PSD2uGUI USER GUIDE

INTRODUCTION

Congratulations! You have purchased your own copy of PSD2uGUI and, in a few steps, you will be using it but, is highly recommended to read this document to take all the advantages of this software. Remind PSD2uGUI is a tool designed to be used in compliance with Adobe Photoshop and Unity3D engine. The tasks the developers usually have to repeat along the Unity software lifetime is:

1. Design the UI,

2. Export from Photoshop or other design tool,

3. Import to Unity,

4. Place the ítems,

5. Add behaviour (buttons, toggles, panels, etc),

6. Test,

7. Back to design if needed.

This tool allows to make this task as easier and fast as possible, in fact, it lets the developers to use two “just one click” ways to perform it: By using Unity Sprites feature or by using the new UI system introduced with Unity Engine 4.6 and all the underlaying features.

When the users choose sprites as target, they should know the software will only be able to allocate ítems into the screen by following Photoshop coordinates but it won’t add any more far from this behaviour.

However, when new uGUI system is selected, developers can enjoy new game interactions and features like image borders, buttons, toggles, etc.

Accordingly to the last paragraphs, the software consists of two well differenced parts:

- Adobe Photoshop Export Script called “*psdDigest.jsx*”. We use Photoshop due to is the most popular and extended image edition tool, but if you use another tool and you keep the same format output, it will works as well.

- A set of Unity c# scripts in folders called “*scripts*” and “*editor*”. In addition, this guide is provided and a “.PSD” example document too. Both are very usefull to understand how PS2uGUI works so, please, take them into account if you have planned to put your UI design into your Unity proyect. Of course, don’t forget this software is continuously under development and it is not errors free. If you see one, please, let us know it via email.

SYSTEM REQUIREMENTS

Windows XP SP3, Vista, 7, 8, 8.1
Mac OS X 10.6+.
Adobe Photoshop (Tested on Photoshop CS3 and CC).
Unity Engine 5.0.

PACKAGE CONTENT

1. Plugins folder: PSD2uGUI.dll which allows to import using layer naming conventions.

2. Editor folder: PSDImport.cs, AssetProcessor.cs, PSD2uGUI Editor.dll.

3. psdDigest.jsx Photoshop javascript file.

CHANGELOG

Changes of 1.85v

- When importing as uGUI, the canvas now has a Canvas Scaler component.
- Thanks to the Canvas Scaler component, the asset doesn't need to compute coords and sizes that depends on the screen, it will scaled automatically and users can see the UI as is. Important: The main resolution will be the chosen one from Photoshop.
- Now, an uGUI button can be parsed with no text layer.
- A GameObject which is parent of an uGUI control, now has a RectTransform.
- DX Editor Window now has a new button. It allows to set Anchors to Corners of each imported UI ítem (Notice that could be some uGUI controls shouldn't be change their anchors to work, like scrollbars or sliders).
- The editor window now has scrollbars to navigate if content is wider than the size used.

Changes of 1.8v

- Atlas creation feature. It allows to create an sprite atlas from your images.
- Two ways for atlas creation are allowed, Unity method and Max Rects Algorithm.
- Built for Unity 5.

Changes of 1.72v

- Fixed an issue when layer just has TOKEN_ as layer names.
- Unity 5 ready.
- Unity 4 dll comes as ".zip".

Changes of 1.71v

- Fixed image import settings issue. Now you can edit them.
- PS Script now is more robust against errors.
- Experimental support for Unity 5 RC2.

Changes of 1.7v

- There was a problem with the name of the root game object of the imported UI. Now it s fixed.
- There was an annoying performance issue when importing the UI to Unity engine. It has been fixed but now, the folder with the UI images exported from Photoshop must be under Resources folder on your Unity Project.
- There was an issue with the PS Script when the image partially is placed out of the canvas. It was cropped but now, it exports as it should.

