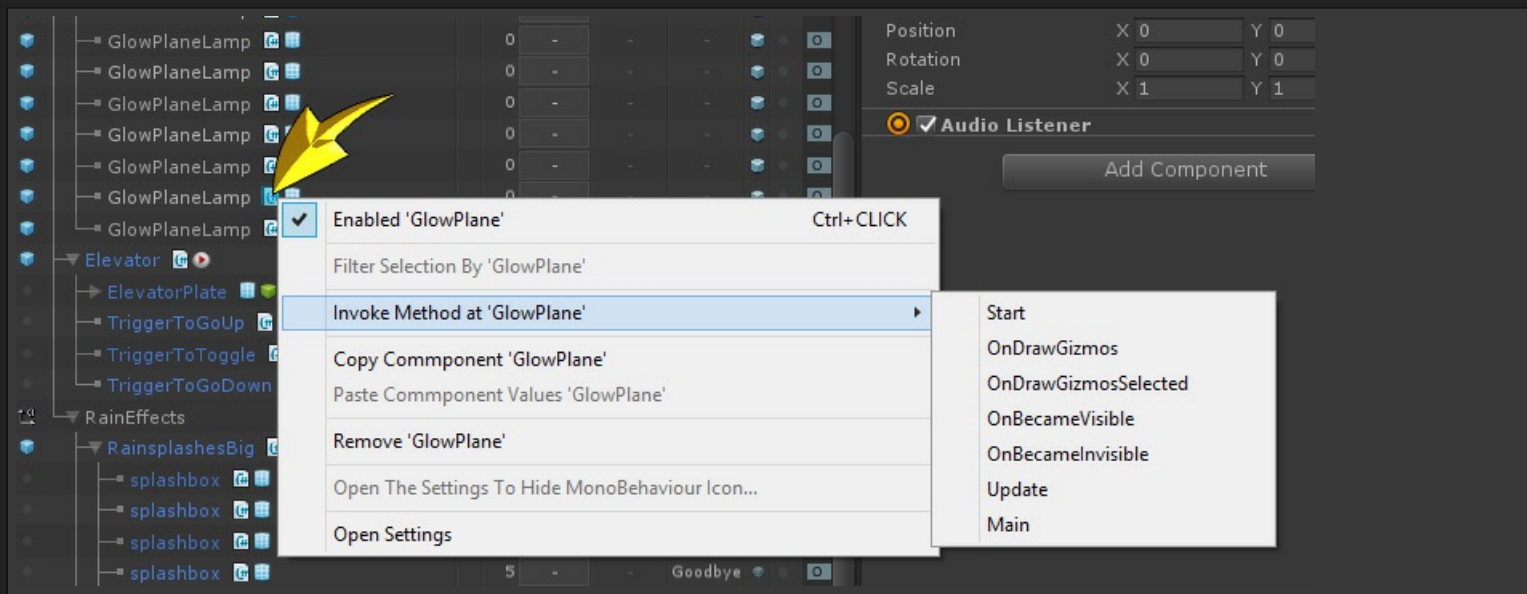


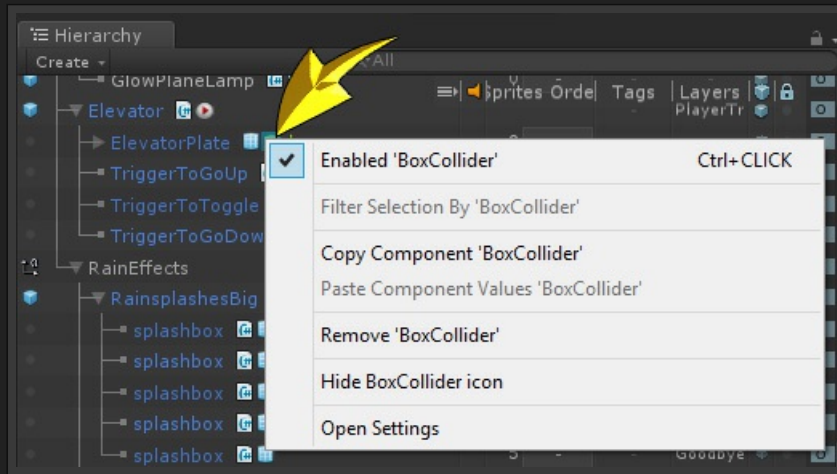
This module displays the components and scripts that have been assigned to the object. By default, icons are displayed next to the object names.

