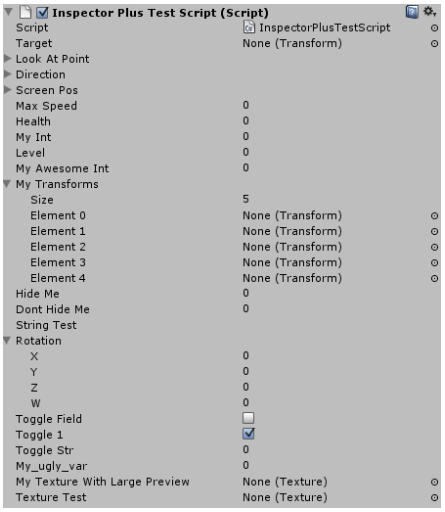
Thank you for purchasing Inspector++. Well, I hope you did. If you didn't: arrr matey, no hard feelings!

No one likes reading (or is that just me) so to get an impression of how this works, watch the video here! http://youtu.be/2lxgnJu2wJo



Default inspector...

More detailed info:

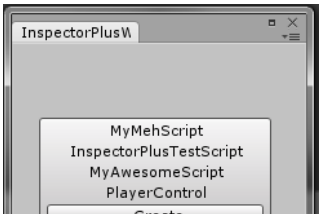
Inspectors

Unity has so called 'inspectors'. This is the window which shows all the components of your gameobject. It's great of unity to let you edit variables that way, but it's not really the most... practical window. Inspector++'s goal is to improve that!

Set-up

As soon as you imported this asset you can start creating inspectors! Open the Inspector++ window via the Window->Inspector++ menu.

Adding Inspectors

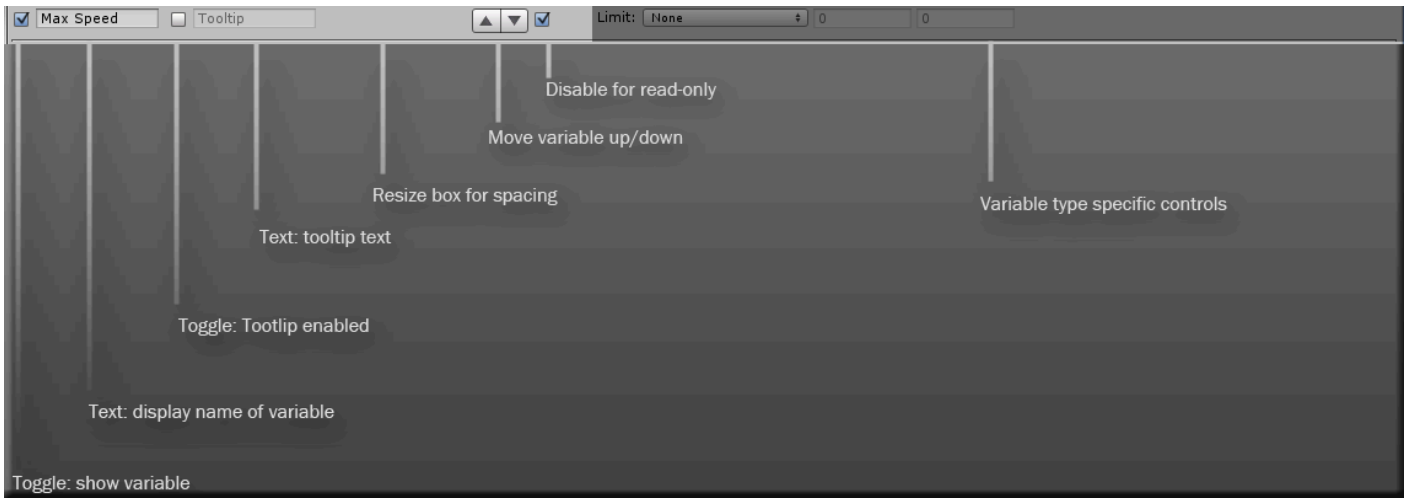


Select your scripts from your project window, and focus the Inspector++ window. All MonoBehaviours or ScriptableObjects you selected appear in the window. Click create and your selected scripts will have better inspectors instantly!

