Input Event System – Gaskellgames

Package made for the Unity 3D game engine

USER GUIDE

Release 1.2.0

February 2025

Table of Contents:

Getting Started:

Overview

This user guide was created to provide a basic overview of the features functionality of the asset.

Installation

Once you have downloaded the asset from the Unity's Asset Store, go to: "Assets > Import Package > Custom Package…". In the Import Asset Window, find and select the package's name. After the 'import package' window appears in Unity, verify that all items to import are selected and then click the import button in the bottom right of the window.

Quick Start

The content of the asset will be found in the project window, under assets and within the toolbar options under sub-heading of **Gaskellgames**.

All content that you as the end user are expected to interact with, are components under the component sub menu of Gaskellgames, and any prefabs contained within the project files folder named Prefabs. An up-to-date copy of this guide can be found under the sub folder with the name **Documentation**. All back-end files and resources that are required to make the assts work can be found within the sub folders with the names **Editor** and **Runtime**.

Any Gaskellgames components added as part of a package can be found under the **Component** toolbar menu and the inspector's **Add Component** button. Some components will also be available to create under the **Right Click** menu under sub-heading of **Gaskellgames**.

Any Gaskellgames editor windows added as part of a package can be found under the **Tools** toolbar menu and **Window** toolbar menu options under a sub-heading of **Gaskellgames**.

Support & API documentation

Should you have any questions or require assistance, please join the official Gaskellgames Discord:

https://discord.gg/nzRQ87GGbD

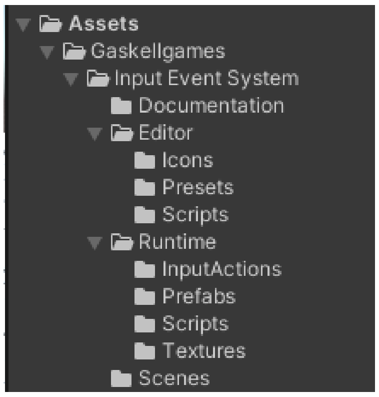
In the event you are unable to find the information you seek on the forums or discord, you can contact Gaskellgames via the weblink:

https://www.gaskellgames.com/contact

Package Content:

File Structure

The files and content within the asset are laid out in the same way as all Gaskellgames assets. You will find the asset name under the header file of **Gaskellgames**:

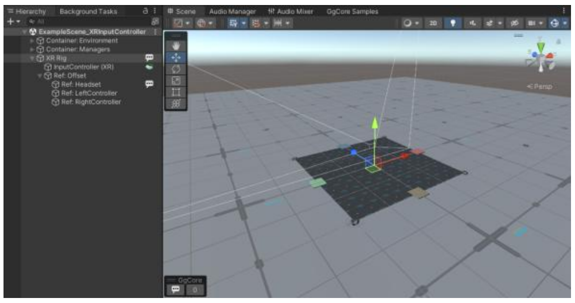


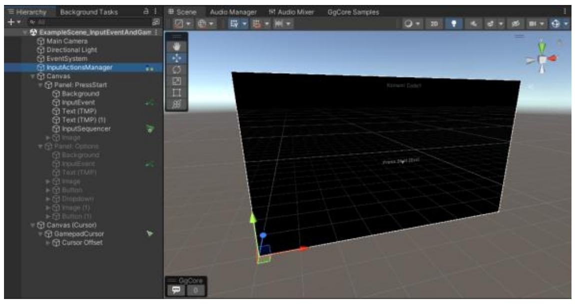
```
Assets
  Gaskellgames
    Input Event System
      Documentation
      Editor
        Icons
        Presets
        Scripts
      Runtime
        InputActions
        Prefabs
        Scripts
        Textures
      Scenes
```

The asset version's up-to-date copy of this guide can be found under the sub folder with the name **Documentation**. All editor-only content within the folder named **Editor**, and all runtime content under the folder **Runtime**. There are example scene(s) within the subfolder named **Scenes**.

