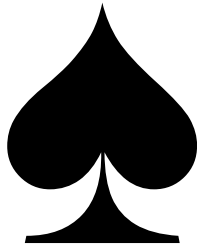


# Better UI

Makes uGui Better!

*version 1.1.2*



## TABLE OF CONTENTS

---

Better UI .....	1
Better UI Elements.....	4
1    Workflow: Make Better.....	4
2    UI and Layout Elements Overview .....	4
3    Better Transitions .....	7
3.1    How to add a Better Transition .....	7
3.2    Transition Modes.....	8
3.2.1    Color Tint .....	8
3.2.2    Sprite Swap .....	8
3.2.3    Animation .....	8
3.2.4    Object Activeness.....	8
3.2.5    Alpha .....	8
4    Screen Dependent Size Modifiers .....	9
4.1    Size Modification Settings.....	9
4.1.1    Pixel Height .....	10
4.1.2    Pixel Width.....	10
4.1.3    Aspect Ratio .....	10
4.1.4    Inverse Aspect Ratio .....	10
4.1.5    Dpi.....	11
4.2    Size Modifier Types.....	11
4.3    Components with Size Modifiers .....	11
5    Better UI Elements .....	13
5.1    Better Image .....	13
5.1.1    Color and Gradient.....	13
5.1.2    Materials.....	13
5.1.3    Border Resizing .....	14
5.2    Better Input Field .....	15
5.3    Better Scroll Rect .....	15
5.4    Better Text .....	16
5.4.1    Sizer Only .....	16
5.4.2    Stay in Bounds (default).....	16
5.4.3    Best Fit .....	17
5.5    Better Toggle .....	17
5.6    Better Grid Layout Group .....	17
6    Text Mesh Pro Support.....	17
6.1    Better Text Mesh Pro UGUI .....	18

6.1.1	Sizer Only .....	18
6.1.2	Stay in Bounds (default) .....	18
6.1.3	Best Fit .....	18
6.2	Better Text Mesh Pro - Input Field.....	18
6.3	Better Text Mesh Pro – Dropdown .....	19
	Resolution Monitoring .....	20
7	Resolution Monitor .....	20
8	Ingame Resolution Monitor .....	21
9	IResolutionDependency .....	21
	Materials.....	23
10	Materials Scriptable Object .....	23
10.1	Basics .....	23
10.2	Add your own Material.....	24
10.3	Animating Properties .....	24
	Editor Windows .....	25
11	Snap Anchors Window .....	25
11.1	Background.....	25
11.2	Workflow .....	25
11.3	Modes.....	26
11.3.1	Border .....	26
11.3.2	Point.....	26
11.3.3	Pivot.....	27
12	Pick Resolution Window .....	27
12.1	Settings .....	28
12.1.1	Resolution Filters .....	29
12.1.2	Options.....	29

# Better UI Elements

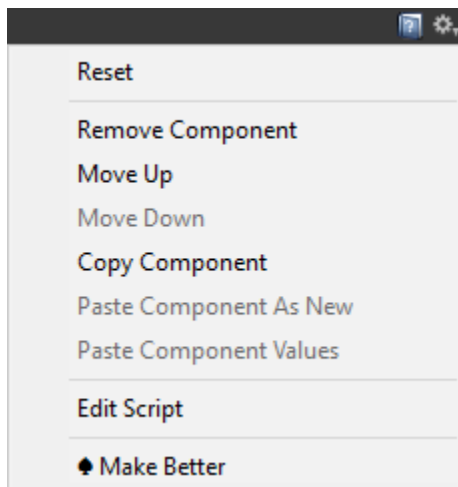
The Better UI Elements are extending the UI elements of Unity's uGUI framework by additional functionality.

## 1 WORKFLOW: MAKE BETTER

---

Better UI is designed to extend uGUI where needed. It is also possible to include it in existing projects and extend certain UI elements with a simple workflow.

Wherever you realize needing some more functionality on an UI element just click the options-gear-wheel on the component and click the "♠ Make Better" option.



This will replace the current UI Element by its "Better" pendant. It will keep all the values of the original uGUI objects properties. It will also search for references to that object in the hierarchy. Any found references to that element will be replaced with references to the "Better" element, so you don't have to worry about manual changes in the scene after making an element "better".





















Please note that only references in the scene will be replaced. If you make a prefab "better" which is referenced somewhere, the reference will get lost and you have to re-assign it manually.













## 2 UI AND LAYOUT ELEMENTS OVERVIEW

---

The following table is showing a short overview about how a "Better UI" element extends the uGUI element.

A detailed description is documented later.

uGui Icon	uGUI Element	Better UI Icon	Better UI Element	What is better
	Selectable		Better Selectable	<ul style="list-style-type: none"> <li>Better Transitions</li> </ul>
	Button		Better Button	<ul style="list-style-type: none"> <li>Better Transitions</li> </ul>
	Dropdown		Better Dropdown	<ul style="list-style-type: none"> <li>Better Transitions</li> </ul>
	Image		Better Image	<ul style="list-style-type: none"> <li>Sliced Mode: <ul style="list-style-type: none"> <li>Keep Border Aspect Ratio</li> <li>Border Scaling</li> </ul> </li> <li>Tiled Mode: <ul style="list-style-type: none"> <li>Border Scaling</li> </ul> </li> <li>Gradients instead of Color</li> <li>Improved Material System <ul style="list-style-type: none"> <li>Predefined Fast Materials (Grayscale, Hue-Saturation-Brightness)</li> <li>Clone custom Materials</li> </ul> </li> </ul>
	Input Field		Better Input Field	<ul style="list-style-type: none"> <li>Better Transitions</li> <li>Additional Placeholders</li> </ul>
	Scrollbar		Better Scrollbar	<ul style="list-style-type: none"> <li>Better Transitions</li> </ul>
	Scroll Rect		Better Scroll Rect	<ul style="list-style-type: none"> <li>Horizontal- / Vertical start position</li> </ul>
	Slider		Better Slider	<ul style="list-style-type: none"> <li>Better Transitions</li> </ul>
	Text		Better Text	<ul style="list-style-type: none"> <li>Font Sizer (Screen Dependent Size Modifier)</li> <li>Fitting Modes</li> </ul>
	Toggle		Better Toggle	<ul style="list-style-type: none"> <li>Better Transitions</li> <li>Additional Toggle Transitions (On / Off)</li> </ul>

