

FastGUI Documentation

Contents

Introduction	2
Forum and Help.....	3
How to Use.....	3
Importing Fonts for NGUI.....	5
Organizing your PSD Groups and Layers.....	7
Widget Names.....	8
Image Button (UIButton).....	8
Checkbox (UIToggle)	8
Slider (UISlider)	9
Progressbar (UISlider)	9
Text Label (UILabel).....	10
Input Text Label (UIInput)	10
Sliced Image Button (UIButton Sliced)	11
Sliced Sprite (UISprite Sliced)	11
Clipping Area	11
Known Issues.....	12

FastGUI Documentation

Revisions

Revision Date	Revisions
04 April 2014	<ul style="list-style-type: none">• Added “Empty Widget”

Introduction

FastGUI allows you to quickly import a formatted PSD into your Unity project as a new NGUI screen, in only a few minutes. With pixel perfect precision, the layout you worked so hard on in Photoshop will appear identically in your Unity scene, with all the appropriate NGUI components attached and configured automatically.

When focused on Unity development, the last thing you want to do is waste your time trying to adjust your NGUI UI to match as it was planned in Photoshop by your artists. FastGUI automates the entire time consuming process of matching your artist's design with your Unity UI.

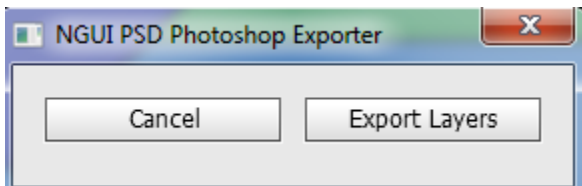
An example PSD file is included with the package, to demonstrate proper naming and grouping.

Forum and Help

The forum for FastGUI can be found at: <http://forum.unity3d.com/threads/156204-FastGUI-The-perfect-solution-for-GUI-Placement-form-Photoshop-to-Unity3D-RELEASED>

How to Use

1. Organize your PSD groups / layers (see below)
2. Run the Photoshop Importer plugin
 - a. \Assets\FastGUI\Photoshop Plugin\FastGUI Photoshop Plugin.jsx
3. Select "Export Layers"
4. Select the location in your Unity project to save the export
5. Import your fonts into Unity, into a common folder (see Fonts section below)

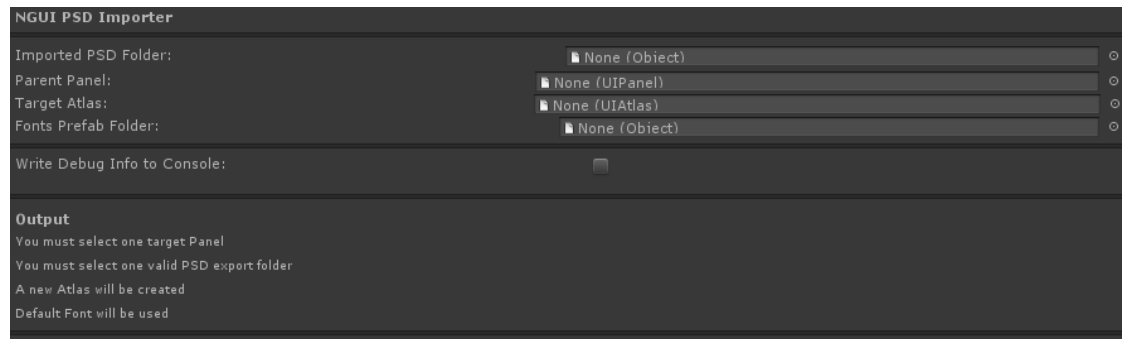


- b.
6. In your Unity scene
 - a. If you have no existing NGUI elements in your scene
 - i. Create a new layer to hold the NGUI elements. Give it a descriptive name like "NGUI"
 - ii. Create an NGUI Panel to act as the root
 1. NGUI > Create > Panel
 - iii. Select the new Camera (UI Root > Camera), then create another new panel
 1. NGUI > Create > Panel