Example scene

The example scenes, found within the subfolder named **Scenes**, can be viewed to see a working version of the asset. For this asset they look as follows:





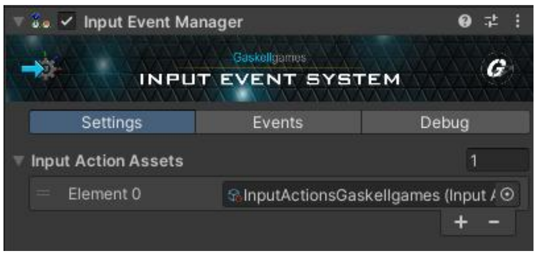
Within the scene, you will find a scene camera and directional light source, along with examples of component setups.

Please note: not all example scenes are 'playable' via the play button, and may instead be examples of setup in the editor.

How to use / setup guide

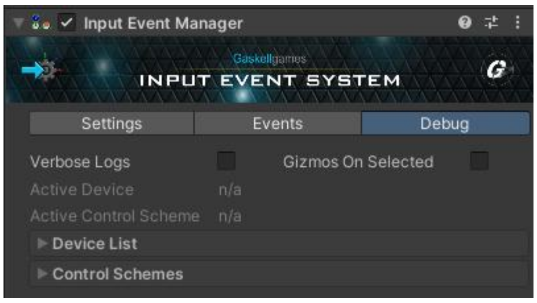
Input Event Manager:

The Input Event Manager automatically enables/disables the use of input action assets listed. This allows other scripts to freely use input action assets without having to manually enable and/or disable inputs. All available devices and control schemes are listed in the debug tab, alongside showing which are currently active.



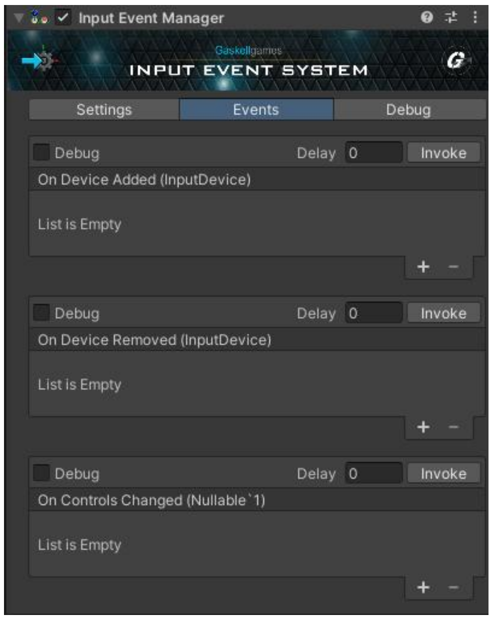
Input Action Assets 1
Element 0 InputActionsGaskellgames (Input /

(Settings Tab)



Verbose Logs Gizmos On Selected
Active Device n/a
Active Control Scheme n/a
Device List
Control Schemes

(Debug Tab)



Debug Delay 0 Invoke
On Device Added (InputDevice)
List is Empty

Debug Delay 0 Invoke
On Device Removed (InputDevice)
List is Empty

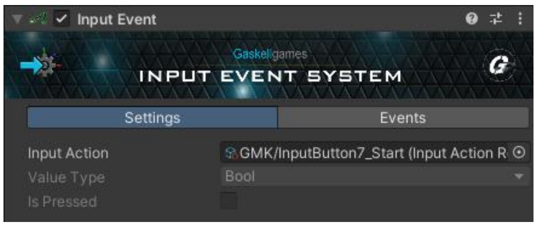
Debug Delay 0 Invoke
On Controls Changed (Nullable`1)
List is Empty

(Events Tab)