Left-click on the icon to open the component menu.



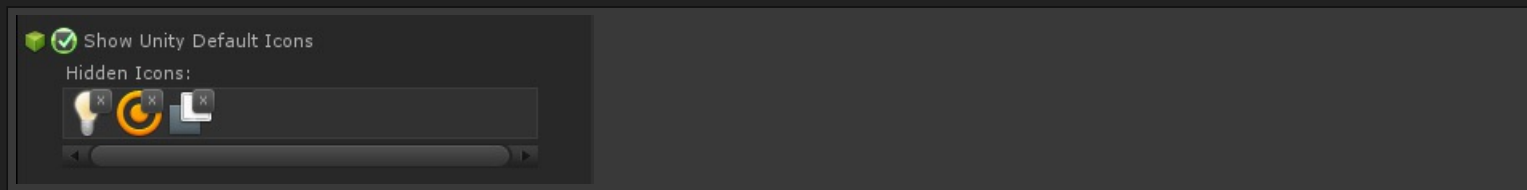
Enabled '...'	Displays and switches the Enable/Disable state of the component. You can hold down 'CTRL' on the keyboard and left-click on the icon
Filter Selection By '...'	Selects objects from selection that contain a certain component
Invoke Method at '...'	Calls any method that was defined in your script
Copy Component '...'	Copies the component to the clipboard (uses the standard Unity buffer and does not support the storage of multiple components at the same time)
Paste Component '...'	Paste the parameters of the previously copied component
Remove '...'	Removes a component
Open Settings	Opens settings

The menu is slightly different for standard Unity components and contains 'Hide BoxCollider icon.'

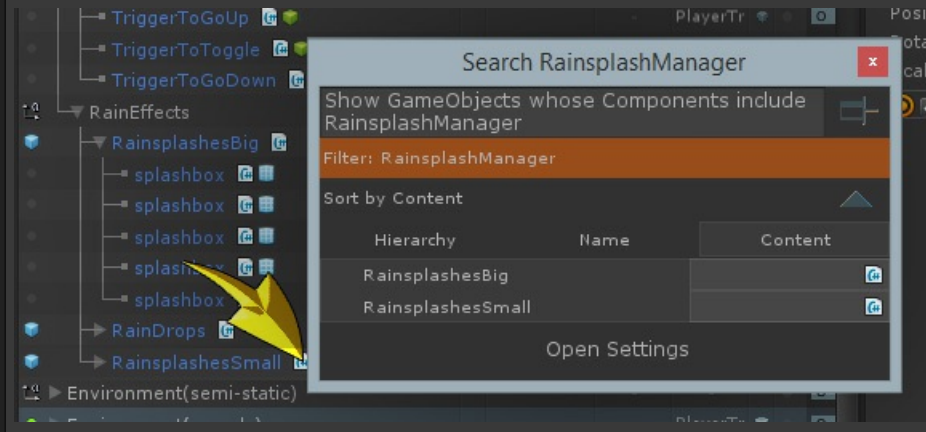


Hide '...' icon	Hides the icon from the component module. You can find the hidden icons in 'Open Settings' (see below)
-----------------	--

In settings, you will see the hidden icons.



To call the search box by parameters, right-click on the module name.



The search box will display to which the specified component is assigned (for more details, see 'Other Tips').

You can place the values of variables and functions directly in the hierarchy window next to the icon of the script. You can specify the width and color of the displayed value.

```
[SHOW_IN_HIER]
[SHOW_IN_HIER(width: float)]
[SHOW_IN_HIER(color: float[])]
[SHOW_IN_HIER(width: float, color: float[])]
```