Changes of 1.6v

- PS Script now creates a new folder again, and place the exported files into this one. The folder name is the file name with extension. (This feature was removed once but it has been restored due to is better for keeping the exported items sorted).
- PS Script now has a Stop button. It allows to stop the operation before exporting everything.
- PSD2uGUI Editor Window and PSD2uGUI DX Editor Window now allows to drop the folder where the software will try to find de UI images. This feature fixes a performance and memory bug, due to the entire Asset folder was scanned to load any image found.
- Some text has been added to the Editor Windows in order to help the developer about how the plugin works.
- The content of the plugin now is stored in a Folder called PSD2uGUI.

Changes of 1.5v

- PS script now allows to select the folder where do you want to export both layer files and exchange file (only if a checkbox is checked). If you use folder naming conventions, this new folder will be placed under the export folder. It will work as ussually if the checkbox is not checked.
- Portrait resolutions.
- PS script bugfixes. (Layer names never will trim white spaces).
- ".toParse" files will be named with the PSD name (psdDigest no longer be used).
- Unity plugin now won't hijack the Main Camera gameobject, it will create one for the new UI.
- NEW PSD2uGUI DX Editor Window.
- It has a new game object field to drag and drop the file we want to parse as new

uGUI.

- In addition, a new GameObject field has been added in order to choose a gameObject from the scene as root for the new uGUI items that will hang from after exporting. If no GameObject is dropped, it creates a new one, using the name of ".toParse" file.
- It will only export to uGUI.
- It has a new button for reset by default values.
- This new Editor Window shows the PSD LayerSet tree.
- This tree parsing is compatible with PSD with PSD2uGUI layer naming conventions. It means it will parse the names and auto select the target UI control.
- For old PSDs or PSDs with no layer naming conventions, it will show the tree, and will allow to the developer to select the target UI control to export by using a popup window item.

Changes of 1.3v

- Photoshop Script has been changed to allow Custom target resolution export.
- Now, Photoshop script can export the exchange file only, avoiding to exporting each layer every time.
- Export file format extensión has been changed from ".txt" to ".toParse".
- Now, Unity plugin can parse all ".toParse" files under asset folder and import all of them as ready to run uGUI.
- Unity 5 Beta 18 experimental support.
- User guide has been updated fixing issues.
- Minor bugfixes.

Changes of 1.2v

- PSD2uGUI won't create UI folder any more. Developer can control the output folder using layer naming conventions as usually. If no folder is targeted, by default PSD2uGUI folder will be used in order to avoid file name collisions with previous files of other sources.
- Changed the method to parse the layer names. Now underscore symbol can be used in image layers.
- User guide has been updated fixing issues.

Changes of 1.1v

- Minor bugfixes.
- Clean software.

Changes of 1.09v

- Added compatibility for Release Candidate 1.
- Fixed a bug when trying to build an Unity Project.

Changes of 1.08v

- Fixed a bug when trying to build a project due to no Editor scripts should use UnityEditor namespace.
- Fixed a bug due to b21 changed the name of an element of RenderMode enum.
- FIxed a bug when tring to locate the sprites under Windows.

Changes of 1.07v

- The user can choose folders where the exported files will be placed following layer naming conventions.
- The files imported by unity now have tags with the name of the folder where they are placed.
- Following the same layer naming convention style, user can define tags for each target game object created. (Be aware the tags must be defined in Unity before).
- Minor bugfixes.

Changes of 1.04v

- Now, thanks to AssetProcessor script, PSD2uGUI asset can set the same tag ("PSD2uGUI") to the sprites in order to allow atlas creation and automatic drawing optimizations (it could need Sprite Packer and Unity Pro license).
- In addition, when sprite assets are imported, PSD2uGUI can automatically set borders if the software can parse it from sprite name.
- Decreased the number of load operations at execution time.

