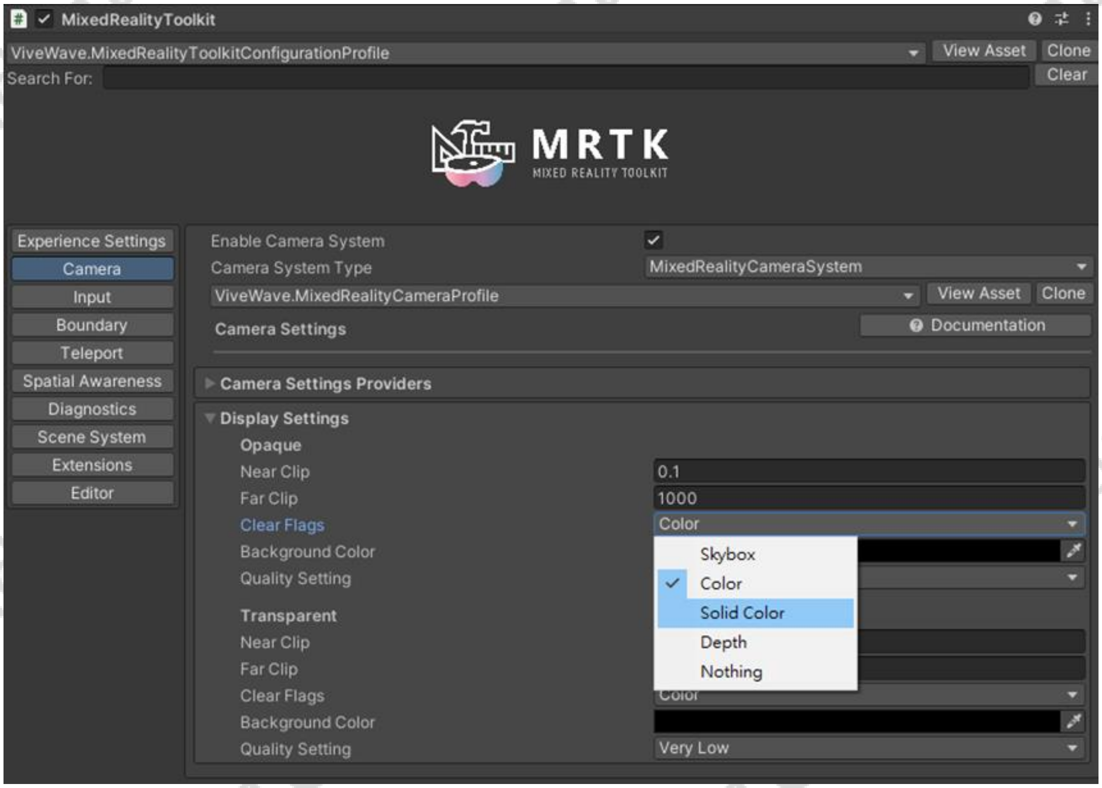
☑	MRTK/Examples/Demos/EyeTracking/Scenes/EyeTrackingDemo-00-RootScene	0
☑	MRTK/Examples/Demos/EyeTracking/Scenes/EyeTrackingDemo-01-BasicSetup	1
☑	MRTK/Examples/Demos/EyeTracking/Scenes/EyeTrackingDemo-02-TargetSelection	2
☑	MRTK/Examples/Demos/EyeTracking/Scenes/EyeTrackingDemo-03-Navigation	3
☑	MRTK/Examples/Demos/EyeTracking/Scenes/EyeTrackingDemo-04-TargetPositioning	4
☑	MRTK/Examples/Demos/EyeTracking/Scenes/EyeTrackingDemo-05-Visualizer	5

Add Open Scenes

8. In the example, you can use eye gaze and pinch gesture to switch scene and implement different interactive features.

3. Enable Passthrough

1. Import WAVE Native into your project from menu **Window > Package manager > Add package from tarball**.
2. Add script to call passthrough APIs, here listed the passthrough APIs:
   public static WVR_Result WVR_ShowPassthroughUnderlay(bool show)
3. The clear flags of Opaque and Transparent in your camera profile should be set to Color or Solid Color and the background color should be set to (0, 0, 0, 0).



