Better UI 2.0 – Beta

Table of Contents

Upgrade Guide 2

Changes – Overview 3

Resolution Monitor 4

- Screen Configurations 4
  - Settings 4
  - Fallback Logic 4
- DPI Overwrites 5

Resolution Picker Window 5

Better Locator 6

Better Content Size Fitter 6

Upgrade Guide

Important:

Before you install the Beta of Better UI 2.0 in your project, make sure you have a backup of the current state of your project (If you use source control, make sure everything is committed. If not using source control, make a copy of your project folder).

Installation:

To install Better UI 2.0 (Beta) follow these steps:

1. Open an empty Scene (File -> New Scene)
2. Delete the folder “Assets/TheraBytes/BetterUI”
3. Import the BetterUI_2.0_Beta.unitypackage
4. If you are using TextMeshPro, also import BetterUI_2.0_Beta_TMP.unitypackage
   Please Note: The latest version of TextMeshPro (which is included in Unity 2018.3) does currently not work with Better UI (will be fixed).

Known Issue:

It might happen that you see NullReferenceExceptions in your logs which are thrown by the ResolutionMonitor and the Materials ScriptableObjects. In this case, please restart Unity after installation.

On Unity 2018.3 there are several warnings about unused variables. These are false positive detections. (So, it is a Unity bug).

Only older versions of text mesh pro are working right now (actually the problem is the changes of editor styles, so this is not breaking the functionality but makes compile errors because of missing styles).

Changes – Overview

Here are the main features of the new version listed. Some minor changes may not in the list.

- The **Resolution Monitor** can define different **Screen Configurations** (Screen-Types / Screen-Modes) which are detected at runtime.
    1. All **"Sizer" Variables** of the Better-UI-Elements and Better-Layout-Elements can have different Settings for each Screen Configuration. Additionally, several **settings** of some components can be defined differently for different Screen Configurations
    2. There is a new Component: "**Better Locator**". It is used to change the RectTransform values for different Screen Configurations. (easily add it by right clicking RectTransform and selecting "Add Better Locator")
    3. The BetterHorizontalLayouotGroup and BetterVerticalLayoutGroup are obsolete. Now there is the **BetterAxisAlignedLayoutGroup** which can be horizontal or vertical (can be changed for different Screen Configurations)
    4. There is a "**BetterAspectRatioFitter**" which allows to have different settings depending on the Screen Configuration.
- There are new Layout Components
    1. The "**SizeDeltaSizer**" is like the "TransformScaler" but instead of changing the scale value (which should be avoided generally), it changes the SizeDelta value of the RectTransform with the Resolution.
    2. There is a "**BetterContentSizeFitter**" now with several additional features.
    3. There is a "**BetterLayoutElement**" which can set the min or preferred width or height relative to the Resolution (and Screen Configuration).
- The Tools got a slight lifting
    1. The **Resolution Picker** can now also display the Screen Configurations. A click on one of them can change the resolution accordingly or just "fake" the current screen to the Screen Configuration. The Window also displays all resolutions used by any Screen Configuration in italics.
    2. The **Snap Anchors** window has two modes now: Border and Point. The Pivot mode can still be activated inside the Point view. The Quick-Selection for a Point is not in a popup anymore but integrated the tool-window itself now.
    3. The Align/Distribute window as well as the Snap Anchors window are changed a bit optically to be more similar.

Resolution Monitor

Screen Configurations

You can design your UIs in the same scene for different screen types, sizes and / or orientations. In the Resolution Monitor you can configure for which kinds of screens you want to support responsive design.

Settings

When you hit the plus button in the inspector of the Resolution Monitor you can add a new Screen Configuration. When choosing the Name for the configuration, be careful: It is also the identifier for it which should not be changed once used.

For each screen configuration you can specify the optimized Resolution and DPI. Usually you always want to have the same values here but have switched width and height between Landscape and Portrait configurations.

Further you can specify the conditions for the detection of the screen configuration. You can specify one or more of the following conditions:

- **Check Screen Orientation**
  Detects if the screen is in Landscape or Portrait mode (if it can change during runtime, don't forget to call *IngameResolutionMonitor.Create()* at startup)
- **Check Screen Size**
  You can check if the height, width or diagonal size of the screen is in between certain values. These values can be specified in inches or centimeters.
- **Check Aspect Ratio**
  Checks if the current aspect ratio is between the given values – or not in between them (inverse). There is a drop down which helps finding the right values.
- **Check Device**
  If you deploy to different platforms you can check the type of screen here (Desktop, Console, Handheld, Virtual Reality, Touch Screen or Other)

After the Condition Section there is the Fallback Order section where you can specify fallbacks for the screen configuration (see below) if there are more than one.

Fallback Logic

The Fallback logic of Screen Configuration is flexible and complex and therefore not so easy to understand. Here is how it works.

The Resolution Monitor knows, which screen configurations are active (where all conditions are true) and which not.

