About this Library

Hand Gesture Tool for Apple Vision Pro is copyright (c) 2024 Graffity inc.

This library is enable easy to use handgesture on Apple Vision Pro.

Summary

- Gesture can be configured with ScriptableObject.
- The ScriptableObject can be used to retrieve various events.
- This package provides the above features.
- This is implemented using Unity XRHands.

Contents

System Requirements

- Unity 2022.3 or later (strongly recommended Unity 2022.3.20f1 or higher)
  - com.unity.xr.hands 1.3.0
  - com.unity.xr.visionos 1.2.3
  - com.unity.textmeshpro 3.0.8
- Xcode 15.3
- R3 1.1.11

License

- R3

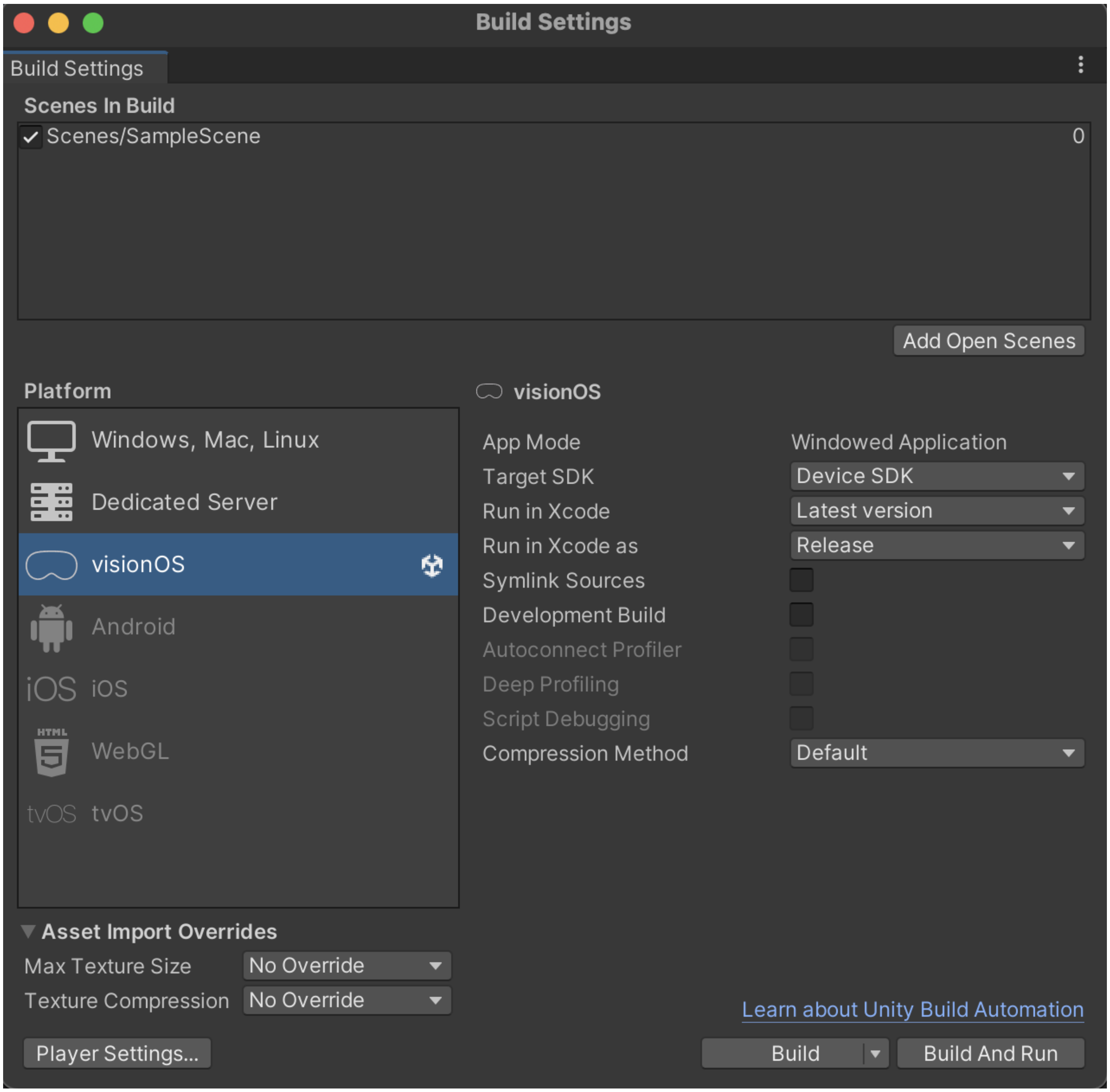
Environment Setup

R3

Install R3 in Unity. Installation method using NugetForUnity is strongly recommended.