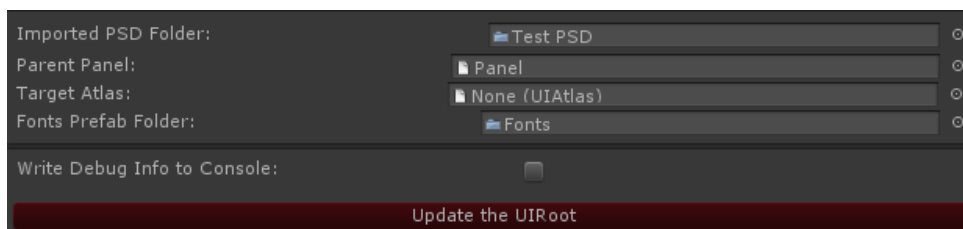
FastGUI Documentation



- 2.
- iv. Select the UI Root. Set its layer to “NGUI”, and assign this layer to all children
- v. Select the UI Root Camera
 1. Set “Culling Mask” to “NGUI” (uncheck everything else from the list)
 2. Set “Event Mask” to “NGUI” (uncheck everything else from the list)
- b. Open the PSD Importer tool
 - i. NGUI > PSD Importer > Import NGUI Screen
 - ii.
- c. Drag-drop the imported PSD folder to the “Imported PSD Folder” field
 - i. NOTE: This is the folder that was created in your Unity project by the Photoshop plugin
- d. Drag-drop the last Panel you created (step a, ii from above) to the “Parent Panel” field
- e. If your PSD has any text or label fields, drag-drop the Fonts folder to the “Fonts Prefab Folder” field



- f.
- g. Click the button “Update the UIRoot” button
- h. When everything is ready to go, a new button will appear – “Import It”. Click this button to begin the Import process.

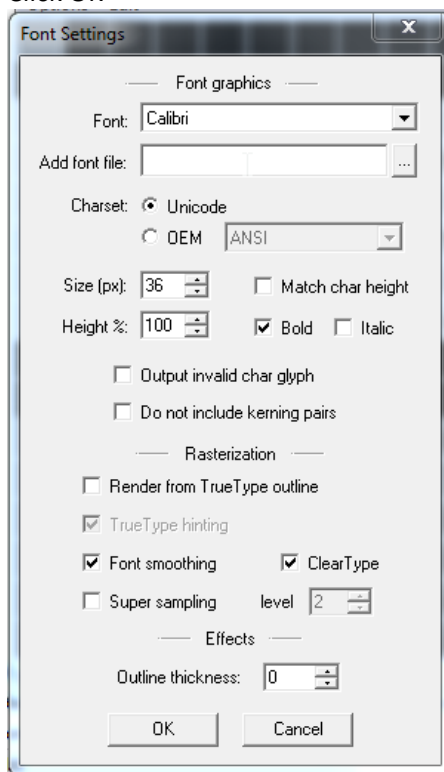


Importing Fonts for NGUI

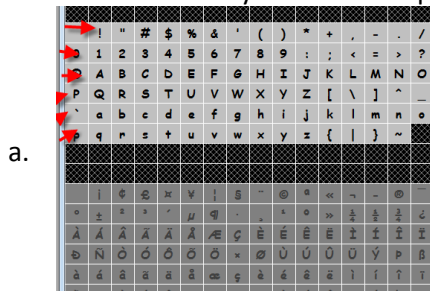
If your PSD contains text or input labels, you will need to import the fonts used before running the PSD import.