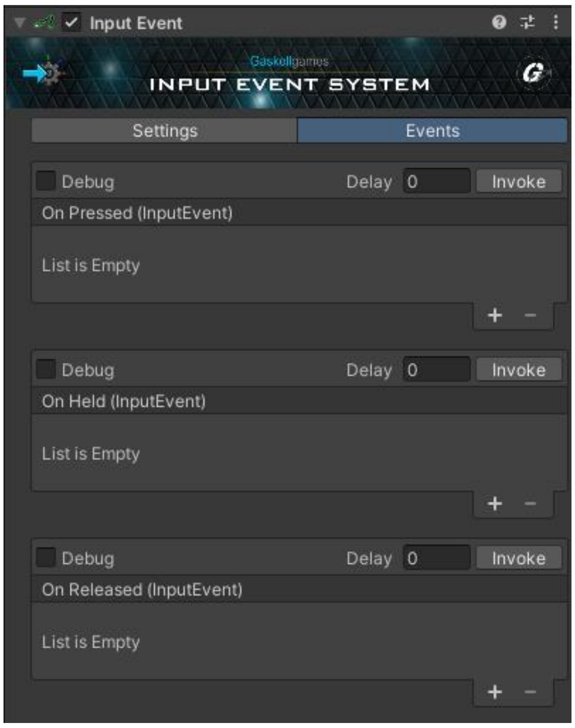
The events allow you to extend the functionality easily either via the inspector, or through listeners in your own scripts, based on callbacks for when a device is added or removed as well as when the control scheme

Input Event:

The input event script is a simple and elegant way to trigger an event when a user input is used. User input is defined by a single Input Action Reference, which can be configured to be any number of inputs via the new input system.


Input Event
Gaskellgames
INPUT EVENT SYSTEM
Settings | Events
Input Action: GMK/InputButton7_Start (Input Action R
Value Type: Bool
Is Pressed

(Settings Tab)

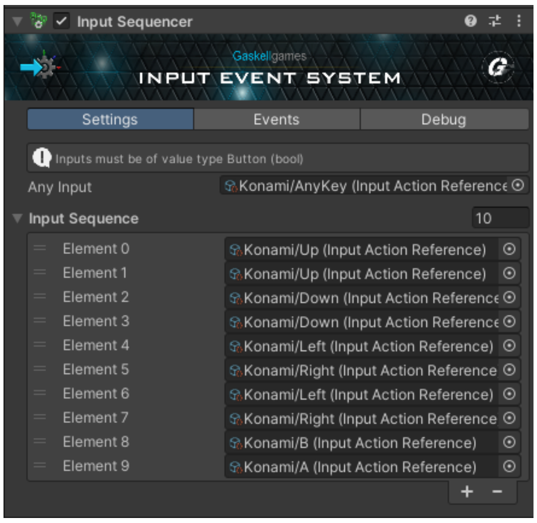

Input Event
Gaskellgames
INPUT EVENT SYSTEM
Settings | Events
Debug Delay 0 Invoke
On Pressed (InputEvent)
List is Empty
Debug Delay 0 Invoke
On Held (InputEvent)
List is Empty
Debug Delay 0 Invoke
On Released (InputEvent)
List is Empty

(Events Tab)

The events allow you to extend the functionality easily either via the inspector, or through listeners in your own scripts, based on callbacks for when the input is pressed, held or released.

Input Sequencer:

The input sequencer allows you to create a sequence of inputs in any order you define. The example below showcases how you can add the classic 'hidden code' commonly referred to as the Konami Code. *(Up, Up, Down, Down, Left, Right, Left, Right, B, A)*

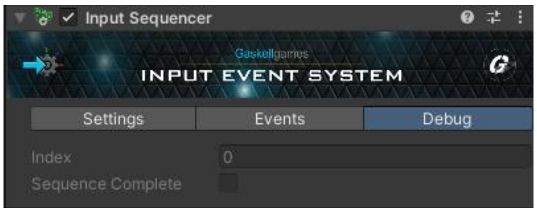

Input Sequencer

Settings | Events | Debug

Inputs must be of value type Button (bool)

Any Input	Konami/AnyKey (Input Action Reference
Input Sequence	10
Element 0	Konami/Up (Input Action Reference)
Element 1	Konami/Up (Input Action Reference)
Element 2	Konami/Down (Input Action Reference
Element 3	Konami/Down (Input Action Reference
Element 4	Konami/Left (Input Action Reference)
Element 5	Konami/Right (Input Action Reference
Element 6	Konami/Left (Input Action Reference)
Element 7	Konami/Right (Input Action Reference
Element 8	Konami/B (Input Action Reference)
Element 9	Konami/A (Input Action Reference)

