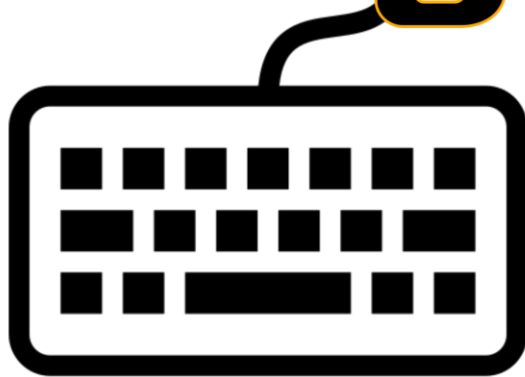


Custom Input Manager



TrickShotAssets

Content

Introduction.....	3
What this Asset does	3
How to set it up.....	3
File Settings.....	3
File Values.....	3
How to use it within your own scripts	4
Methods.....	4
How to set the keys from your own scripts	5
How to save the values to the config	5
How to load the values from the config.....	5
How to use the script for a Movementcontroller.....	5
How to use the inbuilt Keybind Editor	6
Contact	7
Changelog	7
Version 1.0.....	7
New version coming soon.....	7

Introduction

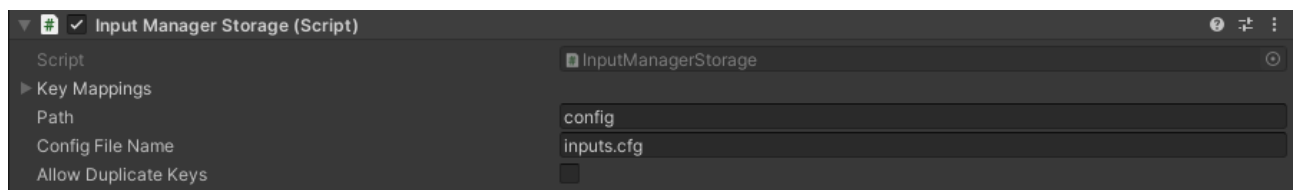
What this Asset does

This Asset can be used to create a config file for your Inputs (eg. Keybinds). You can easily add Keycodes, Default Keycodes as well as their respective names to the config. You have the option to allow/block the use of one Key for multiple Inputs.

How to set it up

File Settings

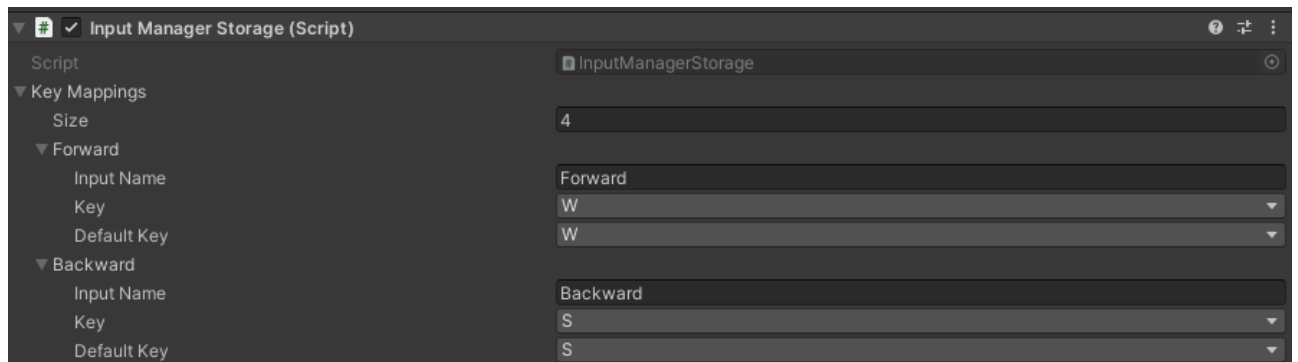
First, you need to create a new Game Object and attach the script “InputManagerStorage.cs” to it. This GameObject will be a singleton and always be tagged with “DontDestroyOnLoad”. You need to add this Object to the first Scene where your game accesses the keybinds. In the next step you will need to define the path and the name with the file extension of the config. The path will always be a path with the game’s root folder as root.



In this Example the config is called “inputs.cfg” and is located at “/pathToYourGame/config/”.

File Values

This is the most important section. Here you can populate your initial config with a user-changeable key, the default key as well as the name of the Input.



The Key name is the name of the “Input”. It could be something like “Forward”. The Key is your predefined key, which the user can change. The Default Key represents the key which will be used to restore the user-defined one on a config-reset.