FAST GUIDE

The basic functionality of this software is to allocate the graphics as photoshop would do. To achieve it, follow the next steps:

1. Take your PSD file document and place it in your Unity Engine proyect, under “*Assets*” folder.

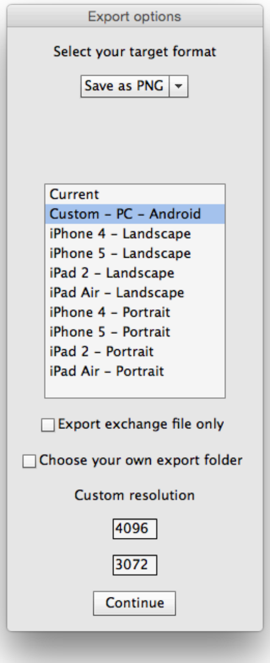
2. Put psdDigest.jsx file provided with this software package under Photoshop “*scripts*” folder (X:\Program files\Adobe\Your Photoshop\Presets\Scripts, by default on Windows. /Applications/Your Photoshop/Presets/Scripts on OS X). It will be available from photoshop.

3. Run the PSD file from your *Assets* folder.

4. Once Photoshop is open, make sure the layers you want to export are visible. (You can choose which of them will be exported by switching between layer visible or not).

5. Click “file” from Photoshop menu bar and select “scripts” (or “action sequence” depending on your Photoshop version). Then click on psdDigest script.

6. Click yes and select your target file type, compression, color depth, and target resolution. Select if you only want to export the exchange file or if you want to select your own target folder and hold on for export operation.



Select your target format

Save as PNG

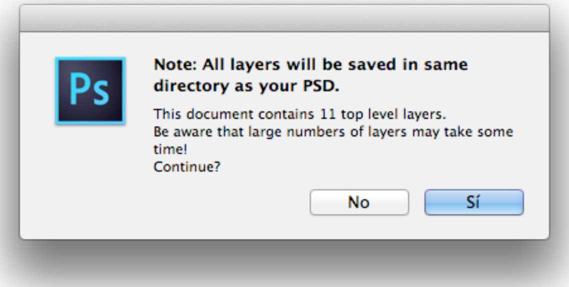
□ Export exchange file only

□ Choose your own export folder

Custom resolution

4096

3072



Note: All layers will be saved in same directory as your PSD.

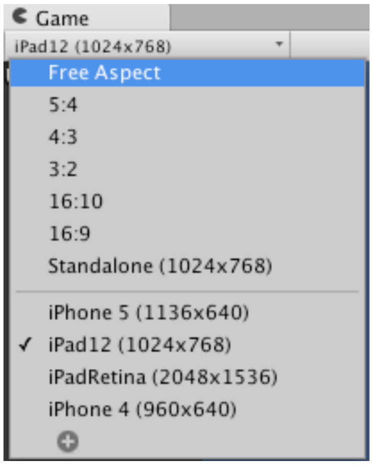
This document contains 11 top level layers.
Be aware that large numbers of layers may take some time!
Continue?

7. When finished, a new folder should have been created called as your psd is, with everything exported inside. Ensure you move this folder into any Resources directory under your project Assets path.

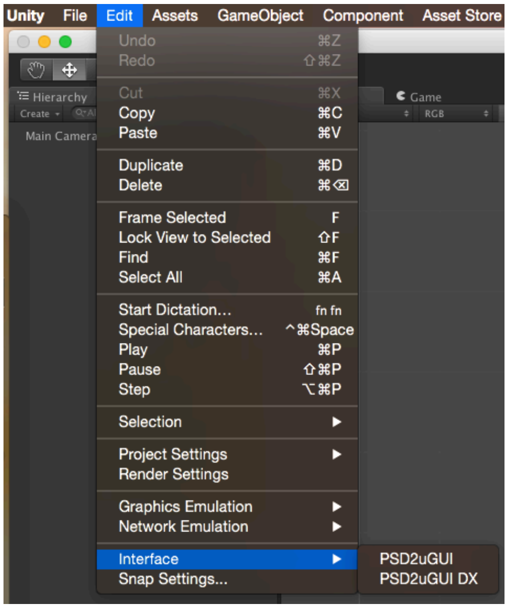
8. Now, take PSD2uGUI folder provided as the software package and place it under your Unity proyect Assets folder (if it is not placed yet). Run Unity Engine and hold on while assets are imported.

