



Insane Systems Input Manager Guide

Content

1. How this asset can be used?
2. Part 1. Deploying UI
3. Part 2. Setting up the input actions
4. Part 3. How to use new Input Manager in your code
5. Autodetect feature
6. Contacts

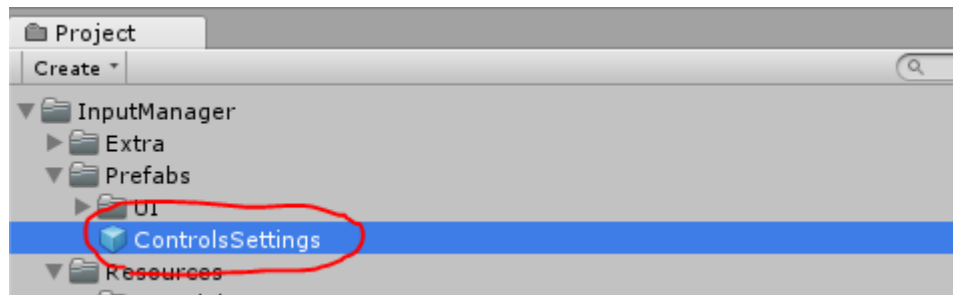
How this asset can be used?

You can use it to add controls settings to your game without any coding needed for it. Next you need just to use methods from **Insane Systems Input Manager (ISIM)** in code, where you read input.

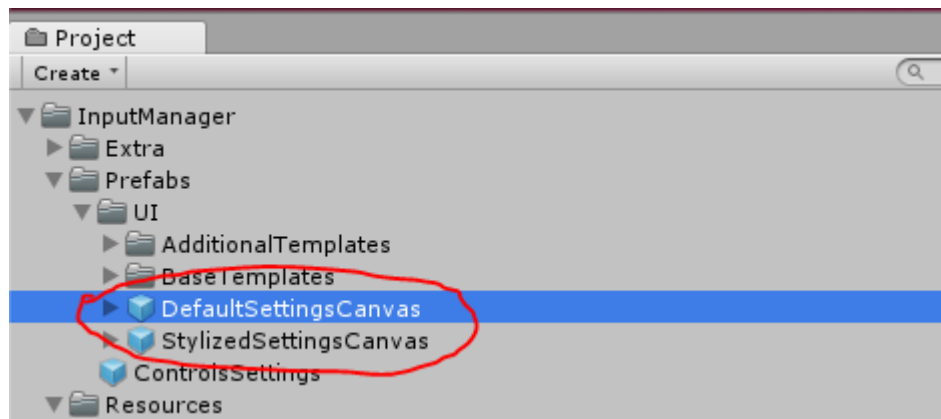
Here you can find detailed guide how to use each component of **ISIM**.

Part 1. Deploying UI

To start use **ISIM** asset, just drag & drop to your menu scene **ControlsSettings** prefab from Prefabs folder:



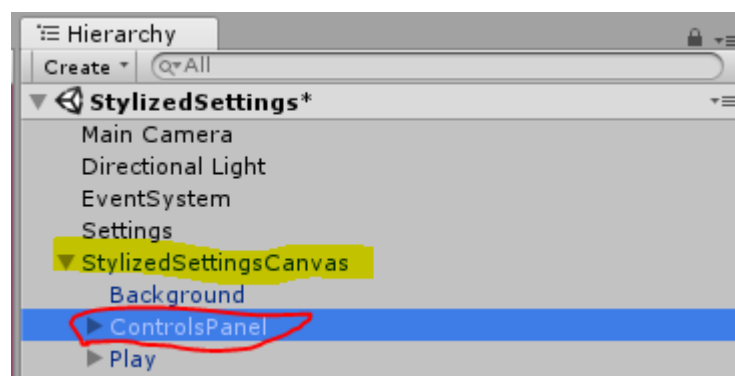
Now you need user interface which allow players to change controls settings. You can use one of default prefabs: **DefaultSettingsCanvas** or **StylizedSettingsCanvas**. They can be found in Prefabs/UI folder:



Stylized canvas is same to default canvas, just have better graphics (example of customization).

After you did this steps, you can start Play Mode and test controls settings.

If you want to use our controls UI in your own Canvas, you need to move **ControlsPanel** object from our canvas to yours:

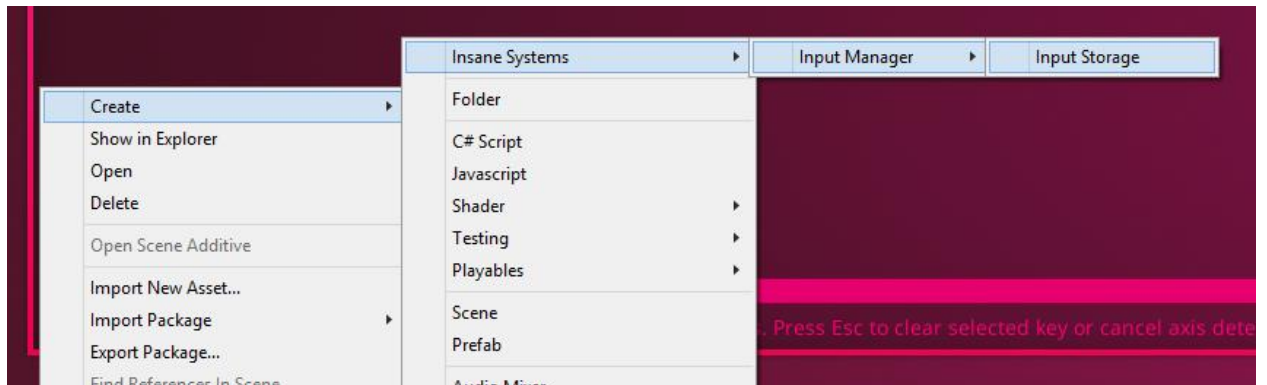


Now you have controls settings in your game!