MixedRealityToolkit

ViveWave.MixedRealityToolkitConfigurationProfile

Search For:

Enable Camera System ✓

Camera System Type: MixedRealityCameraSystem

ViveWave.MixedRealityCameraProfile

Camera Settings

Camera Settings Providers

Display Settings

Opaque

Near Clip: 0.1

Far Clip: 1000

Clear Flags: Color

Background Color

Quality Setting

Transparent

Near Clip

Far Clip

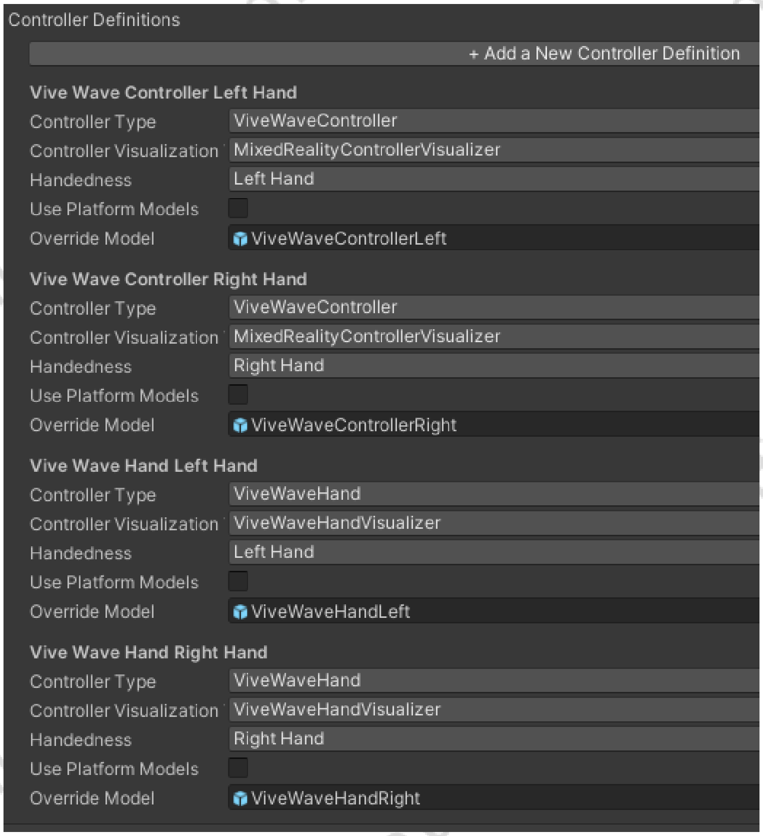
Clear Flags: Color

Background Color

Quality Setting: Very Low

4. Change Model

1. Import WAVE Essence into your project from menu **Window > Package manager > Add package from tarball**.
2. Import Controller Model and Hand Model feature from menu **Edit > Project Settings > Wave XR > Essence** and ensure these prefabs are valid. (Assets/Wave/Essence/Hand/Model/version/Resources/Prefabs/MRTK/) (Assets/Wave/Essence/Controller/Model/version/Resources/Prefabs/MRTK/)
3. Ensure ViveWave.MixedRealityControllerVisualizationProfile from Assets/MRTK/Providers/Vive/Profiles/ is valid if not you should manually add new controller definition.



Controller Definitions

+ Add a New Controller Definition

Vive Wave Controller Left Hand

Controller Type	ViveWaveController
Controller Visualization	MixedRealityControllerVisualizer
Handedness	Left Hand
Use Platform Models	☐
Override Model	ViveWaveControllerLeft

Vive Wave Controller Right Hand

Controller Type	ViveWaveController
Controller Visualization	MixedRealityControllerVisualizer
Handedness	Right Hand
Use Platform Models	☐
Override Model	ViveWaveControllerRight

Vive Wave Hand Left Hand

Controller Type	ViveWaveHand
Controller Visualization	ViveWaveHandVisualizer
Handedness	Left Hand
Use Platform Models	☐
Override Model	ViveWaveHandLeft

Vive Wave Hand Right Hand

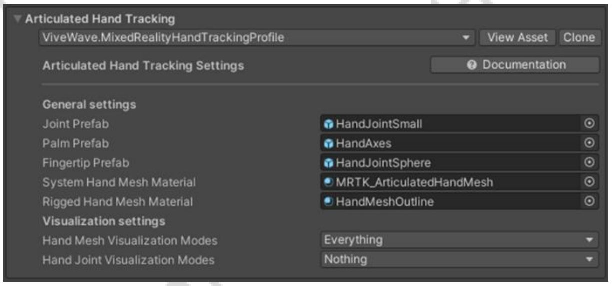
Controller Type	ViveWaveHand
Controller Visualization	ViveWaveHandVisualizer
Handedness	Right Hand
Use Platform Models	☐
Override Model	ViveWaveHandRight

4. Change the Controller Visualization Settings from **Input System Profile > Controller** to ViveWave.MixedRealityControllerVisualizationProfile.

5. Troubleshooting

1. Hand tracking model not show.

⇨ Ensure that your Articulated Hand Tracking from **MixedRealityToolkit > Input** has added profile.