Place the 'SHOW\_IN\_HIER' attribute in your code over a variable function or property.

```
// "Method's button" and custom button width
[SHOW_IN_HIER(width: 26)]
public float Method1()
{
    return 0;
}

// "Method's button" with arguments
[SHOW_IN_HIER]
public void Method2(string str)
{
    Debug.Log( "Method2(): " + string.IsNullOrEmpty(str));
}

// "Property Label" and custom color {r,g,b,a}
[SHOW_IN_HIER(color: new float [] { 1, 0, 0, 1 })]
GameObject Target {
    get { return null ; }
}

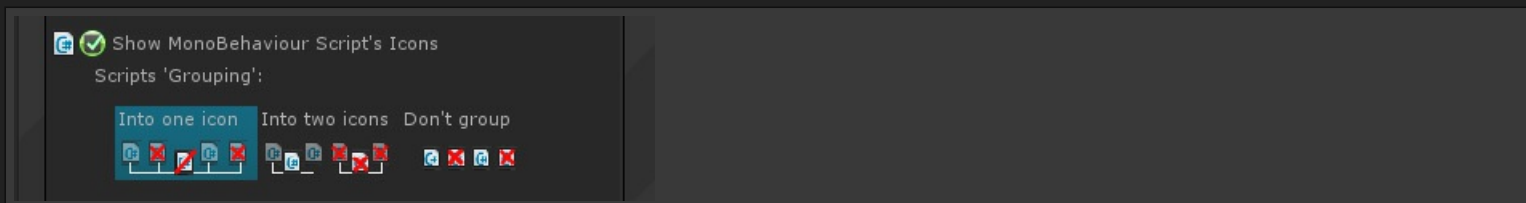
// "Field Label"
[SHOW_IN_HIER]
float speedField;
```

Functions can be called from the hierarchy window. They will be displayed to the right of the icon.

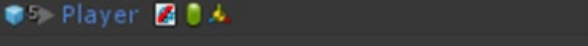
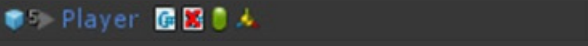
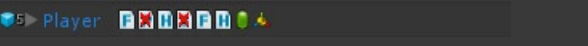


In settings, you can change the distance between the icons and indents.

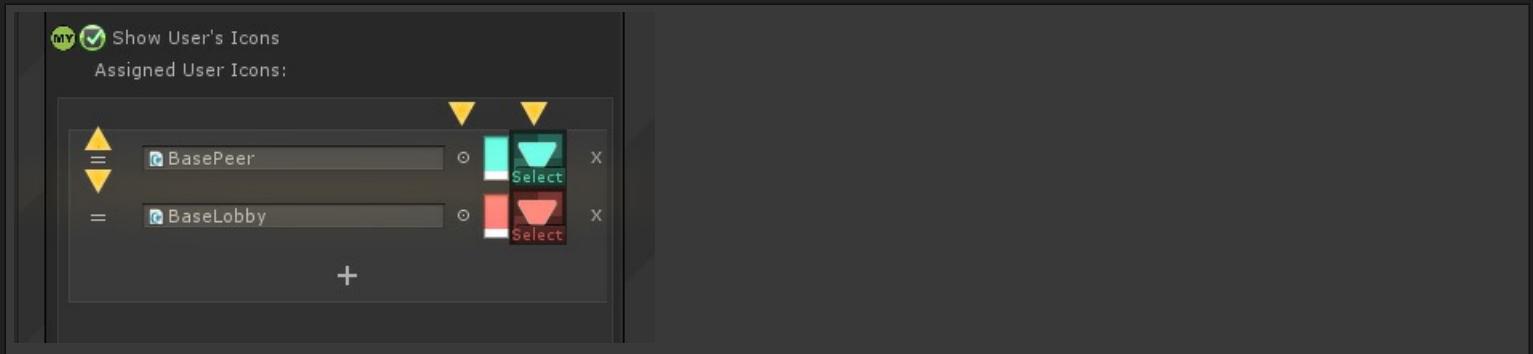
As well as configure the options for grouping MonoBehaviour scripts.



**Into one icon** Groups all scripts into one icon. If the group includes both a disabled and enabled script, then the icon will be displayed with one crossed line

	
Into two icons	<p>The enabled and disabled scripts will be divided into two groups</p> 
Don't group	<p>Scripts will not be grouped, and each script will be displayed with a separate icon with first letter of his name</p> 

You can add your own icons for the most needed components to make them easier to access. It also helps to track the necessary scripts in the hierarchy window.



You can change the display order, script, color, and icon. These settings apply only to the current project.

This module can be displayed as a separate column. The display switch next to the object name is in settings.