



Input Event System - Gaskellgames

Package made for the Unity 3D game engine

USER GUIDE

Release 1.3.0

April 2025

Copyright (C) 2022-2025 Gaskellgames - All Rights Reserved





This manual, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this manual is furnished for informational use only, is subject to change without notice and should not be construed as a commitment by its authors. The author assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.





Table of Contents:

Getting Started

Overview.....	3
Installation.....	3
Quick Start.....	3
Support & API documentation.....	3

Package Content

File structure.....	4
Example scene.....	4
How to use / setup guide.....	5





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:

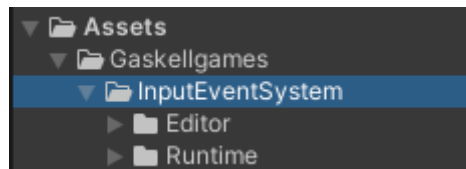
<https://www.gaskellgames.com/contact>



Package Content:

File Structure

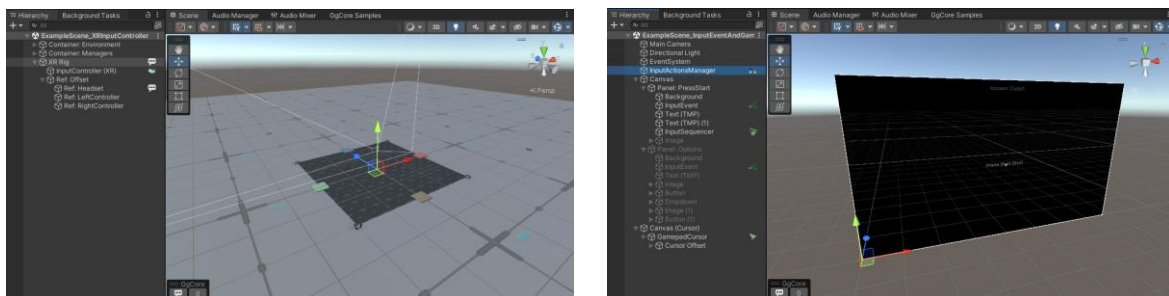
The files and content within the asset are laid out in the same way as all Gaskellgames assets. You will find the asset name under the header file of **Gaskellgames**:



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:



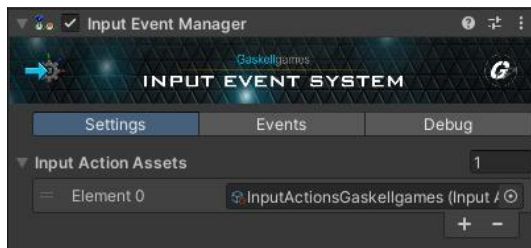
Within the scene, you will find a scene camera and directional light source, along with examples of component setups.

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

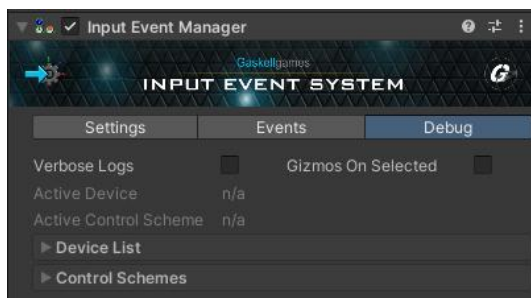
How to use / setup guide

Input Event Manager:

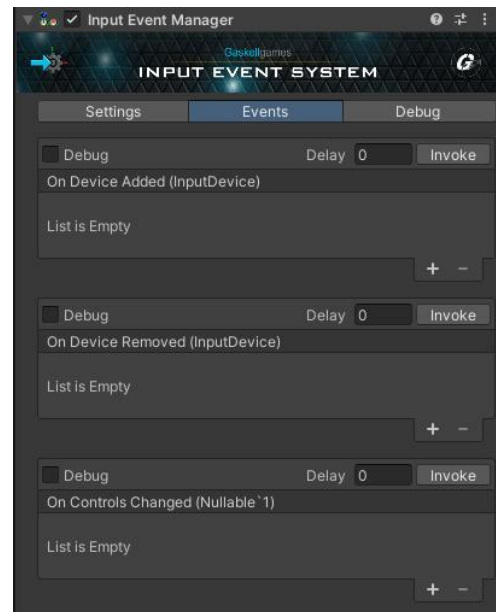
The Input Event Manager automatically enables/disables the use of input action assets listed. This allows other scripts to freely use input action assets without having to manually enable and/or disable inputs. All available devices and control schemes are listed in the debug tab, alongside showing which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