Polyspatial

1. Open BuildSettings and change Platform to visionOS.


Build Settings

Scenes In Build

✓ Scenes/SampleScene 0

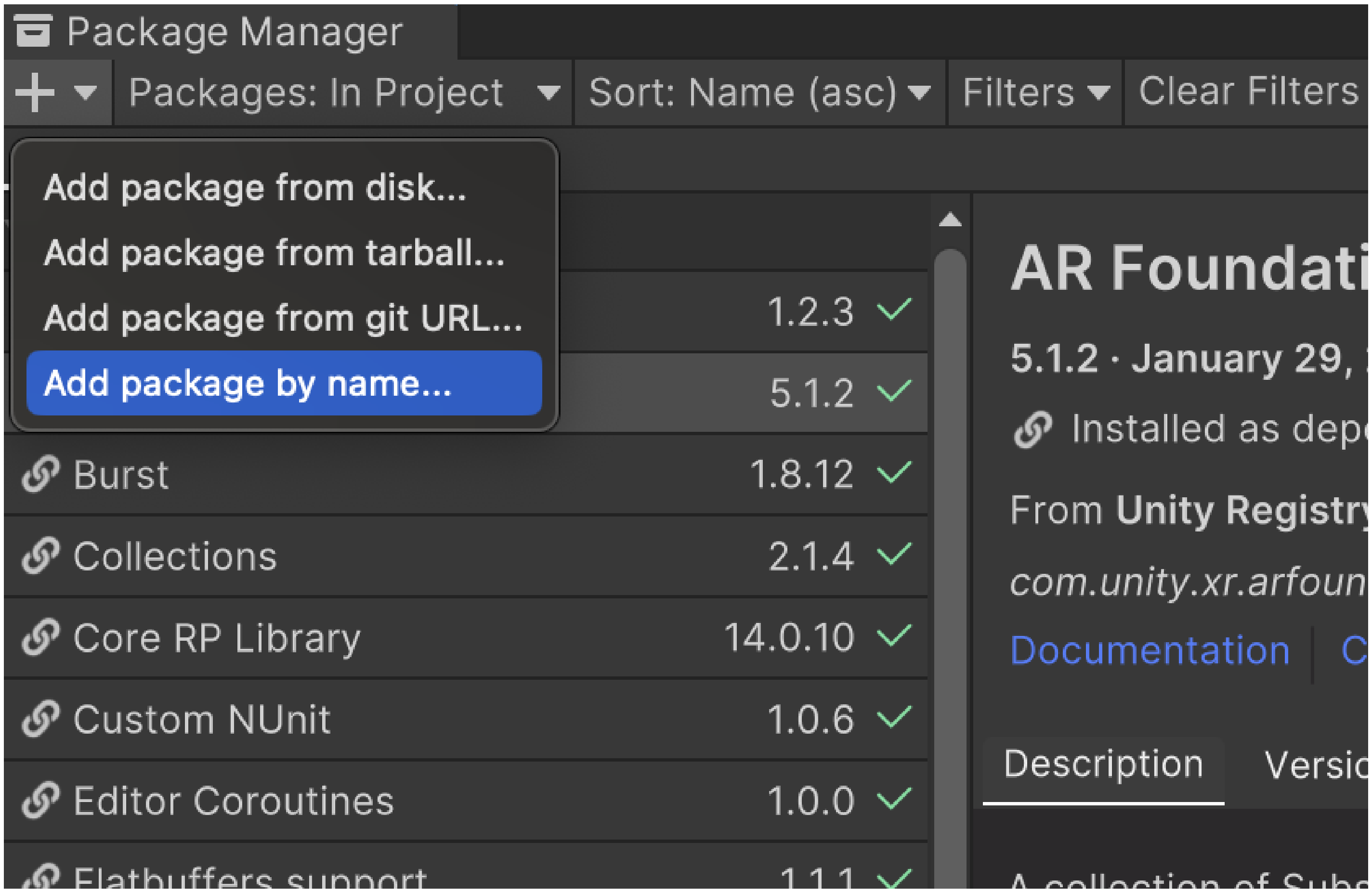
Platform: visionOS

App Mode	Windowed Application
Target SDK	Device SDK
Run in Xcode	Latest version
Run in Xcode as	Release
Symlink Sources	☐
Development Build	☐
Autoconnect Profiler	☐