To import fonts, you will need to use BMFont (<http://www.angelcode.com/products/bmfont/>)

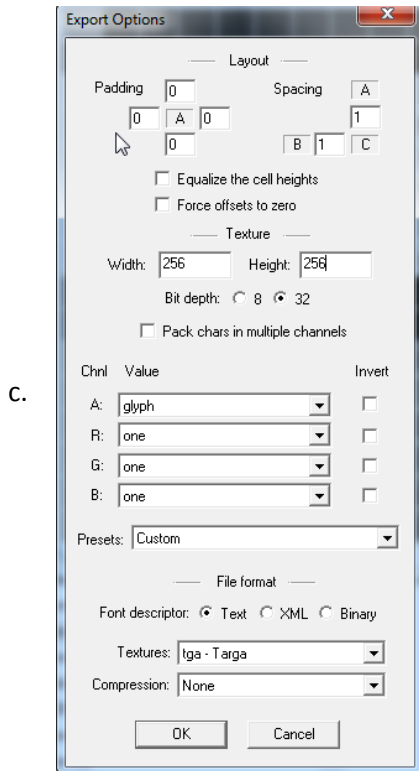
1. Run BMFont Generator
2. Options > Font Settings
 - a. Select your font in the dropdown
 - b. Set Font Size
 - c. Click OK



3. Select the characters you want to export



4. Options > Export Options
 - a. In the Textures dropdown, select "tga - Targa", No compression
 - b. You can adjust the Width / Height of the texture if the fonts won't fit in the texture



5. Options > Visualize
 - a. Make sure the selected characters fits in the texture (they must all fit into one texture). If they don't you can adjust the texture size in step #4B
6. Options > Save Bitmap Font As...
 - a. Give the save file the name of the font used in Photoshop (ex. KristenITC_24)
7. Open Windows Explorer, browse the saved folder
8. Rename the ".fnt" extension to ".txt"
9. Drag-Drop the folder into your Unity Fonts folder (create the folder if it doesn't already exist)
10. In Unity
 - a. Menu NGUI > Open the Font Maker
 - b. Drag the font txt and tga files to the appropriate slots in the font maker window
 - c. Type the new Font Name
 - d. Click "Create a Font without an Atlas"

Organizing your PSD Groups and Layers

The core of the FastGUI is how you organize your PSD groups and layers.

For the import, each group of your PSD represents a Widget or a Sprite which will be imported into Unity. During the import, each group's layers are merged together and exported to a PNG file (later imported into an NGUI Atlas).

The list below outlines the naming requirements for each widget type which can be exported. If a top level group does not contain one of these prefixes, the group will be exported as a Sprite (UISprite).

I've included an example PSD file in the package to show you how everything should be grouped and named.

Please note that each Widget must be contained in a top level group – they cannot be nested as a child in other groups.

Also note that groups and layers which have their visibility turned off (the “eye” is unchecked) will have their visibility turned on during the import. If you do not want a layer to be imported, you will need to remove it from the PSD.

Widget Names

Image Button (UIImageButton)

Image buttons are buttons which use images to denote the states of the button. Each button has four possible states:

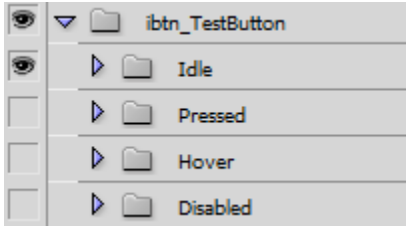
- Idle – this is the default button
- Pressed – this is the image when the button is pressed / clicked
- Hover – this is the image when the mouse hovers over the button
- Disabled – this is the image when the button is disabled. This state is Optional

The group name for Image Buttons starts with “ibtn_”, followed by a name for the button. For example, if you wanted to name the button ‘submit’, the group name would be “ibtn_Submit”.

Each Image Button required child groups to handle each state of the button. Only the “disabled” group is optional.

The names for the child groups are:

- Idle
- Pressed
- Hover
- Disabled



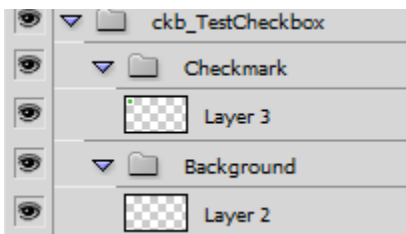
Checkbox (UIToggle)

A checkbox has two images, one to show the unchecked state, and one for the checked state. These images are independent of each other (ie, only one image is displayed at any given time).