(Settings Tab)



(Debug Tab)

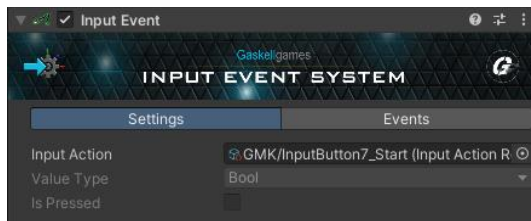


(Events Tab)

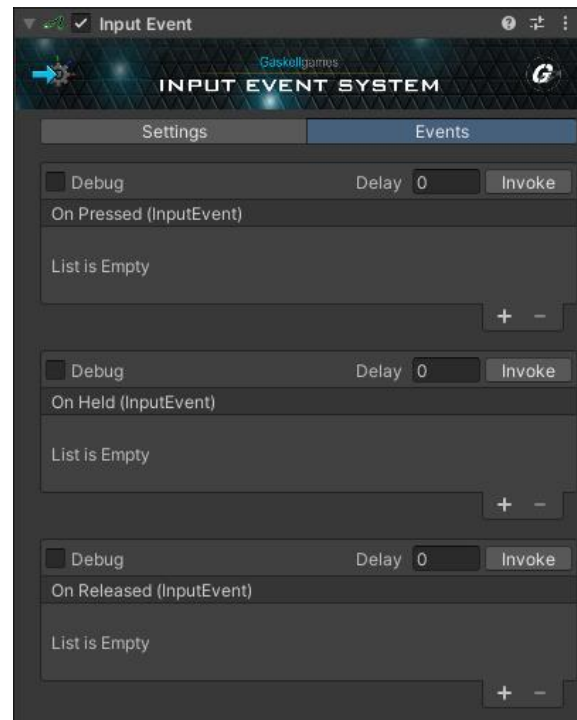
The events allow you to extend the functionality easily either via the inspector, or through listeners in your own scripts, based on callbacks for when a device is added or removed as well as when the control scheme

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.



(Settings Tab)

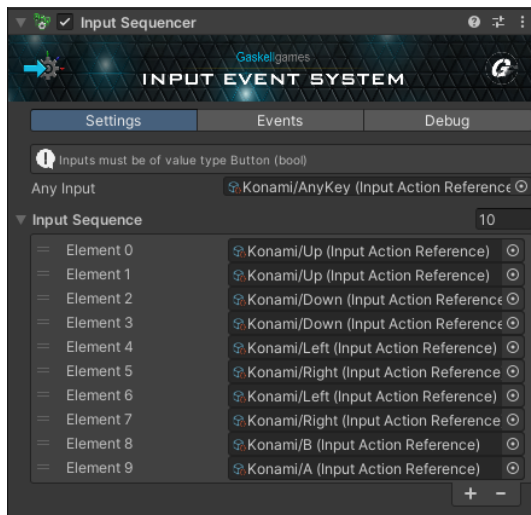


(Events Tab)

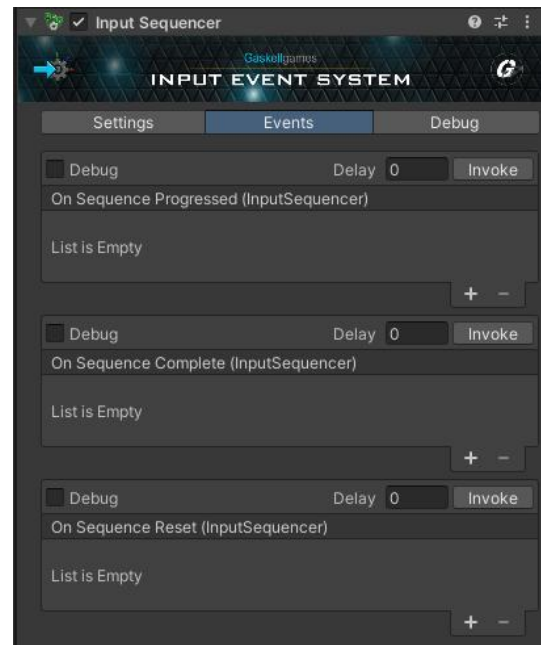
The events allow you to extend the functionality easily either via the inspector, or through listeners in your own scripts, based on callbacks for when the input is pressed, held or released.