Deep Profiling	☐
Script Debugging	☐
Compression Method	Default

Asset Import Overrides

Max Texture Size	No Override
Texture Compression	No Override

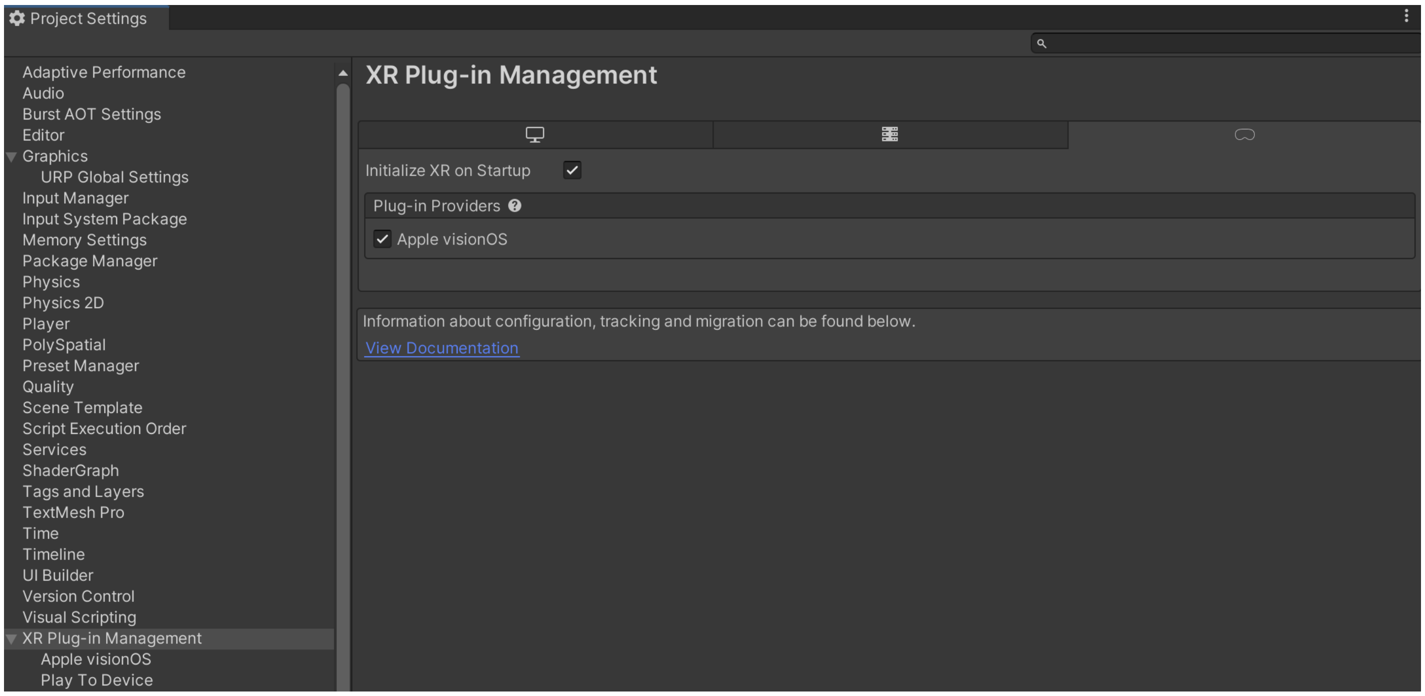
2. Add package by name..." from PackageManager from the PackageManager. Install the following packages.
   - com.unity.polyspatial.xr
   - com.unity.polyspatial.visionos