9. Ensure your Project can import assets as 2D Sprites (Edit-Project Settings-Editor).

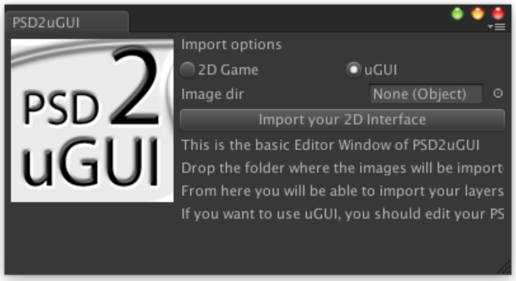
10. Configure Game window size for the resolution you exported previously from Photoshop.



11. If nothing were wrong when importing, you should have another menu item under “Edit” menu bar item, called “Interface”. It has two options, click one in order to open a new tab window.



12. If PSD2uGUI is selected, basic Window Editor of PSD2uGUI software will be shown. There you can choose between uGui (only >= 4.6) or Sprites. Select one, click on "Import psd" button and lets the magic do the rest. You've noticed you must drop the folder where the image files are placed, as both images shows below.

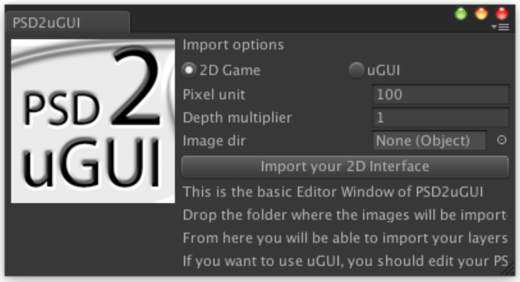


Import options

○ 2D Game ● uGUI

Image dir None (Object)

This is the basic Editor Window of PSD2uGUI
Drop the folder where the images will be import
From here you will be able to import your layers
If you want to use uGUI, you should edit your PS



Import options

● 2D Game ○ uGUI

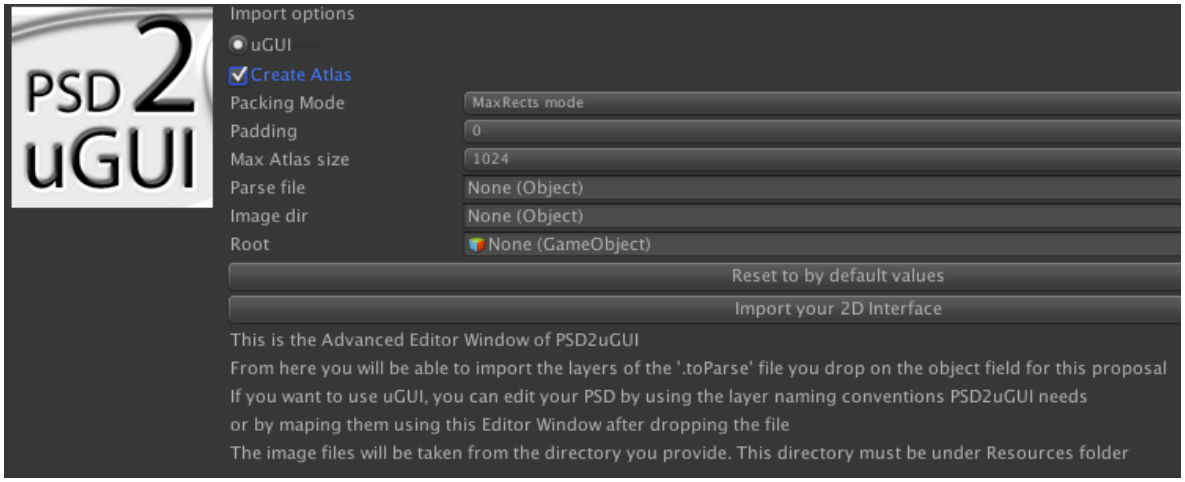
Pixel unit 100

Depth multiplier 1

Image dir None (Object)