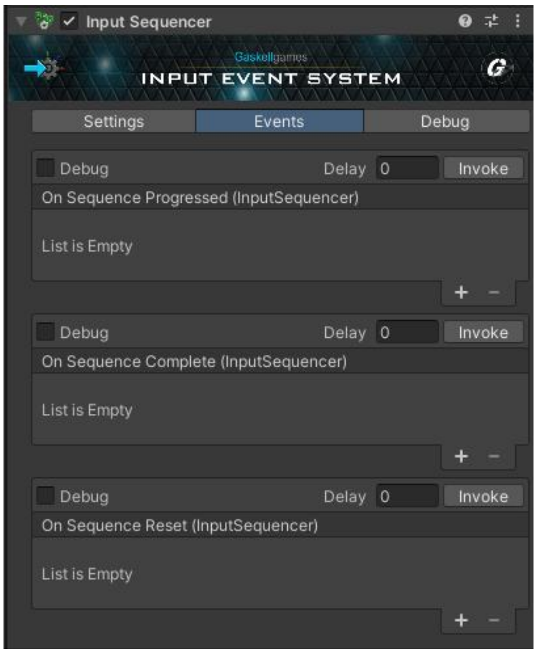
(Settings Tab)


Input Sequencer

Settings | Events | Debug

Index	0
Sequence Complete	

(Debug Tab)


Input Sequencer

Settings | Events | Debug

Debug Delay 0 Invoke
On Sequence Progressed (InputSequencer)
List is Empty

Debug Delay 0 Invoke
On Sequence Complete (InputSequencer)
List is Empty

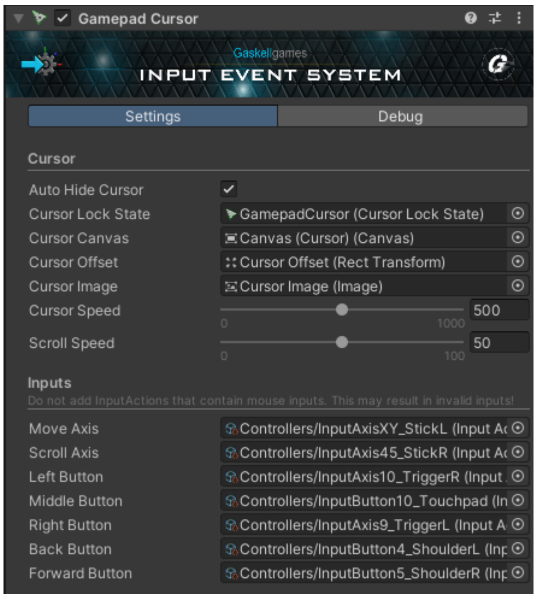
Debug Delay 0 Invoke
On Sequence Reset (InputSequencer)
List is Empty

(Events Tab)

The events allow you to extend the functionality easily either via the inspector, or through listeners in your own scripts, based on callbacks for when the sequence is progressed, completed or reset.

Gamepad Cursor:

The gamepad cursor script can be used with any input system inputs, but has been designed especially for converting gamepad input to an on-screen virtual mouse cursor.


Gamepad Cursor

Gaskellgames INPUT EVENT SYSTEM

Settings | Debug

Cursor

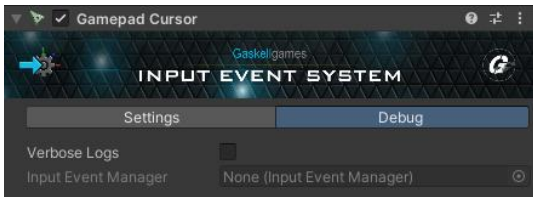
Auto Hide Cursor	✓
Cursor Lock State	GamepadCursor (Cursor Lock State)
Cursor Canvas	Canvas (Cursor) (Canvas)
Cursor Offset	Cursor Offset (Rect Transform)
Cursor Image	Cursor Image (Image)
Cursor Speed	500 (0–1000)
Scroll Speed	50 (0–100)

Inputs

Do not add InputActions that contain mouse inputs. This may result in invalid inputs!