	<b>Grid Layout Group</b>		Better Grid Layout Group	<ul style="list-style-type: none"> <li>Resolution Dependent Size Modifiers: <ul style="list-style-type: none"> <li>Padding Sizer</li> <li>Spacing Sizer</li> <li>Cell Sizer</li> </ul> </li> <li>Constraint: Fixed Column- / Row Count <ul style="list-style-type: none"> <li>Fit (Cell size adjustment to fit in layout group)</li> </ul> </li> </ul>
	<b>Horizontal Layout Group</b>		Better Horizontal Layout Group	<ul style="list-style-type: none"> <li>Screen Dependent Size Modifiers: <ul style="list-style-type: none"> <li>Padding Sizer</li> <li>Spacing Sizer</li> </ul> </li> </ul>
	<b>Vertical Layout Group</b>		Better Vertical Layout Group	<ul style="list-style-type: none"> <li>Screen Dependent Size Modifiers: <ul style="list-style-type: none"> <li>Padding Sizer</li> <li>Spacing Sizer</li> </ul> </li> </ul>
			Transform Scaler	<ul style="list-style-type: none"> <li>Screen Dependent Size Modifiers: <ul style="list-style-type: none"> <li>Scale Sizer</li> </ul> </li> </ul>
	<b>Text Mesh Pro UGUI</b>		Better Text Mesh Pro UGUI	<ul style="list-style-type: none"> <li>Screen Dependent Size Modifiers: <ul style="list-style-type: none"> <li>Font Sizer</li> <li>Min / Max Font Sizer</li> <li>Margin Sizer</li> </ul> </li> <li>Fitting Modes</li> </ul>
	<b>TMP_Input Field</b>		Better Text Mesh Pro Input Field	<ul style="list-style-type: none"> <li>Screen Dependent Size Modifiers: <ul style="list-style-type: none"> <li>Point Sizer (if enabled)</li> </ul> </li> <li>Better Transitions</li> </ul>
	<b>TMP_Dropdown</b>		Better Text Mesh Pro Dropdown	<ul style="list-style-type: none"> <li>Better Transitions</li> <li>Show / Hide Transitions</li> </ul>

## 3 BETTER TRANSITIONS

---

Transitions are used to give feedback for the different states of an object. A selectable UI element can have the following states:

- Normal  
*When it is just displayed on the screen without having any interaction right now*
- Highlighted  
*When the object is in Focus (Mouse is above or the last element where a pointer down action has been performed)*
- Pressed  
*When the pointer is currently pressing the element*
- Disabled  
*When the element is not interactable*

UI Element with "Better Transitions" can have a uGUI transitions as well as additional transitions. These additional transitions can affect any number of objects with different kinds of transitions. Also, there are more options for "Better Transitions" than for the uGUI transition.

*Example: If you need a button where the background color should change as well as the text color you could add a "Better Transition" for the Text object.*

The Following Better UI Elements providing Better Transitions:

- Better Selectable
- Better Button
- Better Dropdown
- Better Input Field
- Better Scrollbar
- Better Slider
- Better Toggle
- Better Text Mesh Pro Dropdown
- Better Text Mesh Pro Input Field

### 3.1 HOW TO ADD A BETTER TRANSITION

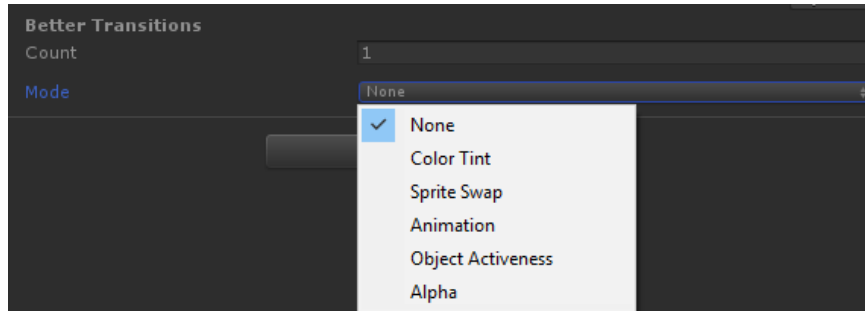
To add a "Better Transition" to one of the UI Elements perform the following steps:

1. Increase the Count of Better Transitions by one
2. Select the Mode of the Transition (see [description for the modes below](#))
3. Drag and Drop a valid Target in the respective Object Field
4. Define the states for every transition

## 3.2 TRANSITION MODES

The following modes can be selected.

It is possible to affect the same object with several transitions. You should make sure not to use the same transition mode for the same target more than once.



### 3.2.1 Color Tint

Colorizes the Target Graphic for each state. (Like the uGUI Color Tint transition)  
The Fade Duration provides information about if and how long a state transition is animated.

You should **make sure that the Target Graphics color is white** unless you want to have mixed up colors.

### 3.2.2 Sprite Swap

Changes the Sprite of the Target Image for each state. (Like the uGUI Sprite Swap transition)

### 3.2.3 Animation

Triggers the given Trigger Names on Target Animator for each state. (Like the uGUI Animation transition)

### 3.2.4 Object Activeness

Can set target GameObjects active or inactive for each state. This cannot be previewed in edit mode.

Make sure the UI Element itself doesn't become inactive through this transition because it cannot enable itself again once it is inactive (it doesn't do anything anymore when inactive).

### 3.2.5 Alpha

Changes the transparency of Target Graphic for each state. (Like Color Tint transition)  
The Fade Duration provides information about if and how long a state transition is animated.

### 3.2.6 Material Property

Changes the material property of a material assigned to Target Better Image. This only works for properties which are mapped to some vertex information (see [Materials](#)).

Choose which property you want to affect and the fade duration which specifies how long the transition takes. If there is no property you cannot select one and nothing happens on transition.



## 4 SCREEN DEPENDENT SIZE MODIFIERS

There are various "Sizer" objects on certain UI elements. They are used to scale certain values depending on the used resolution (and / or DPI).