▼ Articulated Hand Tracking

ViveWave.MixedRealityHandTrackingProfile ▾ | View Asset | Clone

Articulated Hand Tracking Settings | Documentation

General settings

Joint Prefab	HandJointSmall
Palm Prefab	HandAxes
Fingertip Prefab	HandJointSphere
System Hand Mesh Material	MRTK_ArticulatedHandMesh
Rigged Hand Mesh Material	HandMeshOutline

Visualization settings

Hand Mesh Visualization Modes	Everything
Hand Joint Visualization Modes	Nothing

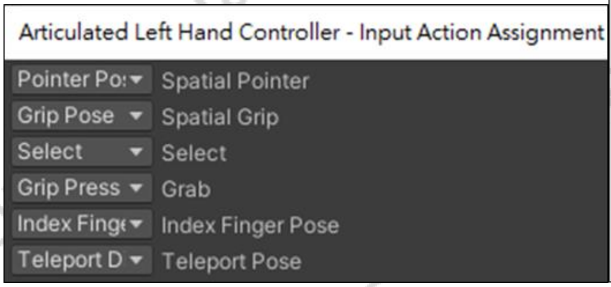
2. The direction of default controller model is warong.

⇨ Set Rotation Offset of ControllerLeft and ControllerRight from Assets/MRTK/SDK/StandardAssets/Controllers/ to (0, 180, 0).

⇨ Set the rotation of gameobjects named left/right under ControllerLeft/ ControllerRight to (0, 180, 0).

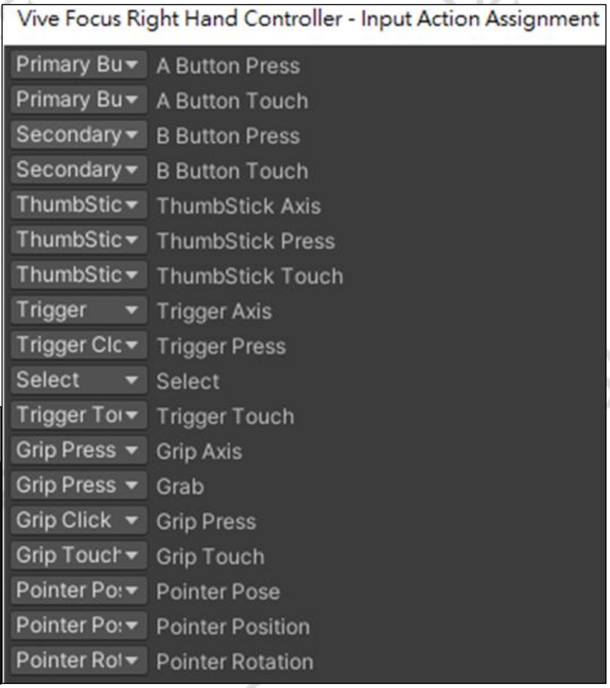
3. Input actions not working with hand or controller.

⇨ Ensure that controller input mapping settings from MixedRealityToolkit > Input > Controllers of articulated hand and vive focus are valid.



Articulated Left Hand Controller - Input Action Assignment

Pointer Po: ▾	Spatial Pointer
Grip Pose ▾	Spatial Grip
Select ▾	Select
Grip Press ▾	Grab
Index Finge ▾	Index Finger Pose
Teleport D ▾	Teleport Pose



Vive Focus Right Hand Controller - Input Action Assignment

Primary Bu ▾	A Button Press
Primary Bu ▾	A Button Touch
Secondary ▾	B Button Press
Secondary ▾	B Button Touch
ThumbStic ▾	ThumbStick Axis
ThumbStic ▾	ThumbStick Press
ThumbStic ▾	ThumbStick Touch
Trigger ▾	Trigger Axis
Trigger Clc ▾	Trigger Press
Select ▾	Select
Trigger To ▾	Trigger Touch
Grip Press ▾	Grip Axis
Grip Press ▾	Grab
Grip Click ▾	Grip Press
Grip Touch ▾	Grip Touch
Pointer Po: ▾	Pointer Pose
Pointer Po: ▾	Pointer Position
Pointer Ro ▾	Pointer Rotation

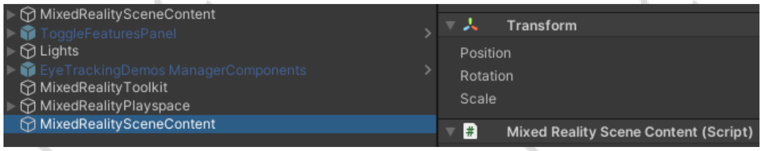
6. Appendix

6.1. Single Scene With Eye Tracking

If you want to run one demo scene in MRTK Example, follow the steps below.

1. Open a demo scene (01~05) from Assets/MRTK/Examples/Demos/EyeTracking/Scenes.

2. Add MixedRealityToolkit, MixedRealityPlayspace and MixedRealitySceneContent from menu **Mixed Reality > Toolkit > Add to Scene and Configure** and delete MixedRealitySceneContent which dosen't contain any gameobjects.



3. Follow 2.5. Eye Tracking Example step 3~6.
4. Build and Run from menu **File > Build Settings**.