	1.2.3 ✓
	5.1.2 ✓
Burst	1.8.12 ✓
Collections	2.1.4 ✓
Core RP Library	14.0.10 ✓
Custom NUnit	1.0.6 ✓
Editor Coroutines	1.0.0 ✓
Flatbuffers support	1.1.1 ✓

AR Foundati
5.1.2 · January 29,
Installed as depe
From **Unity Registry**
com.unity.xr.arfoun
Documentation | C
Description Versio

3. Check Apple visionOS in Project Settings → XR Plug-in Management.



XR Plug-in Management

Initialize XR on Startup ✓

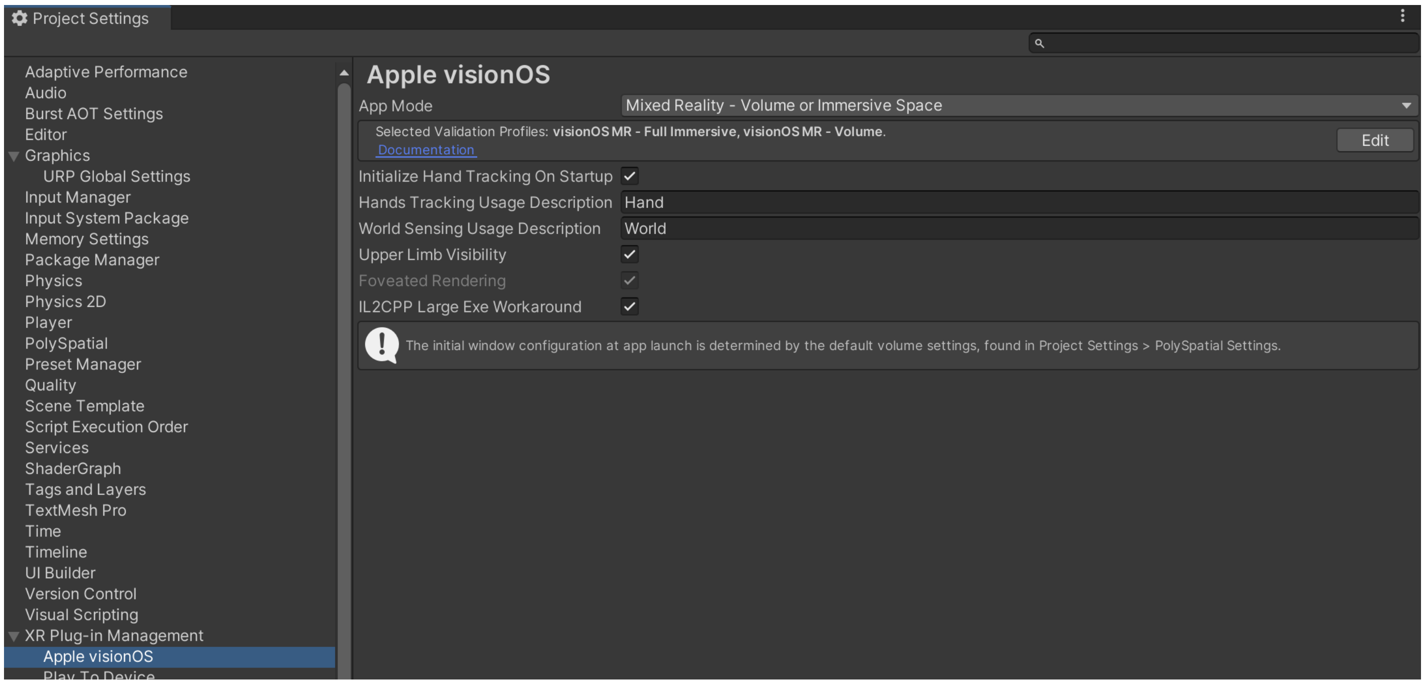
Plug-in Providers

✓ Apple visionOS

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

View Documentation

4. Project Settings → XR Plug-in Management → Apple visionOS settings.



Apple visionOS

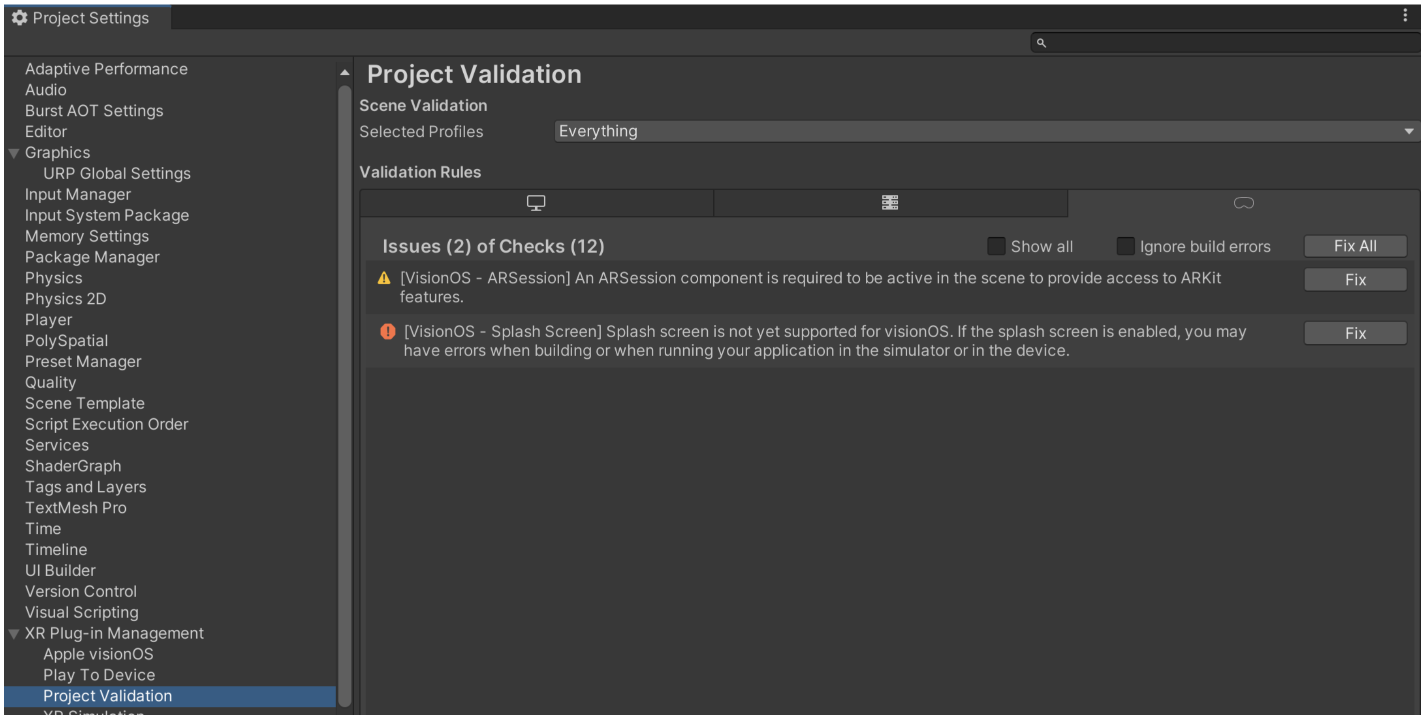
App Mode: Mixed Reality - Volume or Immersive Space

Selected Validation Profiles: **visionOS MR - Full Immersive, visionOS MR - Volume**.