Every "Sizer" contains the following values:

- **Current Value:** The last calculated value for the current resolution / dpi. Cannot be changed directly because it is calculated.
- **Optimized Size:** The fixed size for the optimized resolution / dpi
- **Size Modification** (Foldout)
  - **<X> x <Y> @ <Z> DPI** [change]: The optimized resolution / dpi for this unity project. by pressing the change button the Resolution Monitor object will be selected where the global Optimized resolution / dpi can be changed (see [there](#)).
  - **Min Size:** The minimum size (if the calculation for a resolution / dpi would return a lower value, the Min Size is used instead)
  - **Max Size:** The maximum size (if the calculation for a resolution / dpi would return a higher value, the Max Size is used instead)
  - **Size Modification** (List): Contains Setting entries of how the size should be calculated. If the list is empty, the size will always be the optimized size. If there is more than one entry, the middle is calculated.



### 4.1 SIZE MODIFICATION SETTINGS

Each Modification Setting consists of two values: Mode and Impact.

The Mode defines which Screen-values should be used to calculate the actual size (in the very most cases "Pixel Height" or "Pixel Width")

The Impact defines how much the modification settings affects the change of the size (doesn't affect at zero; affects totally at one).

Depending on the current screen resolution / dpi, the value is recalculated using the size modification settings.

Here are the different Modification Modes in detail.

#### 4.1.1 Pixel Height

If the height of the current resolution is smaller than the optimized resolutions height, the size will get smaller.

If the height of the current resolution is greater than the optimized resolutions height, the size will get bigger.

#### 4.1.2 Pixel Width

If the width of the current resolution is smaller than the optimized resolutions width, the size will get smaller.

If the width of the current resolution is greater than the optimized resolutions width, the size will get bigger.

#### 4.1.3 Aspect Ratio

The aspect ratio is calculated the following way:  $\text{aspectRatio} = \text{width} / \text{height}$

If the aspect ratio of the current resolution is smaller than the optimized resolutions aspect ratio, the size will get smaller.

If the aspect ratio of the current resolution is greater than the optimized resolutions aspect ratio, the size will get bigger.

#### 4.1.4 Inverse Aspect Ratio

The inverse aspect ratio is calculated the following way:  $\text{aspectRatio} = \text{height} / \text{width}$

If the inverse aspect ratio of the current resolution is smaller than the optimized resolutions inverse aspect ratio, the size will get smaller.

If the inverse aspect ratio of the current resolution is greater than the optimized resolutions inverse aspect ratio, the size will get bigger.

#### 4.1.5 Dpi

DPI means "Dots Per Inch" or in other words "how many pixels can fit next to each other within one inch on the current screen".

If you choose to scale in DPI mode, the physical size will always stay the same.

*Example: If you have an object which displays a ruler and you adjust the correct size for the optimized resolution / dpi and you resize that object by using a sizer which modifies the size by dpi, the ruler will have the correct physical size on every device (if the device provides the correct dpi for its screen).*

## 4.2 SIZE MODIFIER TYPES

There are Size Modifiers for the following types:

- Float
- Vector2
- Vector3
- Margin (Wrapper for RectOffset – used for Padding of Layout Groups)

For each dimension of a type (x, y, z or left, right, top, bottom) there are modification lists. If you want to adjust the size uniformly make sure to set the same mode to all dimensions (e.g. set all to "Pixel Height").

## 4.3 COMPONENTS WITH SIZE MODIFIERS

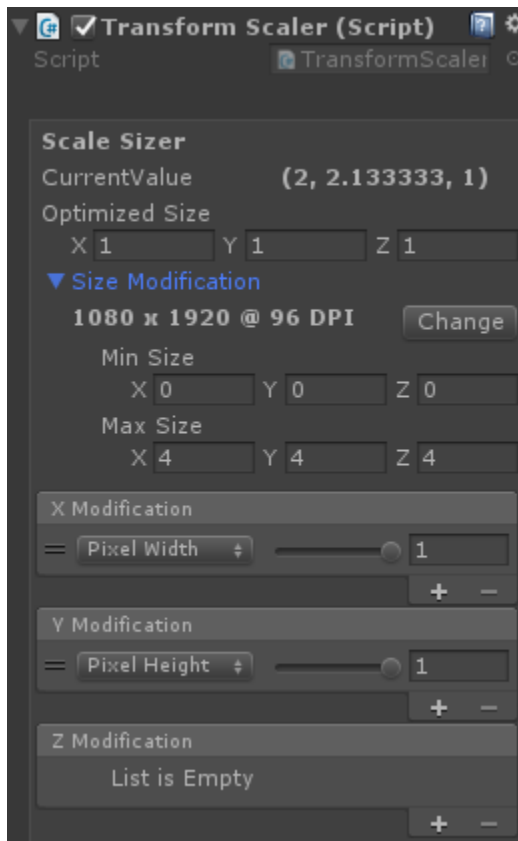
There are some Better UI elements using the Size Modifiers:

- Better Image
  - Sprite Border Scale (Sliced & Tiled mode)
- Better Text
  - Font Sizer
- Better Grid Layout Group
  - Padding Sizer
  - Spacing Sizer
  - Cell Sizer
- Better Horizontal- & Better Vertical Layout Group
  - Padding Sizer
  - Spacing Sizer
- Better Text Mesh Pro UGUI
  - Font Sizer
  - Min / Max Font Sizer (depending on fitting mode)
  - Margin Sizer

- Better Text Mesh Pro Input Field
  - Point Size Scaler (If “Override Point Size” is enabled)

Additionally, there is an extra Component using a size modifier:

- Transform Scaler
  - Scale Sizer



The Transform Scaler affecting the scale of the object. You should use it wisely. Scaling inside the UI often leads to unexpected positions and sizes of objects.

As a rule of thumb, you should use the Transform Scaler only for objects which have the min and max anchor positions on the same spot and which has no children or no complex children hierarchy.

## 5 BETTER UI ELEMENTS

In addition to the "Better Transitions" and Size Modifiers some elements also have custom behavior.

### 5.1 BETTER IMAGE

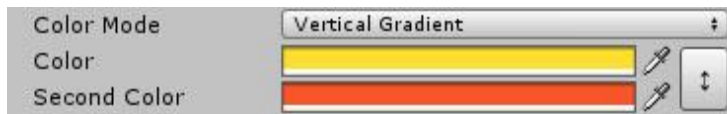
#### 5.1.1 Color and Gradient

**Color Mode:** Can switch Between Color (default behavior), Horizontal Gradient and Vertical Gradient. A Gradient in this case consists of two colors which defines the start and end color. The colors in between are interpolated.