Move Axis	Controllers/InputAxisXY_StickL (Input Ac
Scroll Axis	Controllers/InputAxis45_StickR (Input Ac
Left Button	Controllers/InputAxis10_TriggerR (Input .
Middle Button	Controllers/InputButton10_Touchpad (In
Right Button	Controllers/InputAxis9_TriggerL (Input A
Back Button	Controllers/InputButton4_ShoulderL (Inp
Forward Button	Controllers/InputButton5_ShoulderR (Inp

(Settings Tab)


Gamepad Cursor

Gaskellgames INPUT EVENT SYSTEM

Settings | Debug

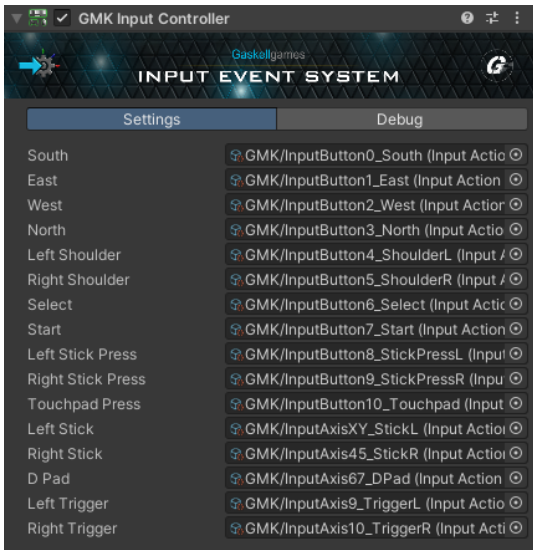
Verbose Logs	☐
Input Event Manager	None (Input Event Manager)

(Debug Tab)

This allows you to create more complex UI / menu systems that users would struggle to navigate without the use of a mouse cursor.

GMK Input Controller:

The gamepad, mouse and keyboard (GMK) input controller acts as a single point of contact for other systems to check if user input has occurred. The GMK controller will convert multiple user inputs into a standardized input layout of a gamepad.

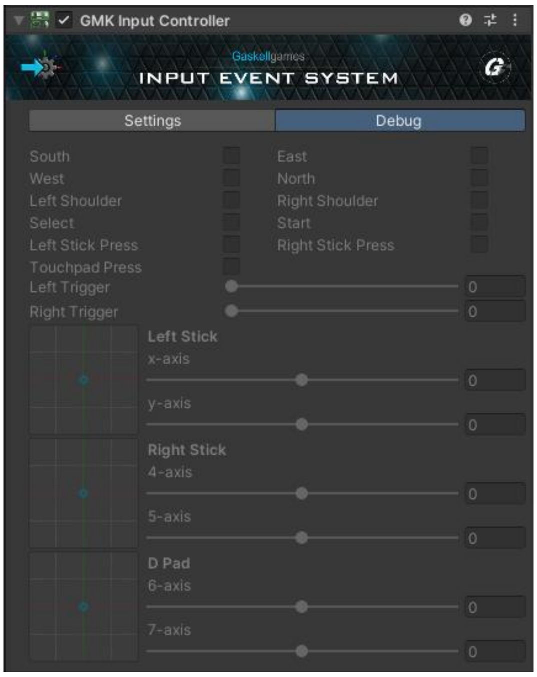

GMK Input Controller

Gaskellgames
INPUT EVENT SYSTEM

Settings | Debug

South	GMK/InputButton0_South (Input Actio
East	GMK/InputButton1_East (Input Action
West	GMK/InputButton2_West (Input Actior
North	GMK/InputButton3_North (Input Actio
Left Shoulder	GMK/InputButton4_ShoulderL (Input /
Right Shoulder	GMK/InputButton5_ShoulderR (Input /
Select	GMK/InputButton6_Select (Input Actic
Start	GMK/InputButton7_Start (Input Action
Left Stick Press	GMK/InputButton8_StickPressL (Input
Right Stick Press	GMK/InputButton9_StickPressR (Inpu
Touchpad Press	GMK/InputButton10_Touchpad (Input
Left Stick	GMK/InputAxisXY_StickL (Input Actio
Right Stick	GMK/InputAxis45_StickR (Input Actio
D Pad	GMK/InputAxis67_DPad (Input Action
Left Trigger	GMK/InputAxis9_TriggerL (Input Actio
Right Trigger	GMK/InputAxis10_TriggerR (Input Acti

(Settings Tab)