Input Sequencer:

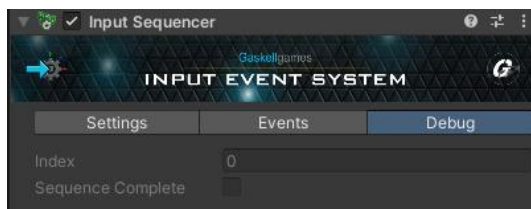
The input sequencer allows you to create a sequence of inputs in any order you define. The example below showcases how you can add the classic 'hidden code' commonly referred to as the Konami Code. (*Up, Up, Down, Down, Left, Right, Left, Right, B, A*)



(Settings Tab)



(Events Tab)

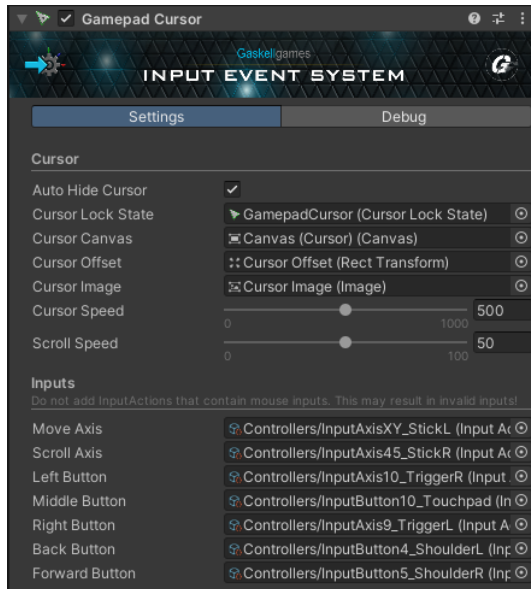


(Debug Tab)

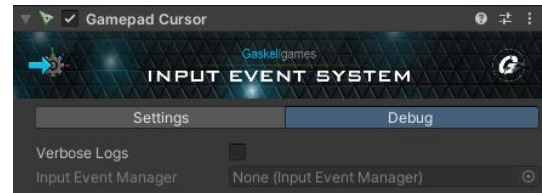
The events allow you to extend the functionality easily either via the inspector, or through listeners in your own scripts, based on callbacks for when the sequence is progressed, completed or reset.

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.



(Settings Tab)

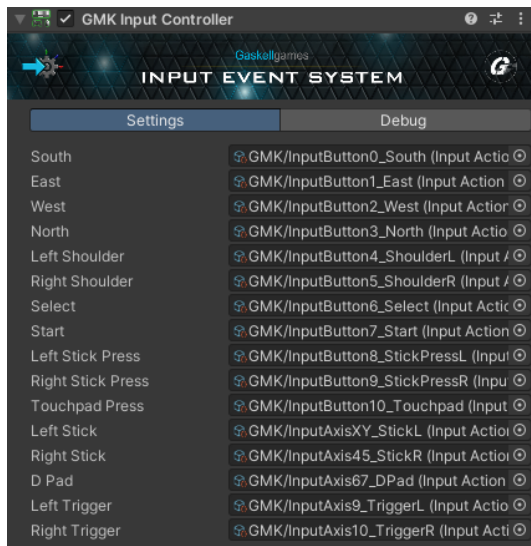


(Debug Tab)

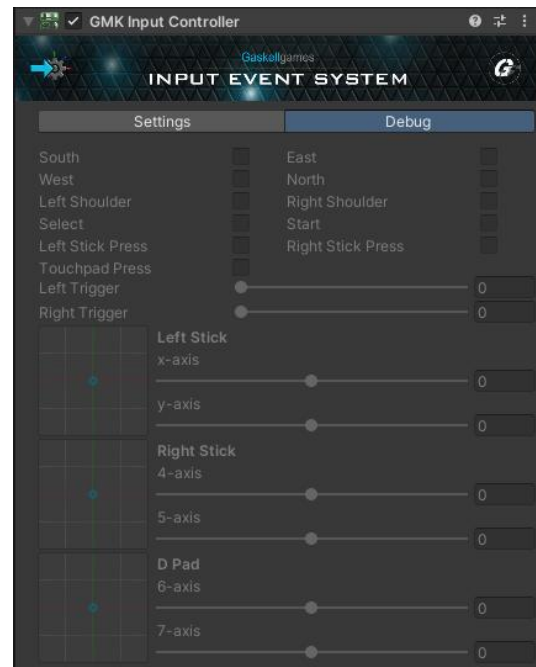
This allows you to create more complex UI / menu systems that users would struggle to navigate without the use of a mouse cursor.