*Gradients are not supported for Images with Filled mode.*

**Color:** The Color which is used to colorize the image. If a gradient mode is selected, it is the color on the left or top of the gradient.

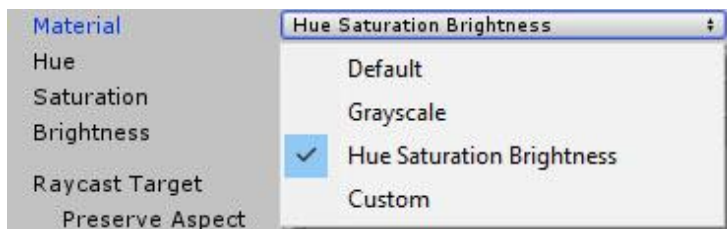
**Second Color:** The Second Color is only available when a Gradient is selected. It defines the color on the right or bottom of the gradient.



The button on the right-hand side in gradient mode allows to swap the colors to mirror the gradient.

#### 5.1.2 Materials

**Material:** This is a Dropdown in Better UI where you can select any predefined material (see [Materials Scriptable Object](#)), the default material or a Custom Material.



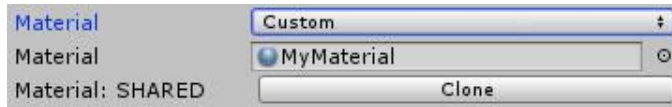
**Standard Material:** It is like the Default Material but can have assigned an effect.

**Grayscale Material:** Applies a simple shader which displays the colors of the sprites as grayscale values.

**Hue Saturation Brightness Material:** Allows to shift the colors of the sprites (hue), make them intense or gray (saturation) and darker or brighter (brightness). This is a very powerful shader which is shared between all Images using it. However, the values can be different on each instance. This is possible because of a trick and is much better for performance if you have a lot of images with this shader (See [Materials Scriptable Object](#)).

*Hue Saturation Brightness Material is not supported for Images with Filled mode.*

**Custom Material:** Assign any material you want. After assigning there will appear a “Clone” Button. You want to clone a material if you want to change properties of it only for that image and the original material should keep the original values.



#### 5.1.2.1 Material Effects

For the Better UI predefined Materials (Standard, Grayscale and Hue Saturation Brightness) there is an option to define the Effect of the image. This enables the image to be mixed with the background in some way.

**Normal:** The image is not mixed with the background in any special way. It is just put on top of it.

**Additive:** The colors of the image are combined with the colors of the background so that the colors are mixed and saturated. The brighter the color of the image and / or background the more intense the color appears.

**Linear Dodge:** like additive but the colors are even more intense. The background color is not so important for the intensity but the images colors is.

**Multiply:** The colors are mixed with the background, while darker colors are dominating the brighter colors. Because alpha becomes opaque black by this effect the transparent pixels are clipped which leads to hard pixel lines. This may not be a non-acceptable look so you should consider wisely if you want to use the effect at all.

#### 5.1.3 Border Resizing

Border Resizing options are only available for Sliced or Tiled Images

##### 5.1.3.1 Sliced

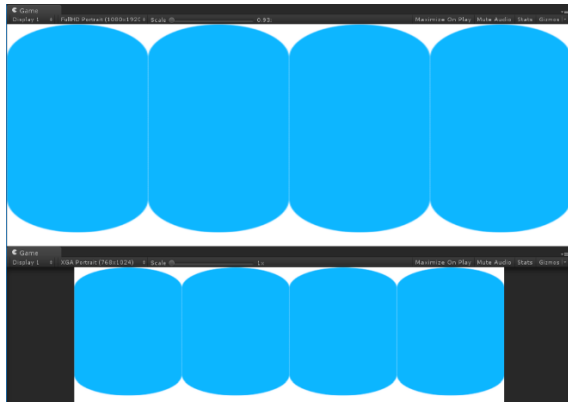
**Keep Border Aspect:** Scales the borders of the image to preserve their aspect ratio if necessary.



**Sprite Border Scale:** Scales the borders of the image using the settings inside the sizer. Also scales the aspect ratio for the "Keep Border Aspect" option.

### 5.1.3.2 Tiled

**Sprite Border Scale:** Scales the size and border of the tile-pattern. If set up correctly the same amount of tiles are shown at any resolution.



## 5.2 BETTER INPUT FIELD

In addition to the Better Transitions the Better Input Field has a list of **Additional Placeholders**.

These Placeholders are Graphic objects (Text, Image or Raw Image) which are visible (enabled) when there is no text entered in the input field and invisible (disabled) as soon as the user starts to write.

*Example: If you need e.g. an image as a placeholder as well as a text, add them to the Additional Placeholders list.*

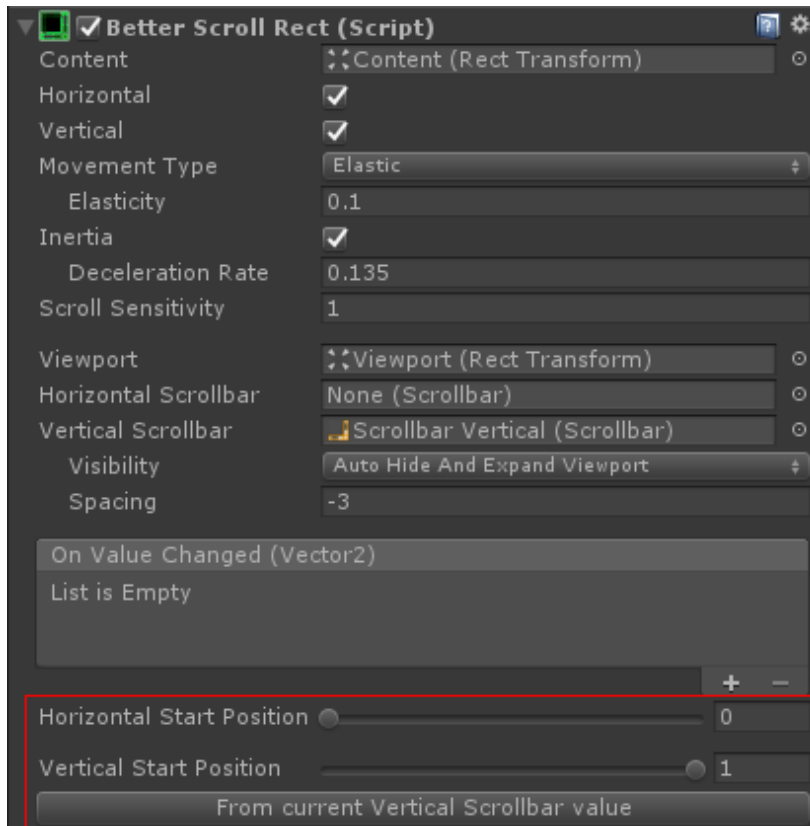
## 5.3 BETTER SCROLL RECT

uGUIs Scroll Rect has the odd behavior to always jump somewhere in the middle of the vertical scroll area on start.