Create a top level group with the prefix “ckb_” and the name (ex. “ckb_IAgree”)

Create two child groups, one for each state.

- Background – this is the default (unchecked) image
- Checkmark – this the image when the checkbox is checked



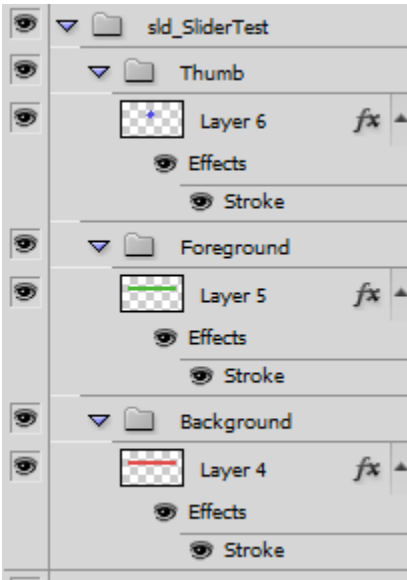
FastGUI Documentation

Slider (UISlider)

Sliders have three images which define the parts of the slider. The top level group has a prefix of “sld_” (ex. “sld_Volume”).

The three child groups are named:

- Thumb – the widget the user manipulates to adjust the slider values
- Foreground – the image to display when the slider’s value is 100%
- Background – the image to display when the slider’s value is 0%

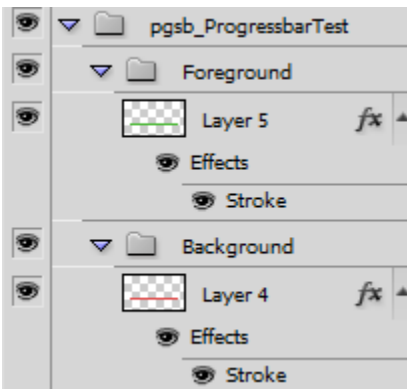


Progressbar (UISlider)

Progressbars are similar to Sliders, except they don’t use a Thumb as a manipulator. Both use the UISlider component.

The top level group prefix is “pgsb_”. The child groups are named:

- Foreground – the image to display when the slider’s value is 100%
- Background – the image to display when the slider’s value is 0%



FastGUI Documentation

Text Label (UILabel)

Text labels have a special name prefix which uses “txt_” plus the name of the font as you will name it in your Unity Fonts folder.

For example, if you named your font (in Unity) “KristenITC_24”, the prefix of the text label would be “txt_KristenITC_24”.

NOTE: The font name cannot contain spaces, but it can have underscores.

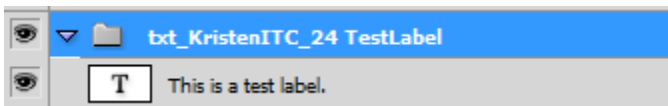
The second part of the text label will be the descriptive name of the label (the name used in the Unity project).

There is a space (“ ”) between the font name and the label name.

For example, if you use the font name “KristenITC_24”, and you want the label to be named “Enter_Your_Name”, the full name of the group would be “txt_KristenITC_24 Enter_Your_Name”.

Remember: You must import the fonts into Unity before you can import the PSD file. See the Importing Fonts section, above, for more info.

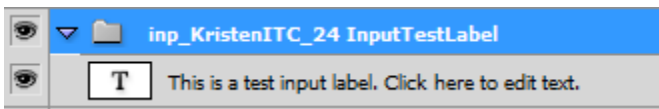
NOTE: The text label group can have only one layer, which uses the Photoshop Text tool



Input Text Label (UIInput)

Input text labels are set up the same as text labels (above), with the only difference being the prefix starts with “inp_”.

Ex. “inp_KristenITC_24 Enter_Your_Name”.