How to use it within your own scripts

First, you need to reference the “InputManagerStorage.cs” Script in your own one. For this you must import the given NameSpace “TrickShotAssets.CustomInputManager”. And Assign the Instance of the Script to it.

```
using System.Collections;
using System.Collections.Generic;
using TrickShotAssets.CustomInputManager;
using UnityEngine;

Unity-Skript | 0 Verweise
public class SceneSwitcher : MonoBehaviour
{
    private InputManagerStorage inputManagerStorage;

    Unity-Nachricht | 0 Verweise
    private void Start()
    {
        inputManagerStorage = InputManagerStorage.Instance;
    }
}
```

Methods

If you have done everything correctly, you can now call the Methods below.

- GetIndexByName(string inputName)
- GetKeycodeByName(String inputName)
- ResetAllKeyMappings()
- SaveToXML()
- LoadFromXML()

The first Method “GetIndexByName(string inputName)” returns the index of the entry with the given InputName. The second Method returns the Keycode by searching for a specific InputName. The Method “ResetAllKeyMappings()” resets all Keybinds to the default keys defined in the Inspector.

How to set the keys from your own scripts

- To set a key, you need to call the reference of the “InputManagerStorage” and assign a KeyCode to the variable key in the keyMappings List.

Example 01:

```
inputManagerStorageInstance.keyMappings[i].key = YourNewKeyCode;
```

Note: You can get the index i by using the previously mentioned method `GetIndexByName(string inputName)`

Example 02:

```
inputManagerStorageInstance.keyMappings[inputManagerStorageInstance.  
GetKeycodeByName("Your InputName eg. Forward")].key = YourNewKeyCode;
```

- The same also works for the default key. Instead of using the `keyMappings[i].key`, use the `keyMappings[i].defaultKey`.
- For the InputName you will use `keyMappings[i].inputName`

How to save the values to the config

If you want to save the values from the script to the config, you will only need to call the method `SaveToXML()`.

How to load the values from the config

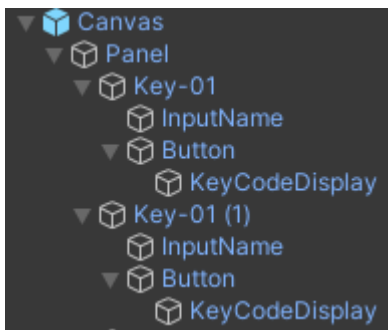
If you want to manually load the values from the config into your game, you will have to call the method `LoadFromXML()`.

How to use the script for a Movementcontroller

```
if (Input.GetKey(inputManagerStorage.keyMappings[inputManagerStorage.GetIndexByName("Forward")].key))  
{  
    :  
    Player.transform.Translate(Vector3.forward * Time.deltaTime);  
}
```

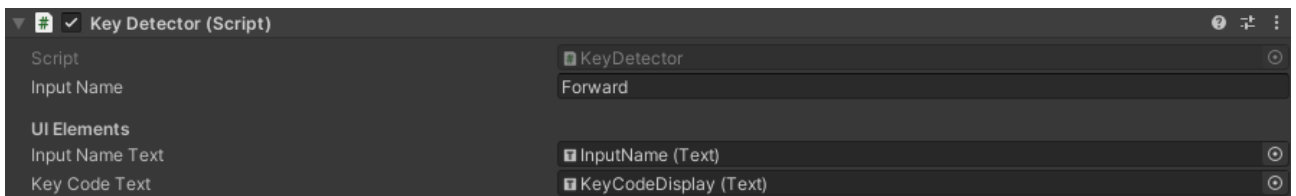
Example for implementation in a movement controller

How to use the inbuilt Keybind Editor



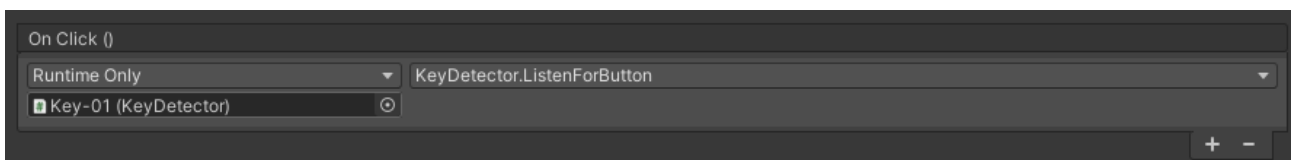
This is an example structure in your Settings UI

To add a Button for a specific Input, you need to create a parent object (eg. Key-01 *in figure above*) and add the KeyDetectorScript.



After that you need to create a Text (eg. InputName) where the name of the given Input will be shown. You also need to create a Button. After you have done that, you will have to reference these two Texts in the “KeyDetector” by dragging it into the specific field.

The last thing you need to do is to add the Script to the Button’s OnClick() Method as shown below.



Contact

If you have any further questions, visit <http://trickshotdev.ddns.net/en-us/contact.html> and send me a message. I will come back to you as fast as possible.

Changelog

Version 1.0

Initial Release of the Asset

New version coming soon