Initialize Hand Tracking On Startup ✓
Hands Tracking Usage Description: Hand
World Sensing Usage Description: World
Upper Limb Visibility ✓
Foveated Rendering ✓
IL2CPP Large Exe Workaround ✓

The initial window configuration at app launch is determined by the default volume settings, found in Project Settings > PolySpatial Settings.

5. Project Settings → XR Plug-in Management → Project Validation and press the Fix button to fix the problem.



Project Validation

Scene Validation

Selected Profiles: Everything

Validation Rules

Issues (2) of Checks (12)

[VisionOS - ARSession] An ARSession component is required to be active in the scene to provide access to ARKit features.

[VisionOS - Splash Screen] Splash screen is not yet supported for visionOS. If the splash screen is enabled, you may have errors when building or when running your application in the simulator or in the device.

Flow

1. How to create a GestureAsset
    1. Create GestureAsset
    2. GestureAsset Settings
    3. Setup GestureAsset in GestureManager
2. Get Event with Script
    1. Result

2. GestureState

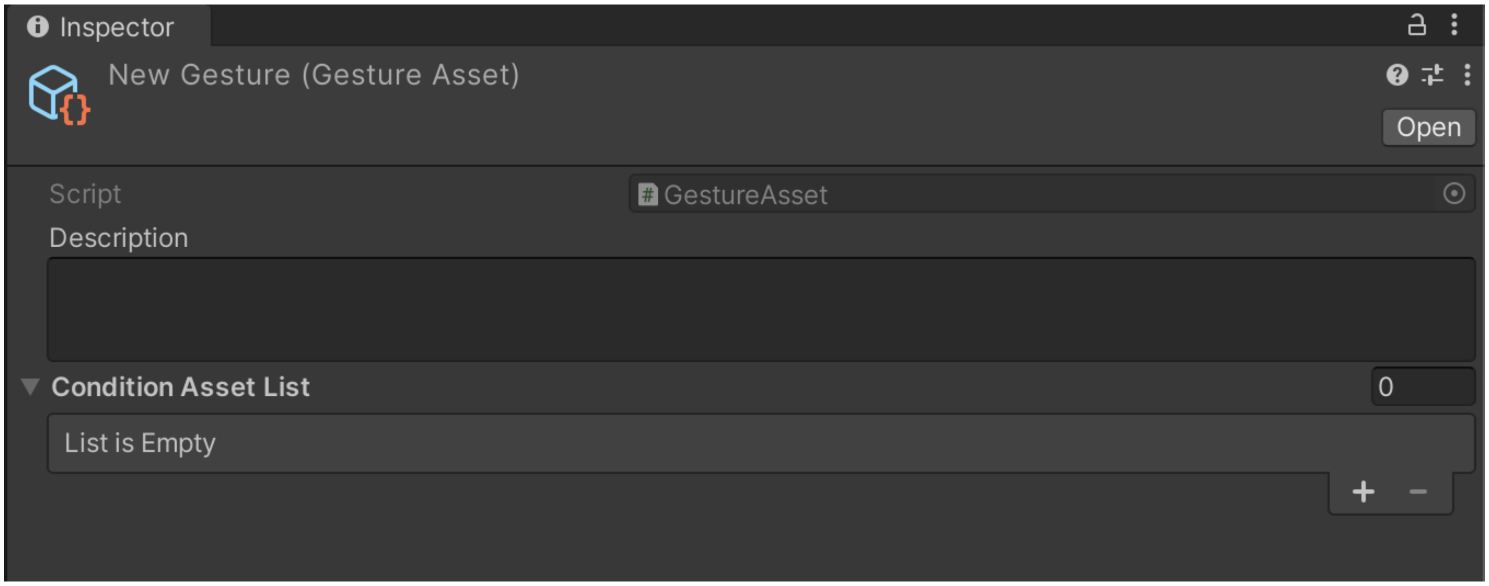
1. How to create a GestureAsset

Create GestureAsset

A new GestureAsset can be created by following the steps below.

