

CUSTOM INPUT MANAGER v1.4

BEFORE WE START...

Before we go on i would like to thank a few people first...

Ian Kirkland: for a bug fix and some help

BLi'TZWING & NCarter: for never ending support and help

Colin : for being the first one to donate something -thanks man ,you made my day :)

Kristel : for putting up with me !

And finally YOU ! : For checking this custom input manager out, i hope you find a good use for it.
And if you do, i gladly accept a small donation. My paypal adres is:

ward.dewaele@pandora.be



WHAT'S THIS ?

'What is this custom input manager', i hear you say , ' Doesn't unity comes with a good input manager all ready' ?

The answer is yes and no. Unity comes with a good input manager if you don't require changing inputs on the fly. Unity's default input manager only allows you to change inputs at the start of your game. Another problem is that the default input manager can't be called from a web-based game. And did i mention it also handles the xbox360 controller fine ?

Both problems get tackled by the custom input manager,now at version 1.4 !

The custom input manager allows you to call a menu where you can reconfigure your input keys.

HOW DO I USE THIS ?

First off you gotta know that if you use the custom input manager you won't be able to use the default input manager anymore,but there is no reason for that anyway.

To start with you can open one of the 2 example scenes and play with it.

But my guess is that you rather use this in your own projects ,so allow me to explain you how to do that in a few simple steps.

- **Step1:** - Close your project and unity !
 - - Copy over the 'Input manager.asset' that comes with this input manager over the
 - 'Input manager.asset' from your project,these are located in the Library sub folder.
- **Step2:** - Open your project in unity and import the 'input manager' unitypackage.
- **Step3:** - Drag the 'InputManagerController' prefab that just got imported into you scene
- **Step4:** - In the script where you have your controller code, add `'public custom_inputs inputManager'` above your functions,then select the 'InputManagerController' object in the inspector.
- **Step5:** - Now you can,in the script where you handle your controller code, use following commands...
 - `inputManager.isInput[0]` - to check if input 0 is pressed
 - `inputManager.isInputDown[0]` - to check if input 0 is pressed down
 - `inputManager.isInputUp[0]` - to check if input 0 is released
 - **example:**
 - ```
if (inputManager.isInput[0])
{
 // enter your controlcode here
}
```
- **Step6:** - Customize the input manager in the inspector on the script attached to 'InputManagerController'.  
More about this in the next chapter !

## CUSTOMISE THE INPUTMANAGER !

Customising the custom input manager we do in the inspector. It can't be much easier then this. Select the 'InputManagerController' object in your project so the settings get visible in the inspector. Here you can change the appearance and the settings of the input manager. Let's go over the list.

**InputManagerLogo:** Logo that you want to show above your input keys.

**DescriptionString:** The size will be the amount of different controls you want to assign. Fill in the names of each control on each element.

**Default\_InputKeys:** The default inputs you want to assign. Make sure the length of this array is the same as the **DescriptionString** !!!

**Alt\_Default\_InputKeys:** The alternative default inputs you want to assign. Make sure the length of this array is the same as the **DescriptionString** !!!

**Boxes\_Y:** Start coordinate on the Y axis where the buttons will start.

**BoxesMargin\_Y:** Space between the boxes in the Y axis.

**DescriptionSize:** Size of the description box in the X axis.

**buttonsize:** Size of the buttons in the X axis.

**resetbuttonLocY:** Y coordinate of the reset button.

**resetbuttonText:** Reset button text.

**mouseAxisOn:** Allow mouse axis as inputs.

**mouseButtonsOn:** Allow mouse buttons as inputs.

**allowDuplicates:** Allow duplicate inputs to be entered.

**OurSkin:** GUI skin to customise the looks.

**analogFeel\_gravity:** This can be used to emulate the virtual gravity feel from the default input manager

**analogFeel\_sensitivity:** This can be used to emulate the virtual sensitivity feel from the default input manager

### **NOTES:**

- The analogfeel\_gravity & \_sensitivity work fine when used right, but don't always have to be used. This input manager is made to handle the inputs. The controls you have to work out yourself to get them perfect.

- When you change the size of the **DescriptionString/Default\_InputKeys/Alt\_Default\_InputKeys** you might have to click on 'reset to defaults' once. This will **NOT** happen ingame because the player won't be able to change the size of the arrays ofcourse.

## **LEGAL NOTICE AND FUTURE...**

Although i tried to make sure this script works fine, i am not liable for any damage resulting from use of this script/software. If you notice any bugs or problems please email me at [ward.dewaele@pandora.be](mailto:ward.dewaele@pandora.be).

I spend a lot of time in making this script and even i don't needed any money for it, i do appreciate donations. Actually it would help me out a lot. Even if it's just a few bucks.

so once more...

[ward.dewaele@pandora.be](mailto:ward.dewaele@pandora.be)



I have plans to extending this input manager to also handle screen resolutions, audio controls,etc..

If i would decide the extended version would not be free, then everyone who ever donated would get it for free anyways. No matter how little the donation.

### **Links:**

**Unity forum post :** <http://forum.unity3d.com/viewtopic.php?t=57501&start=0&sid=c2609f2284ea720c3df4ee9f92f37a80>

**Download link:**

<http://roidz.weebly.com/>