Editing Inspectors

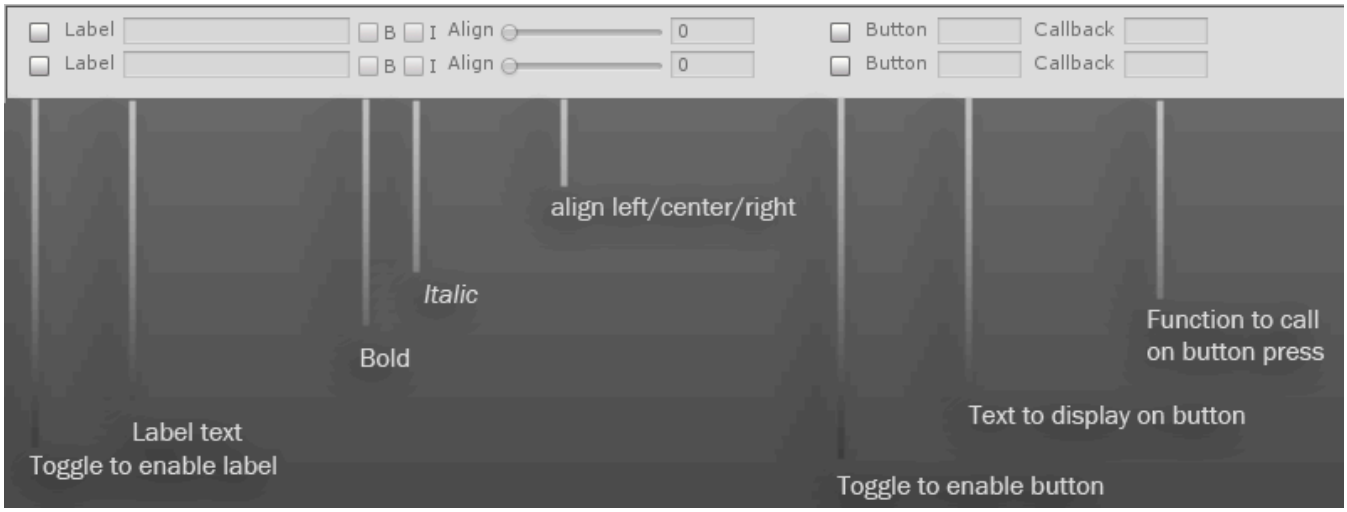
But we're control freaks (or maybe that's just me again). Inspector++ includes a window to visually edit your inspectors. Open it by clicking 'edit' next to the inspector you want to edit.

The window displays all variables, with some options. This picture explains it all in detail:



The variable type specific controls are too many to list here. They should be self explanatory, and if not, be sure to experiment!

The 'resize box for spacing' is actually a small box you can resize to put some space between vars. When the box is large enough you will notice some additional controls.



You can add a maximum of 16 buttons (4 per line) and 4 labels. When you have 2 or more buttons on a line, a toggle ‘condense’ appears. Check this to make the buttons clump together.

Tooltips
You can make tooltips via a toggle and textfield in the editor, but also through scripting, in 2 different ways:

1. a summary (C# only)
///<summary>
///this will be my tooltip!
///</summary>
public float tooltip;

2.Attribute:
[InspectorTip(“This will be my tooltip!”)] (or for JS @InspectorTip
public int myInt;

Save to file

You might have noticed you can not only edit an inspector but also 'Save to file'. This features produces a .cs file you can share in your projects or use in an asset store project. The .cs file is a optimized version of your custom inspector.
Note that when you edit your inspector, this file for distributing doesn't get updated (although your inspector still does). You need to regenerate after any changes.

To support Inspector++ a small subtle watermark will be added to the inspector. If you disagree with this you are free to open the generated file and remove the small watermark at the bottom

of the OnInspectorGUI function (more instructions there).

PlayMaker integration

InspectorPlus works for PlayMaker as well! In the bottom right you will find a button to open the PlayMaker editor window. Select the GameObjects with the FSM's you want an inspector for, and click create, that's it!

Notes:
1. Sometimes you will need to 'refresh' the inspector, when adding new variables or changing the names of existing ones.

2.InspectorPlus can only identify a FSM by it's name. Please ensure you have unique names, you can not have two different inspectors with the same name. In the following version this will hopefully be improved.

3. To use that inspector on more GameObjects: Find the script 'InspectorPlusPlayMaker + FsmName' in the 'InspectorPlusPlayMaker' folder. Place that script on GameObjects with an FSM that has the same variables.