> Assets → Create → Graffity → HandGesture → GestureAsset

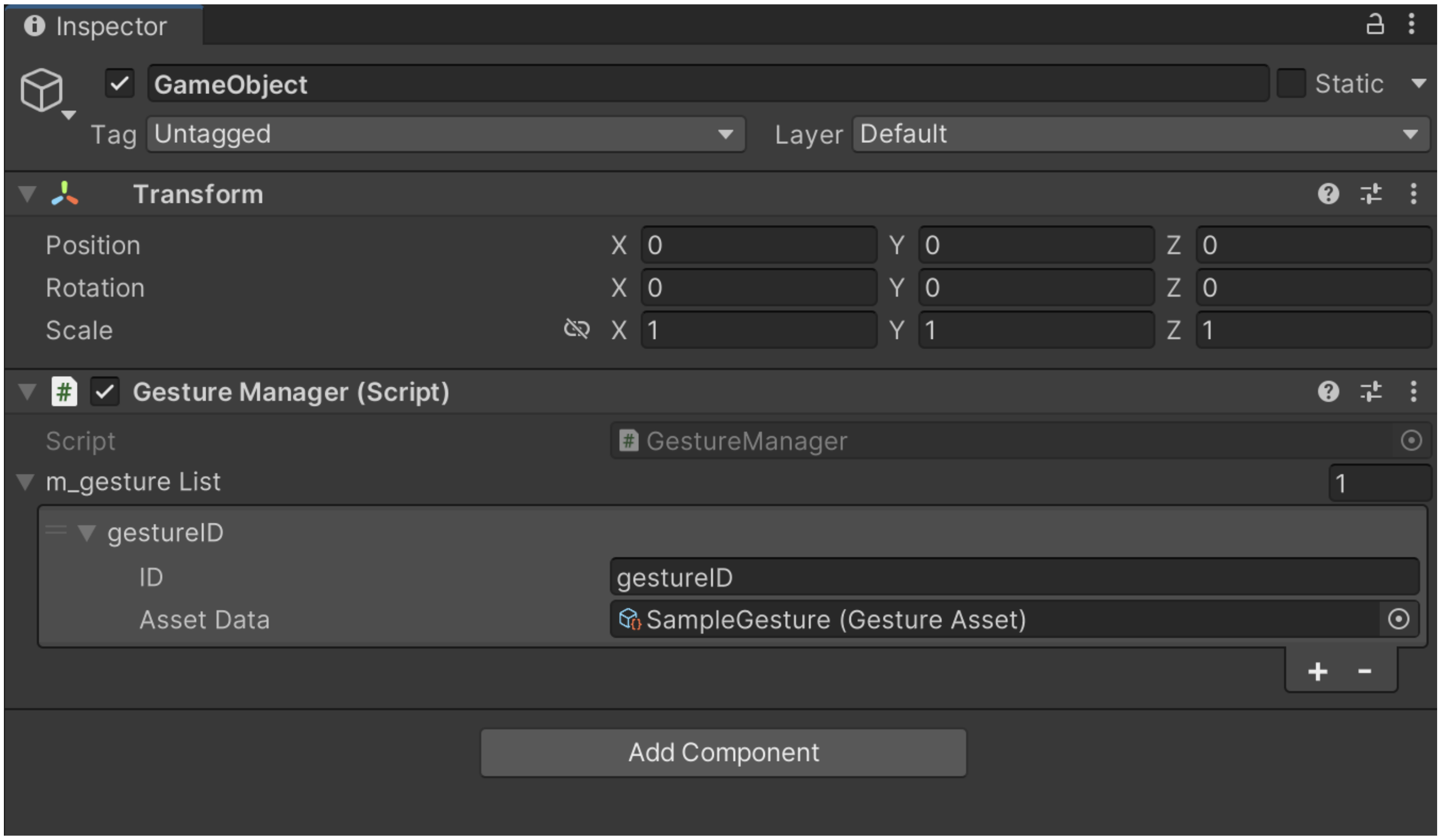
GestureAsset Settings



Add gesture conditions to ConditionAssetList. See below for more information on conditions.