The Better Scroll Rect provides two extra fields: **Horizontal Start Position** and **Vertical Start Position**.

The Scrollbar will jump to that (relative) position on start.

If you have horizontal / vertical Scrollbars assigned, you can inject the current value of them by a click on the button below the respective start position slider.



In the background a Coroutine is started which waits a frame before changing the positions. Therefore, it is not possible in this version of Better UI to have the position change in edit mode.

## 5.4 BETTER TEXT

Better Text can Scale the Font Size according to the current resolution / dpi with the Font Sizer.

It also has three kinds of Fitting Modes:

### 5.4.1 Sizer Only

Will just affect the font size of the text using the Font Sizer. If the size becomes too large, parts of the text may be cut off or overflow the bounds (depending on the overflow settings).

### 5.4.2 Stay in Bounds (default)

The size is affected by the Font Sizer. But if the text wouldn't fit inside its bounds anymore the size is reduced to fit in.



### 5.4.3 Best Fit

Equal to the "Best Fit" option of uGUI. The text will grow or shrink to fit inside the bounds. The Min Size and Max Size values of the Font Sizer are used to avoid too small or too large font sizes.

## 5.5 BETTER TOGGLE

In addition to the "Better Transitions" there are "**Better Toggle Transitions (On / Off)**".

They function the same way as "Better Transitions" but have the states "**On**" and "**Off**" instead of "Normal", "Highlighted", "Pressed" and "Disabled". Accordingly, they are triggered when the "Is On" state of the toggle changed.

It might lead to unexpected behavior if you assign the same kind of transition with the same target to "Better Transitions" and "Better Toggle Transitions". However, the Better Toggle logic tries to behave as you would expect in such a case.

## 5.6 BETTER GRID LAYOUT GROUP

Besides the Size Modifiers for Padding, Spacing and Cell Size, there is an extra option if you select "Fixed Column Count" or "Fixed Row Count" as a constraint: **Fit**

If "Fit" is enabled, the cell height (fixed row count) or cell width (fixed column count) are stretched or shrunk to exactly fit all together inside the parent while respecting the size of the padding and spacing.

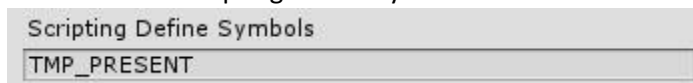
# 6 TEXT MESH PRO SUPPORT

---

If you are using Text Mesh Pro you can activate the Text Mesh Pro support of Better UI by adding a scripting define symbol.

To do so, follow these steps:

1. Navigate to the Player Settings (Edit -> Project Settings -> Player)
2. Unfold the "Other Settings" section
3. In the "Configuration" section Insert "TMP\_PRESENT" (all capitals, without quotes) in the field with the title "Scripting Define Symbols"



4. Repeat 2. and 3. for each Platform you want to deploy for.

Then you can convert the Text Mesh Pro components to its "better" pendant.

## 6.1 BETTER TEXT MESH PRO UGUI

Better Text Mesh Pro UGUI also has a Font Sizer to adjust the font size according to the current resolution.

Text Mesh Pro UGUI has a margin for the text. Of course, this margin will be represented by a sizer object on the better pendant (to scale the margin with the resolution).

Like Better Text it provides the Fitting Modes “Sizer Only”, “Stay In Bounds” (default) and “Best Fit”. The logic behind works a bit different here.

### 6.1.1 Sizer Only

The Font Sizer is responsible for the Font size. It scales with the resolution but doesn’t do any fitting stuff.

### 6.1.2 Stay in Bounds (default)

The size is affected by the Font Sizer. But if the text wouldn't fit inside its bounds anymore the size is reduced to fit in.

Internally “Auto Size” of Text Mesh Pro UGUI is enabled. The “Min” value can be adjusted with a sizer. The “Max” value is the value of the Font Sizer.

### 6.1.3 Best Fit

Equal to the "Auto Size" option of Text Mesh Pro UGUI. The text will grow or shrink to fit inside the bounds. The Min Size and Max Size values can be adjusted with the corresponding Sizer objects which appear in this mode.

## 6.2 BETTER TEXT MESH PRO - INPUT FIELD

Better Transitions and Additional Placeholders work like Better Input Field (see [there](#)).

Additionally, there is a checkbox “Override Point Size”. If checked a Sizer object “Point Size Scaler” appears. This will override the “Point Size” field provided by Text Mesh Pro, while it scales with the resolution. This sets the font size of the Text and Placeholder(s). If you use “Better Text Mesh Pro UGUI” inside the input field, you need to use “Override Point Size” because the default Point Size logic may lead to unexpected behavior here.

### 6.3 BETTER TEXT MESH PRO – DROPDOWN

In addition to the “Better Transitions” there are “Show / Hide Transitions”. These Transitions are triggered when the option list of the dropdown is opened (Show) or closed (Hide). The same transition types as in “Better Transitions” are available.

As with the “Toggle Transitions” also the “Show / Hide Transitions” may behave unexpected when the same target and transition type is selected for a normal or “Better” Transition as well as for a “Show / Hide Transition”.