FastGUI Documentation

Sliced Image Button (UIImageButton Sliced)

A Sliced Image Button is identical to a normal Image Button, except the images use 9 slice scaling (see this link for more info on 9 slice scaling - <http://rwillustrator.blogspot.com/2007/04/understanding-9-slice-scaling.html>)

The parent group prefix name for sliced image buttons is "ibtnslc_". If your button is to be named 'submit', the parent group name would be "ibtnslc_Submit".

All child groups are the same as for the Image Button.

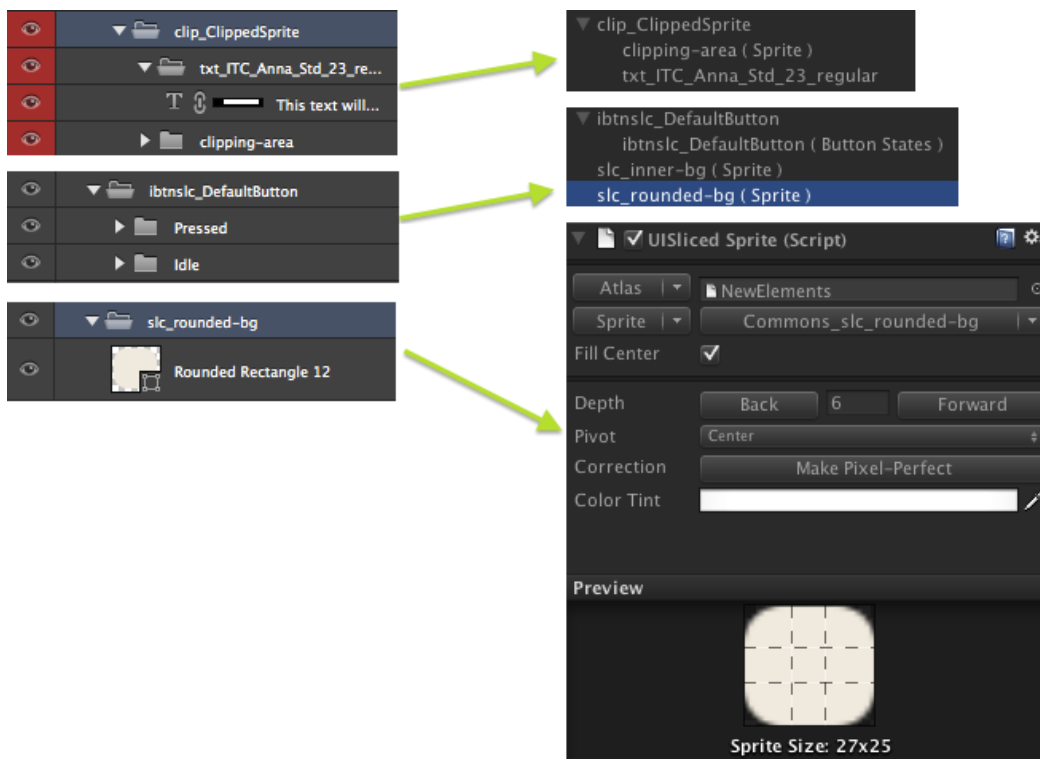
Sliced Sprite (UISprite Sliced)

The Sliced Sprite is a simple sprite using 9 slice scaling (see above) to scale the sprite.

The group prefix name for a sliced sprite is: "slc_". If your sprite is named 'header', the group name would be "slc_Header".

Clipping Area

A clipping area can be created using a group with the prefix "clp_", and child group "clipping-area". The child group "clipping-area" determines the size of the clipping area – all content inside this layer will be clipped based on the "clipping-area" size.



FastGUI Documentation

Empty Widget

To create an empty widget (a UIWidget with no sprite) use the prefix “wdgt_”. The widget will be sized to match the content within.

NOTE: The content within the wdgt group will not be imported into Unity. The content will only be used for sizing.

Known Issues

FastGUI uses a popup progressbar window in Unity to display progress as each widget is imported, added to the NGUI Atlas, and components are attached. NGUI v3 has added its own popup progressbar window when the Atlas is modified. This popup progressbar window can obscure or take over the progressbar window used by FastGUI.