[Conditions]

Setup GestureAsset in GestureManager


Inspector

GameObject   Static

Tag Untagged   Layer Default

Transform

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

Gesture Manager (Script)

Script GestureManager

m_gesture List 1

gestureID

ID gestureID

Asset Data SampleGesture (Gesture Asset)

1. AddComponent GestureManager to the GameObject.
2. Add GestureAsset and ID to gestureList. Set a unique ID so that the ID will not be covered by gestureList.

After the above settings are completed, the Script will be able to retrieve Gesture from GestureManager.

2. Get Event with Script

Result

Result is of type bool. True can be obtained if the condition set by GestureAsset is matched.

```
// Get Gesture from GestureManager
if (GestureManager.TryGetGesture("gestureID", out var gesture))
{
                // When Result is changed
                gesture.OnChangeResult()
                                .Subscribe(value =>
                                {
                                                // Current Result.
                                                value.result
                                                // Information on both hands.
                                                value.updateInfo
                                })
                                .RegisterTo(this.destroyCancellationToken);

                // When Result is true
                gesture.OnTrueResult()
                                .Subscribe(value =>
                                {
                                                // Information on both hands.
                                                value
                                })
                                .RegisterTo(this.destroyCancellationToken);

                // When Result is false
                gesture.OnFalseResult()
                                .Subscribe(value =>
                                {
                                                // Information on both hands.
                                                value
                                })
                                .RegisterTo(this.destroyCancellationToken);
}
```

GestureState

GestureState is a string type. Used in Gesture with flow. It is intended to be used for Gesture that needs to acquire a state.

```
// Get Gesture from GestureManager
if (GestureManager.TryGetGesture("gestureID", out var gesture))
{
                // When GestureState is changed
                gesture.OnChangeState()
                                .Subscribe(value =>
                                {
                                                // Updated State.
                                                value.newState
                                                // Information on both hands.
                                                value.updateInfo
                                })
                                .RegisterTo(this.destroyCancellationToken);
}
```

How to create your own Condition

To create your own Condition, ConditionAsset and ConditionInstance must be created.

ConditionAsset

ConditionAsset is the class needed to configure settings in GestureAsset's Inspector. Be sure to use Serializable as the attribute.

```
[Serializable]
public class SampleCondition : ConditionAsset<SampleConditionInstance>
{
    [field: SerializeField, Tooltip("Settings in the Inspector")]
    public int Value { get; private set; } = default;
}
```

ConditionInstance

ConditionInstance is the class that implements the condition.

```
public class SampleConditionInstance : ConditionInstance<SampleCondition>
{

    // Update conditions.
    public override void Update(IConditionUpdater.UpdateInfo updateInfo)
    {
        // ConditionAsset value can be obtained.
        Asset.Value
    }

    // Return the result of the condition
    public override bool GetResult()
    {
        return true;
    }

    // Called only once at instance
    protected override void Setup()
    {
        base.Setup();
    }

    public override void Dispose()
    {
        base.Dispose();
    }

}
```

Namespace Graffity.HandGesture

Classes

Gesture

Class that manages a single gesture

GestureAsset

Asset of conditions representing a single gesture

GestureManager

Class for managing and updating all gestures

HandInfo

Class that converts and manages library hand information

IGestureUpdater.UpdateInfo

Information required for update

JointInfo

Class that converts and manages library joint information

Interfaces

IGestureResult

Interface to retrieve whether the gesture conditions are met

IGestureState

Interface for managing gesture state

IGestureUpdater

Interface for updating gestures

Enums

HandInfo.HandType

Hand type