This is the basic Editor Window of PSD2uGUI
Drop the folder where the images will be import
From here you will be able to import your layers
If you want to use uGUI, you should edit your PS

By selecting PSD2uGUI DX, the next window will be visible:



Import options

● uGUI

☑ Create Atlas

Packing Mode MaxRects mode

Padding 0

Max Atlas size 1024

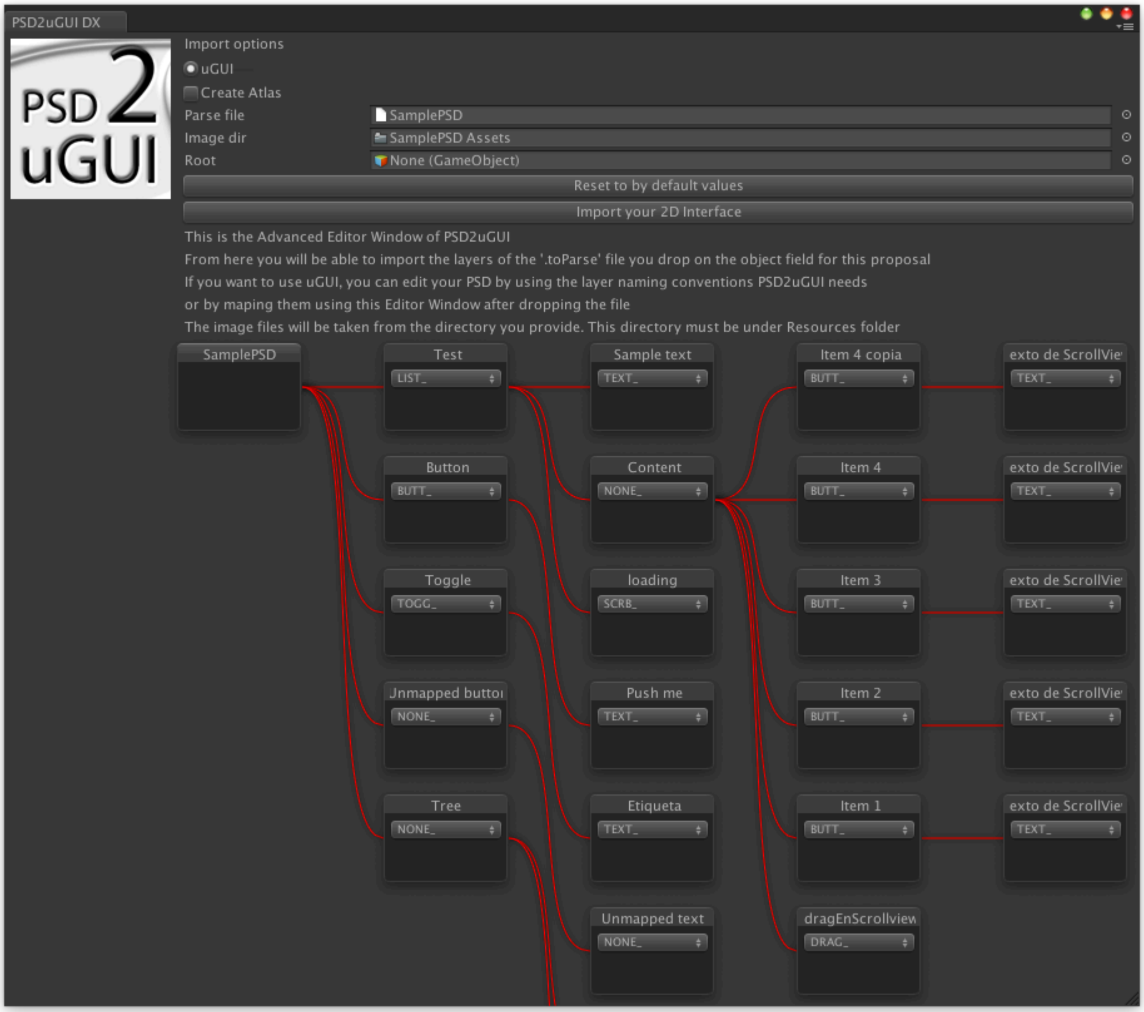
Parse file None (Object)

