

NisGab

New Input System Global Action Binder

OVERVIEW

This is a tool to quickly generate a performant, static event bindings for input action maps in Unity's new input system

HOW TO USE

- Configure an Input action asset with all the input action maps you will need for your game. There is a default one provided in the package under NisGab/Config
- Once your asset is configured and saved, right-click the input action asset. Select the option "Generate Input Action Binding"
- It may take a moment, but a folder should then be generated under NisGab/Scripts/Generated. In there you will see the following:
 - InputEvent.cs - this is a lazy singleton that allows the developer to enable/disable input action maps at will as well as serves as a container for all static events
 - For each input action map in your input action asset, you will see {MapName}InputActions.cs - this functions as an isolated container for the static events for the associated input action map. It relays the events from the singleton instance to the static events
- Once the code is generated, one or more of the input action maps must be enabled in order to start receiving events. This is done simply by calling `InputEvent.Instance.Enable[MapName]Input();`
- Once action maps are enabled, events can be registered to with logic such as:
`InputEvent.UI.Click += OnUIClick;`
`Private void OnUIClick(InputAction.CallbackContext context) {...event code goes here...}`

NOTES

While input action maps can be enabled/disabled at will, I personally recommend that developers contain this functionality in some kind of automated process like a game mode manager or scene transition manager such that you have a single point of context-switching where changing input action maps makes the most sense and is easiest to debug