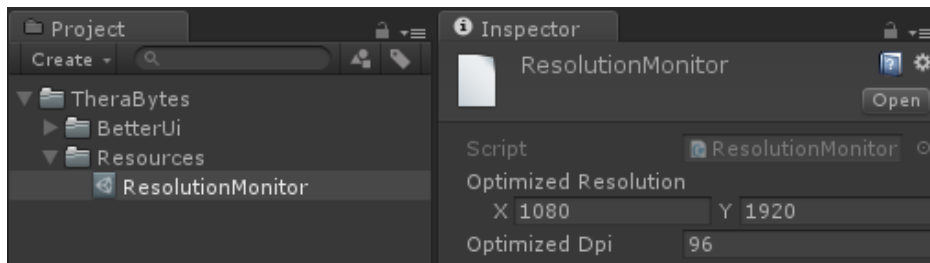
# Resolution Monitoring

## 7 RESOLUTION MONITOR

The Resolution Monitor is a scriptable object which holds the optimized resolution / dpi and which tracks changes of the resolution and notifies certain scene objects about a change.

The Resolution Monitor object can be found in your Unity project at "TheraBytes/Resources/ResolutionMonitor". There you can set the optimized Resolution / DPI for your project.

If there is no ResolutionMonitor object for some reason, you can create it manually by selecting "Window > Thera Bytes > Force ScriptableObjectSingleton Initialization" from the menu bar of Unity.



The following static properties can be accessed:

- ResolutionMonitor.**OptimizedResolution**: A Vector2 indicating the resolution which is used to build the UI.
- ResolutionMonitor.**OptimizedDpi**: A float value indicating the optimized DPI you are working on.
- ResolutionMonitor.**CurrentResolution**: The current Resolution (or the resolution which has been detected the last time)  
In the Editor it is the selected resolution of the main game view window. This might be different to Screen.width and Screen.height.
- ResolutionMonitor.**CurrentDpi**: The current DPI (only tracked for changes inside the editor)  
The scale of the main game view window is affecting the DPI inside the Editor. Scaling of the game view can be used to simulate different DPI values.

The Resolution Monitor automatically keeps track of resolution changes inside the Editor. All MonoBehaviour's of the scene which implement the interface IResolutionDependency will be automatically notified about the Resolution Change.

If you want to have the same behavior in the final build, you need to create an Ingame Resolution Monitor.

## 8 INGAME RESOLUTION MONITOR

---

The Ingame Resolution Monitor is a simple MonoBehaviour which triggers the logic of the Resolution Monitor in-game. If the user can change the Resolution inside of your game, you should use it.

To do so, call the following code once at start of your game:

```
IngameResolutionMonitor.Create();
```

Once created the Ingame Resolution Monitor will not be destroyed anymore. So make sure to create only one at the beginning and not more.

Note that the ResolutionMonitor.CurrentResolution and ResolutionMonitor.CurrentDpi are equal to Screen.width / Screen.height and Screen.dpi during runtime.

## 9 IRESOLUTIONDEPENDENCY

---

If you want to add your own script which should update itself whenever the resolution changes, implement IResolutionDependency. You will then need to implement a method called "OnResolutionChanged()". Trigger your update stuff there. Note that you probably also need to perform the same logic in your OnEnable() method to make sure it is also calculated when the scene is opened and the resolution has not changed.

The Resolution Monitor will update your MonoBehaviour then automatically when the resolution changed.

The following code is an Example how the IResolutionDependency can be used (this example logs whether the screen is in Portrait or Landscape mode):

```
using TheraBytes.BetterUi;
using UnityEngine;

public class OrientationTracker : MonoBehaviour, IResolutionDependency
{
    // Called by Unity
    private void OnEnable()
    {
        // Check the resolution when the object becomes active
        LogOrientation();
    }

    // Implementation of IResolutionDependency
    public void OnResolutionChanged()
    {
        // Check the resolution after it changed
        LogOrientation();
    }

    // Logs the orientation of the current resolution (Portrait / Landscape)
    private void LogOrientation()
    {
        Vector2 resolution = ResolutionMonitor.CurrentResolution;
        if (resolution.x < resolution.y)
        {
            Debug.Log("Portrait");
        }
        else
        {
            Debug.Log("Landscape");
        }
    }
}
```

# Materials

## 10 MATERIALS SCRIPTABLE OBJECT

For Better Images, there are some pre-defined materials which can be selected from a drop down (see [there](#)).

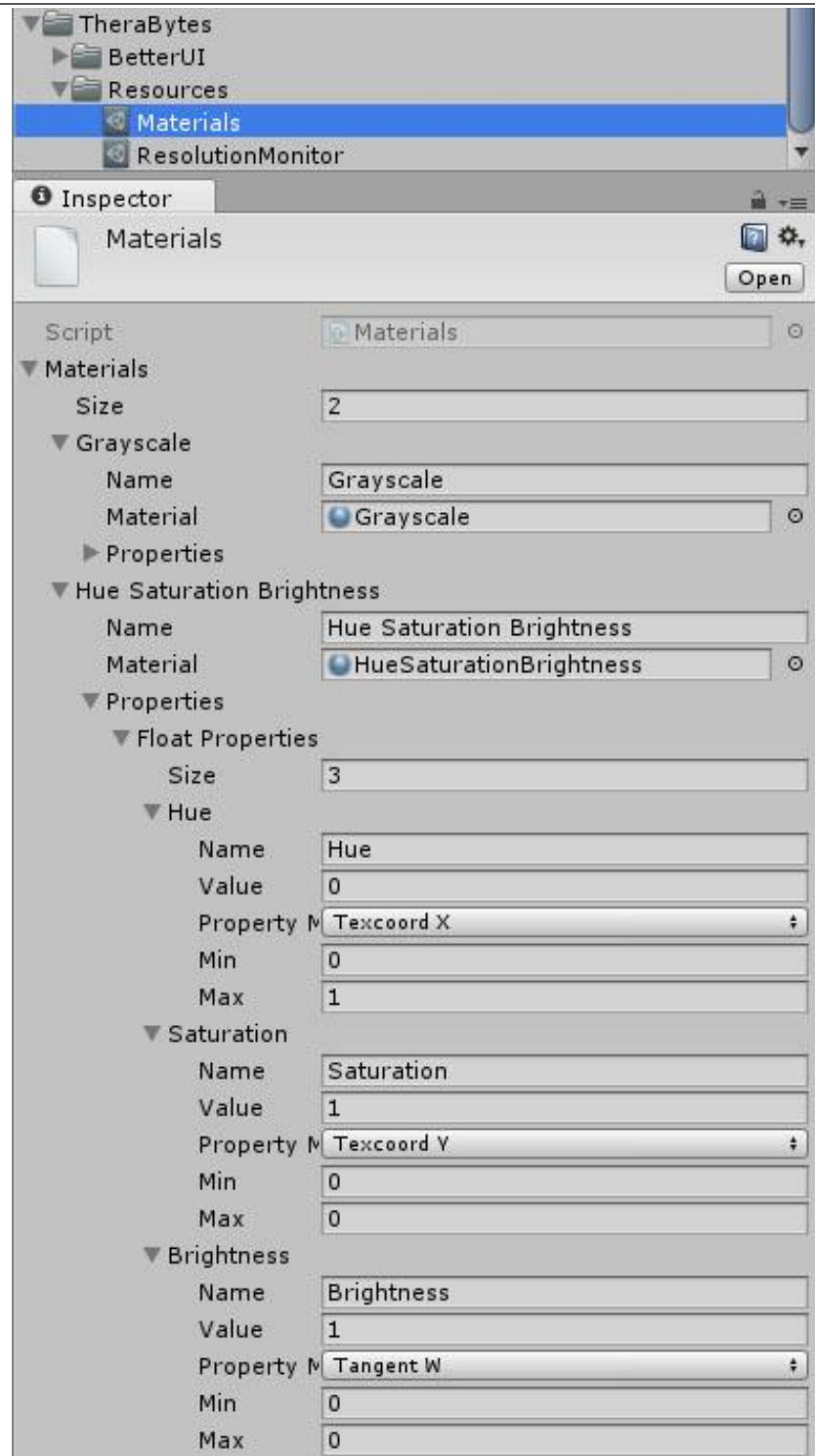
The Materials listed in that dropdown are defined in the Scriptable Object “Materials” at “TheraBytes -> Resources” (if not present click “Window -> Thera Bytes -> Force ScriptableObjectSingleton Initialization”).