1. The UI Element checks, if it has a special setting for the **upper most active screen configuration**.
    a. If true: it uses the settings.
    b. If false: it goes through the "**Fallback Order**" list of the configuration (even of the conditions are not matched) and checks for the first configuration which is used by the UI Element.
        i. If it finds setting for a such Fallback Configuration, it uses it.
        ii. If not: It goes on with 1. But with the **next active screen configuration**

2. If not any of the screen configurations is found in the settings of the UI Element, the **Fallback Configuration** is used (which has no conditions and is present for all settings in UI Elements).

Example

You have the following screen configurations:

- "Landscape Small" (Landscape mode and small size of the screen)
- "Landscape Large" (Landscape mode and large size of the screen)
- "Landscape" (Landscape mode with no further checks)

The order above is good. The most special case should always be on the top (so it is checked first).

To understand how to set up the fallbacks, let's understand the problem:

> We have a UI Element which specifies a setting for "Landscape Small" only (as well as the default fallback). The Resolution Monitor detects "Landscape Large" and "Landscape". These are not specified in the settings, so it will use the fallback.

In the given case, the first detected entry ("Landscape Large") probably should not use the Default Fallback but "Landscape" or – if that doesn't exist either (as in the example) – "Landscape Small". That also applies for other combinations of detections / UI-settings.

So, the Fallbacks should be specified as this:

- "Landscape Small"
    1. "Landscape"
    2. "Landscape Large"
- "Landscape Large"
    1. "Landscape"
    2. "Landscape Small"
- "Landscape" (here the order may be different)
    1. "Landscape Small"
    2. "Landscape Large"

DPI Overwrites

The Resolution Monitor contains a (empty) List now where you can overwrite dpi values of special Devices.

Some Devices have no or wrong DPI settings because the manufacturer forgot about it or was not accurate. If you are scaling by DPI with Better UI you can use that List if you found a problematic device (if so, please let me know).

Resolution Picker Window

The resolution Picker Window Has a "Screen Configuration" Section in the settings now.

- "Show" will display a button in the window for each screen configuration defined in the Resolution Monitor. A click on it will simulate that screen configuration (even if the resolution is not changed and the screen doesn't match the criteria).

- “Apply Resolution” will change the resolution to the optimized resolution of that configuration when clicked on it. If the resolution doesn’t exist in the list of resolutions, it will be added – automatically.

The non-fallback screen configurations have a diamond in front of the name. It is filled black, when they are active. A star (*) behind it indicates that the configuration is simulated right now. A superscript 1 next to it indicates that it is the first detected active screen configurations. These indicators are also in the resolution monitor window where you can simulate a configuration as well by clicking on its name.

Resolutions which are the optimized resolution of any screen configuration appear in italics.

Better Locator

A Better Locator is only relevant if you use more than one screen configuration. Easily add it by right-clicking on a RectTransform and select “Add Better Locator”.

It provides all the values which the RectTransform has but can store different values for different Screen Configurations. This allows for example to have some element on the top of the screen in portrait mode while having it on the left of the screen in landscape mode.

The easiest way to work with the Better Locator is by having “**Auto Pull**” enabled. This way you can simply work with the RectTransform itself (or any tools which modifies it, like moving it in the scene or using the Snap Anchors Window). The changes are automatically updated to the currently active configuration in the Better Locator.
Don’t forget to add all desired configurations, so that it always pulls into the right setting.

There is another option called “**Auto Push**”. That allows you to edit the values of the current Better Locator settings which are then automatically applied to the RectTransform.

If neither “Auto Pull” nor “Auto Push” is enabled, you can edit the values without preview. The buttons with the arrows will allow you to pull (Arrow down) or push (Arrow up) the state from or to the RectTransform manually.

Better Content Size Fitter

The Better Content Size Fitter extends the normal Content Size Fitter a lot. Of course, you can have different settings for different screen configurations. Besides that, there are some additional options:

- **Source**
  If not set the object itself is used for fitting. Otherwise it uses the size of the given object and tries to have the same size (plus the modifications – see below)
  Note: This doesn’t work with all kinds of objects yet. It is designed to be used with Text objects. This feature is experimental.
- **Has Max Width / Has Max Height**
  If you specify the Horizontal or Vertical fit to Min Size or Preferred Size, a checkbox will appear. If you check it, another section will appear where you can specify the maximum width / maximum height of the object. Even if it would be bigger than the specified size, it is trimmed to the given maximum instead.

- **Size Offset**
  The Size Offset will be added to the calculated size. Pass positive values to make it bigger and negative values to make it smaller. It is kind of a padding.
- **Is Animated** (Experimental)
  When the content of a (Better)ContentSizeFitter changes, it automatically applies the new size. If “Is Animated” is checked, it will not change the size immediately. Instead changes the size over the specified time until it has reached the target size.

Experimental means, that the given Feature works but not under any circumstances. So, try it. If it works you can use it also in production. If it doesn’t behave as expected, don’t use it for the given case.