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.



(Settings Tab)

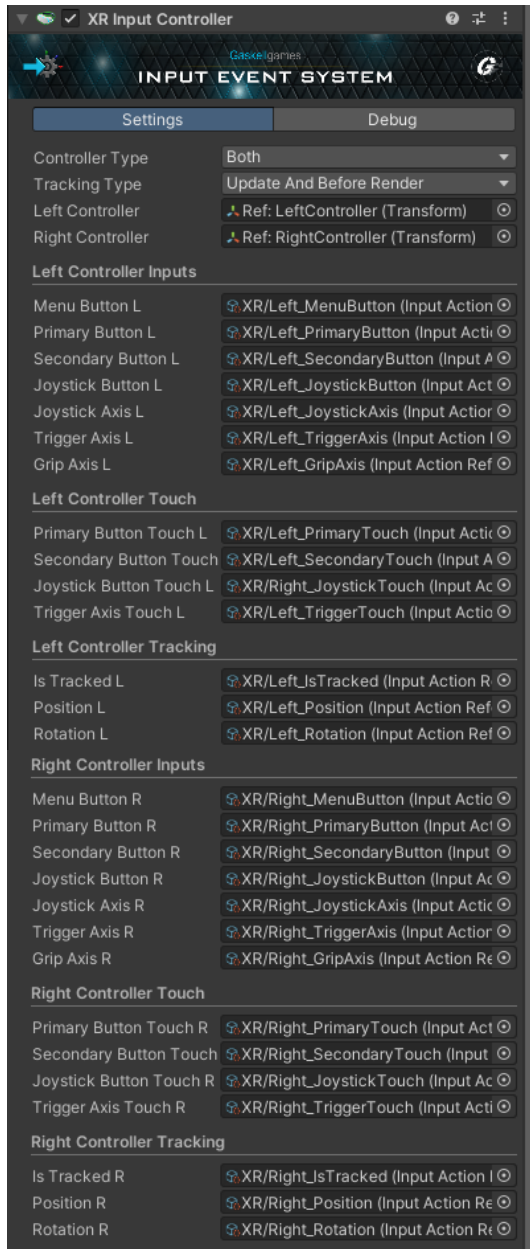


(Debug Tab)

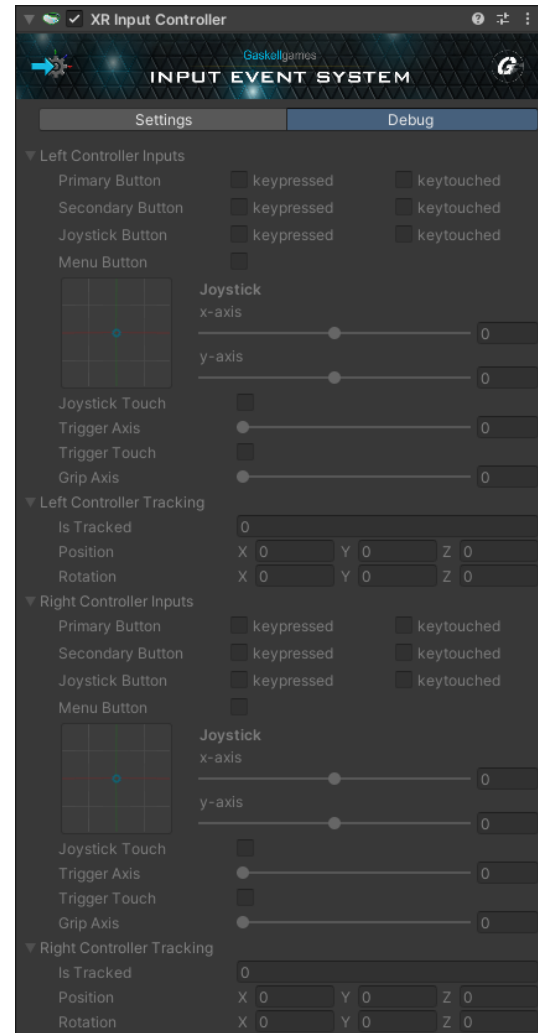
This system allows your other scripts to use a standardised input layout, no matter what input device or control scheme is currently in use. Other systems and scripts can use this standardised input for gamepads, via a reference to the "User Inputs" variable.

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.



(Settings Tab)



(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.