### 10.1 BASICS

It is reserved for materials without any properties and for materials which use the float2 TEXCOORD1 and the w-value of the float4 TANGENT semantics.

By using these Vertex-variables for the properties of your shader, it is possible to share one material but having different properties on each Image. This allows not only a faster workflow, it also increases performance if you have many Images using one Material (instead of using cloned materials).

**The Materials List is not for materials with normal properties.**



## 10.2 ADD YOUR OWN MATERIAL

You can add your own shaders to the Materials list to easily access it from any better image. Here is how to do it:

1. Increase the **Size** of Materials by one.
2. Expand the last material definition
3. Give it a Name in the **Name** field (This name will be displayed in the Image's dropdown)
4. Drag and Drop your Material in the **Material** field
5. Expand **Properties** and **Float Properties**
6. Insert the number of smart Properties in the **Size** field (maximum: 3)
7. For each Property Field do the following:
  - a. Enter the Name of the Property in the **Name** field (Label for the Property in the Image Inspector)
  - b. Enter the default value in the **Value** field
  - c. Select the Vertex variable where this property maps to in the **Property Map** dropdown
  - d. If you want to display a slider in the Image Inspector for the property, you can define the **Min** and **Max** values in the respective fields. If Max is smaller or equal Min, any number can be chosen without a slider.

## 10.3 ANIMATING PROPERTIES

To Animate a property there are three variables available inside Better Image: materialProperty1, materialProperty2 and materialProperty3 connected to the first, second and third defined Property of your material.

These properties are mapped to the properties of the respective material. You don't have to care about them too much because the properties can be modified in the inspector while animating where their property name is displayed.



# Editor Windows

## 11 SNAP ANCHORS WINDOW

---

The Snap Anchors Window can be opened from the menu bar of Unity under "Window > Thera Bytes > Snap Anchors".

### 11.1 BACKGROUND

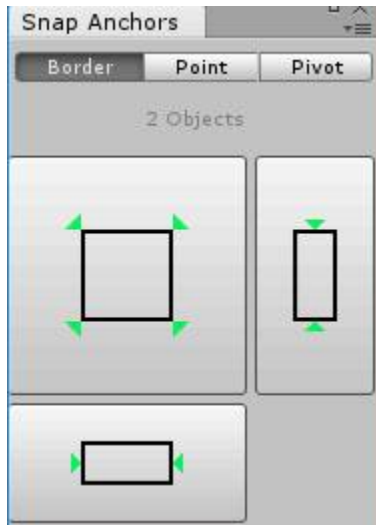
To really work resolution independently you should never use constant pixel sizes anywhere. Instead all measurements should be done with "percentages of the screen". To achieve this you should use Anchors instead of position and size values in pixels. Anchors describe the top, left, bottom and right anchor of an object in relative coordinates (relative to it's parent). uGUI comes with an anchoring system which allows exactly this. However, the predefined Anchoring settings often doesn't fit the requirements, so you must setup the anchors by hand which can be really annoying. The Snap Anchors window allows you to adjust the anchors in the most used way by just clicking a button.

### 11.2 WORKFLOW

Add an UI element to a canvas in while working with the optimized resolution. Move and resize the object as you want (but avoid it to scale or rotate it). When you are ready click the big button of the Snap Anchors Window which will change the anchors of the object but keeps its position and size.

Change the resolution to see how the objects position and size is affected.

You can also select several Objects and apply the anchoring to all of them. Note that only anchors of selected objects will be adjusted. Their children keep their previous anchoring as long as they are not selected as well.



It is also possible to snap only the horizontal Anchors (button on bottom) or the vertical anchors (button on left) and keep the other axis as it were before.

Keep in mind that the anchors are always relative to their parents. If the parent doesn't change its size, the anchors will be irrelevant. So, make sure that the parents are anchored correctly as well.

For some objects anchoring doesn't work as expected. This is the case for objects where the anchors are changed inside the code. Such an object is the Handle of a Slider for example.

## 11.3 MODES

There are three modes how to snap the anchors which can be selected at the top of the window (Border, Point, Pivot).

The Border mode is the most important and probably used more than 90% in comparison to the others.