If you have troubles, check the example scene named **BasicSettings** or **StylizedSettings**.

Part 2. Setting up the input actions

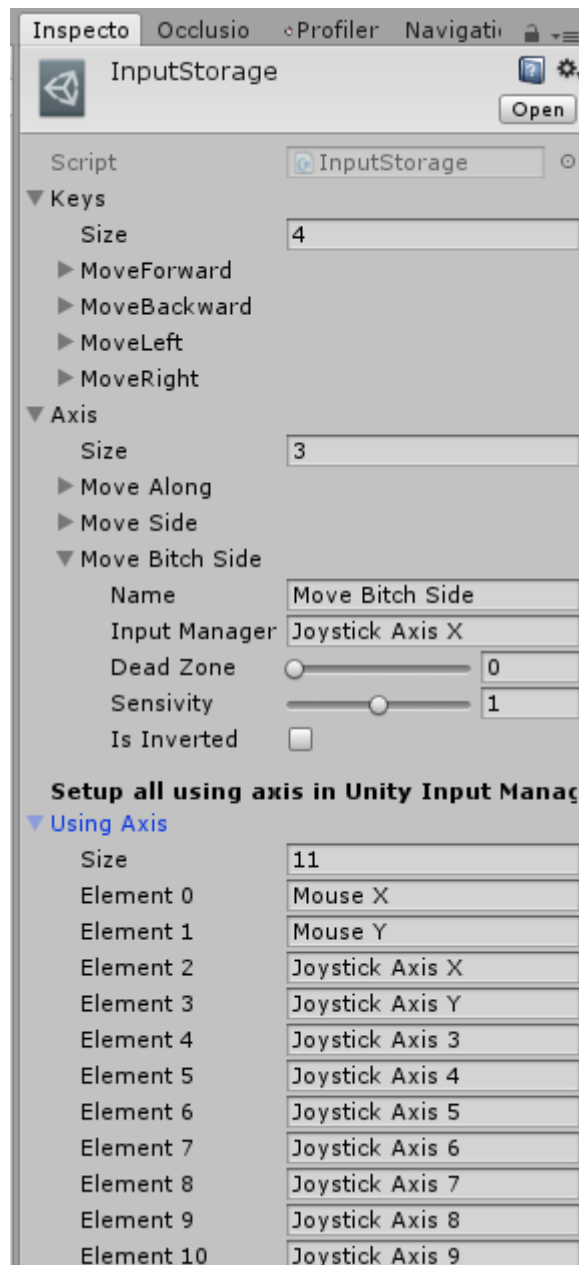
Now you need to set up key and axis actions. Player will be able to choose his own keys/joystick axis for actions, but to let do it to him, firstly we need to **have** this actions. So, you can set up it in **ScriptableObject** asset named **InputStorage**. You can create it in RMB create menu, option **Insane Systems – Input Manager – Input Storage**:



By the way, there should be default **InputStorage**, you can use it to set up your custom key actions.

Note: Game should have only one **InputStorage** asset, named as **InputStorage** and placed in **Resources** folder. You can change this in code, if you want.

Next, select your **InputStorage** file and check the **Inspector** window.



You can see **three** primary options: **Keys**, **Axis** and **Using Axis**.

In **Keys** partition you can add custom keys actions, which you need.

Axis partition is same, but needed for axis. Perfect for joystick or gamepad axis. But can be used for keyboard and mouse too.

Using axis contains list of axis from **default Unity Input Manager**. Here you should place all axis, which can be used in your game. Player will be able to select any of them in controls configuration.

Part 3. How to use new Input Manager in your code

It is very simple and looks similar to the default Unity Input System. Just add using `InsaneSystems.InputManager;` in the beginning of your scripts

where you want to use new input system. Now you will be able to use **Insane Systems Input Manager**.

Example 1 – Key down

Standart code:

```
Input.GetKeyDown(KeyCode.Escape);
```

New code:

```
InputController.GetKeyActionIsDown("Cancel");
```

Where "Cancel" is your key action name (from Input Storage).

Example 2 – Axis value

Standart code:

```
Input.GetAxis("Mouse ScrollWheel");
```

New Code:

```
InputController.GetAxisActionValue("Mouse ScrollWheel");
```

Again: **Mouse ScrollWheel** is from **Axis** option from **InputStorage**.

Note: You also can use methods **GetKeyActionIsActive** (same to Input.GetKey), **GetKeyActionIsUp** (same to Input.GetKeyUp).

Autodetect feature

This asset has autodetect feature for **Axis** settings. It means, that player can press Autodetect button and axis which he inputting will be detected automatically. It works like key setup in **Key** settings, but for axes.

Contacts

You can ask your questions or send your suggestions to us. ☺

To contact use email **godlikeaurora@gmail.com**