GMK Input Controller

Gaskellgames
INPUT EVENT SYSTEM

Settings | Debug

South	East
West	North
Left Shoulder	Right Shoulder
Select	Start
Left Stick Press	Right Stick Press
Touchpad Press	

Left Trigger 0
Right Trigger 0

Left Stick
x-axis 0
y-axis 0

Right Stick
4-axis 0
5-axis 0

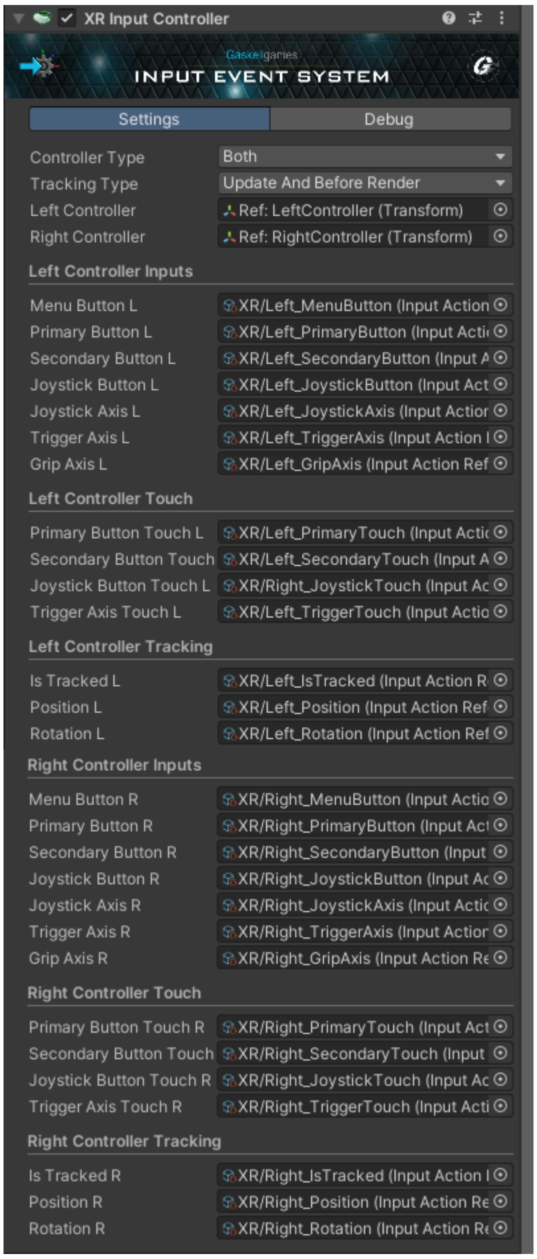
D Pad
6-axis 0
7-axis 0

(Debug Tab)

This system allows your other scripts to use a standardised input layout, no mater what input device or control scheme is currently in use. Other systems and scripts can use this standardised input for gamepads, via a reference to the "User Inputs" variable.

XR Input Controller:

The extended reality (XR) input controller simplifies setup of XR character rigs. Controller tracking, touch inputs and physical inputs are all standardised for both left and right controllers.



XR Input Controller

Gaskellgames INPUT EVENT SYSTEM

Settings | Debug

Controller Type	Both
Tracking Type	Update And Before Render
Left Controller	Ref: LeftController (Transform)
Right Controller	Ref: RightController (Transform)

Left Controller Inputs

Menu Button L	XR/Left_MenuButton (Input Action
Primary Button L	XR/Left_PrimaryButton (Input Acti
Secondary Button L	XR/Left_SecondaryButton (Input A
Joystick Button L	XR/Left_JoystickButton (Input Act
Joystick Axis L	XR/Left_JoystickAxis (Input Actior
Trigger Axis L	XR/Left_TriggerAxis (Input Action I
Grip Axis L	XR/Left_GripAxis (Input Action Ref

Left Controller Touch

Primary Button Touch L	XR/Left_PrimaryTouch (Input Actic
Secondary Button Touch	XR/Left_SecondaryTouch (Input A
Joystick Button Touch L	XR/Right_JoystickTouch (Input Ac
Trigger Axis Touch L	XR/Left_TriggerTouch (Input Actio