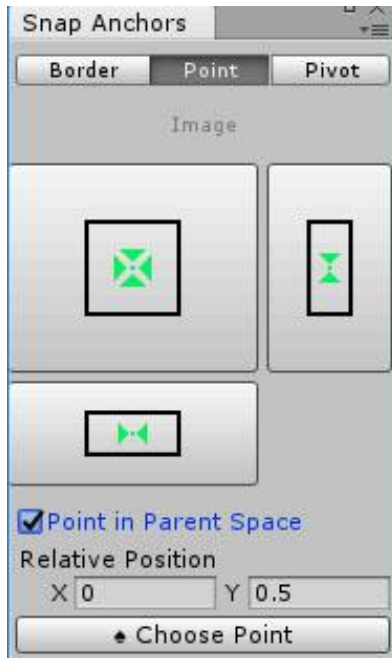
### 11.3.1 Border

In Border mode, the Anchors are snapped to the bounding of an object. This eliminates any pixel-distances and converts them to percentages relative to the parent. If the parent changes its size (e.g. due to a different resolution), the anchored object will change its position and size relative to that change.

### 11.3.2 Point

The Point mode adds some extra options to the window. It sets the anchors to a relative point of the selected object or its parent (if "Point in Parent Space" is selected). The "Choose Point" Button opens a popup where one of the most used options can be selected.

This option doesn't eliminate pixel distances. It is most likely used to only set the horizontal or vertical anchors and set the other axis in Border mode.



### 11.3.3 Pivot

The Pivot mode sets the Anchors to the current position of the pivot point of the selected object.

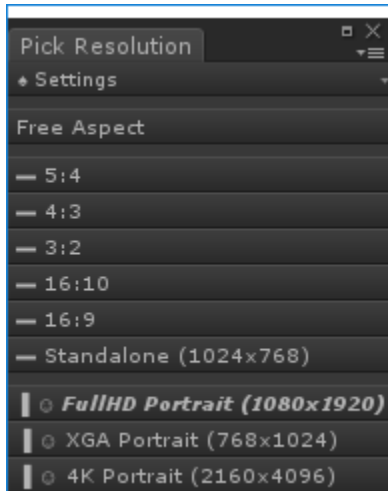
This option doesn't eliminate pixel distances. It is most likely used to only set the horizontal or vertical anchors and set the other axis in Border mode.

## 12 PICK RESOLUTION WINDOW

---

The Pick Resolution Window is a shortcut to change resolutions with one click. When working with different resolutions it can be very annoying to use the resolution drop down of the game view all the time. It is a lot faster to just click a button which is not hidden inside a drop down.

The Pick Resolution Window can be opened from the menu bar of Unity under "Window > Thera Bytes > Pick Resolution".



In that window, you can find all the resolutions and aspect ratios which can be selected. They are the same as in the drop down of the game view.

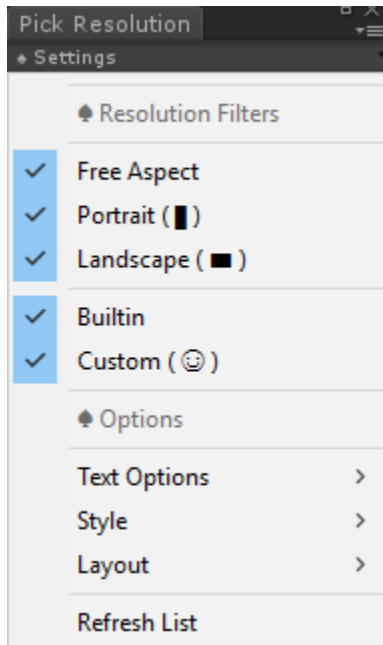
With default settings, there is a horizontal or vertical bar at the beginning of each entry name indicating if the entry is for landscape or portrait mode. Entries which are defined by the user have also a smiley indicator before the name 😊.

If a resolution matches the optimized resolution specified in the Resolution Monitor object, it is displayed in italics.

The currently active resolution of the main game view is displayed in bold.

## 12.1 SETTINGS

the very first control of the Pick Resolution Window is a drop down to change some settings. These settings are saved and loaded between the sessions.



### 12.1.1 Resolution Filters

With the Resolution Filters you can hide Entries from the list. This can be handy if you only work in one orientation or want to use only your own resolution entries.

The Filter options are:

- **Free Aspect:** Should the "Free Aspect" button be visible or not?
- **Portrait:** Should Portrait resolutions be in the list?
- **Landscape:** Should Landscape resolutions be in the list?
- **Builtin:** Do you want to have the predefined entries in your list?
- **Custom:** Do you want your own created entries in your list?

The filter options can be checked and unchecked to show or hide the respective entry.

Note that all filter conditions must be fulfilled to show an entry. An exception is the "Free Aspect" entry. It is not considered as builtin nor as custom.

### 12.1.2 Options

There are some options which allows to modify the look and layout of the window.

#### 12.1.2.1 Text Options

Text Options define what you see on the buttons. The upper three options are for the actual text:

- **Name and Size:** Displays the Label of the entry as well as the resolution or aspect ratio (e.g. "My Favorite Resolution (320x240)")
- **Name:** Displays only the Label of the entry – if the label has no text, the resolution / aspect ratio is displayed instead (e.g. "My Favorite Resolution")
- **Size:** Displays the resolution / aspect ratio without the label (e.g. "320x240" or "4:3")

The option "**Orientation Hint**" shows or hides horizontal or vertical bars in front of each entry to indicate if an entry is in landscape or portrait mode.

The option "**Mark Custom**" shows or hides a smiley in front of custom entries to indicate that the user created it.

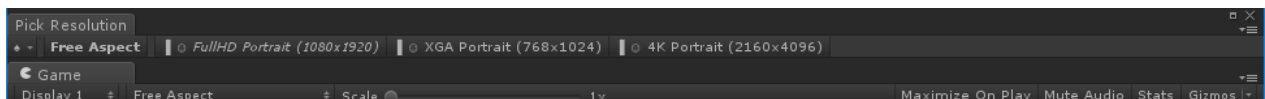
### 12.1.2.2 Style

You can use **Small** or **Big** style. Big style may not look as good as the small style but the buttons are easier to click and separated more clearly from each other.

### 12.1.2.3 Layout

The Buttons can be ordered next to each other (**Horizontal**) or below each other (**Vertical**).

The horizontal layout can be placed nicely above or below the game view. This way it doesn't need much space and is close to the area it affects.



### 12.1.2.4 Refresh

If you added, deleted or edited a resolution it is not immediately put into the Pick resolution window. To have it there just click Refresh.

In very rare cases some entries display the wrong resolution. A click on Refresh should resolve such a problem.