Left Controller Tracking

Is Tracked L	XR/Left_IsTracked (Input Action R
Position L	XR/Left_Position (Input Action Ref
Rotation L	XR/Left_Rotation (Input Action Ref

Right Controller Inputs

Menu Button R	XR/Right_MenuButton (Input Actio
Primary Button R	XR/Right_PrimaryButton (Input Act
Secondary Button R	XR/Right_SecondaryButton (Input
Joystick Button R	XR/Right_JoystickButton (Input Ac
Joystick Axis R	XR/Right_JoystickAxis (Input Actic
Trigger Axis R	XR/Right_TriggerAxis (Input Actior
Grip Axis R	XR/Right_GripAxis (Input Action Re

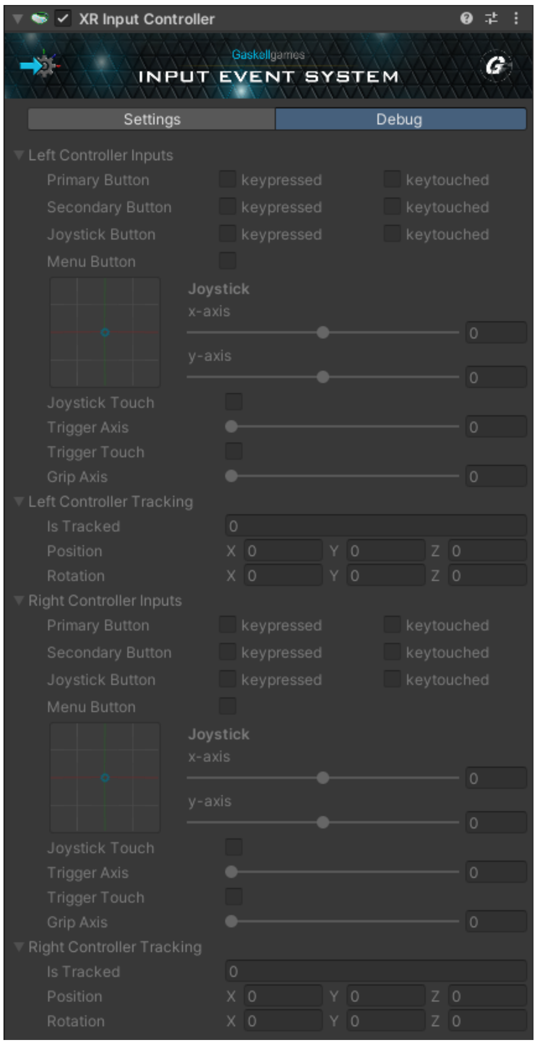
Right Controller Touch

Primary Button Touch R	XR/Right_PrimaryTouch (Input Act
Secondary Button Touch	XR/Right_SecondaryTouch (Input
Joystick Button Touch R	XR/Right_JoystickTouch (Input Ac
Trigger Axis Touch R	XR/Right_TriggerTouch (Input Acti

Right Controller Tracking

Is Tracked R	XR/Right_IsTracked (Input Action I
Position R	XR/Right_Position (Input Action Re
Rotation R	XR/Right_Rotation (Input Action Re

(Settings Tab)



XR Input Controller

Gaskellgames INPUT EVENT SYSTEM

Settings | Debug

Left Controller Inputs

Primary Button	keypressed	keytouched
Secondary Button	keypressed	keytouched
Joystick Button	keypressed	keytouched
Menu Button		

Joystick
x-axis 0
y-axis 0

Joystick Touch	
Trigger Axis	0
Trigger Touch	
Grip Axis	0

Left Controller Tracking

Is Tracked	0		
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0

Right Controller Inputs

Primary Button	keypressed	keytouched
Secondary Button	keypressed	keytouched
Joystick Button	keypressed	keytouched
Menu Button		

Joystick
x-axis 0
y-axis 0

Joystick Touch	
Trigger Axis	0
Trigger Touch	
Grip Axis	0

Right Controller Tracking

Is Tracked	0		
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0

(Debug Tab)

Other systems and scripts can use this standardised input for either left or right controllers, via a reference to the "Left Controller Inputs", "Right Controller Inputs", "Left Controller Tracking" or "Right Controller Tracking" variables.