Image dir None (Object)

Root None (GameObject)

This is the Advanced Editor Window of PSD2uGUI
From here you will be able to import the layers of the '.toParse' file you drop on the object field for this proposal
If you want to use uGUI, you can edit your PSD by using the layer naming conventions PSD2uGUI needs
or by maping them using this Editor Window after dropping the file
The image files will be taken from the directory you provide. This directory must be under Resources folder

You must provide the exchange file, the directory where the asset will try to find the UI images and the root game object (optional). When ".toParse" file is dropped into Parse File field, you will see the layer tree. Set up each node by selecting the import options of your wish, but remeber PSD2uGUI needs concrete layer structs to work properly (learn how to on "PSD Layer Structures" section).



PSD2uGUI DX

Import options

● uGUI

□ Create Atlas

Parse file: SamplePSD

Image dir: SamplePSD Assets

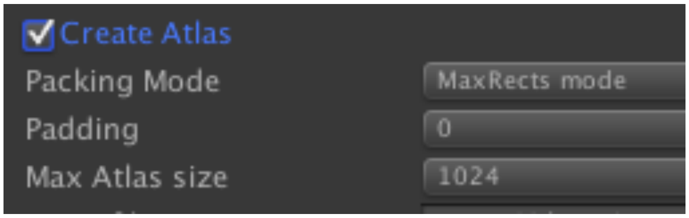
Root: None (GameObject)

Reset to by default values

Import your 2D Interface

This is the Advanced Editor Window of PSD2uGUI
From here you will be able to import the layers of the '.toParse' file you drop on the object field for this proposal
If you want to use uGUI, you can edit your PSD by using the layer naming conventions PSD2uGUI needs
or by maping them using this Editor Window after dropping the file
The image files will be taken from the directory you provide. This directory must be under Resources folder

13. Atlas creation. It is the new great feature of the asset. It allows to fill an sprite with all the layers you did import previously from Photoshop. If the checkbox is marked, three new options will be shown:



☑ Create Atlas

Packing Mode: MaxRects mode

Padding: 0

Max Atlas size: 1024

The first one is for choosing the algorithm to create the atlas. Unity mode uses packTextures algorithm and will downsample the images to fit the sprite if able. MaxRects mode is better but won’t create the sprite if all the images don’t fit on the atlas texture with Max Atlas size.

Padding means the space between images inside the atlas.

PSD2uGUI IN DEPTH

Once you have seen the basics, now you are ready to learn the advanced behaviour defined as "*commands*".

Commands

Commands are instructions the developer can send to Unity directly from Photoshop, like in a Unix/Windows Shell but simpler. It only will work when a command is set as name to a LayerSet in Photoshop but not when is set to a base layer.

In order to specify a concrete command, the following syntax should be followed (pay attention on "_", ":" and "=" are mandatory separators):

*[TOKENname_] gameObjectName [:var=val]**

Where TOKENname_ can be only one of the following (or empty):

- TOGG_: It tells to Unity this layer builds an UI Toggle.
- TEXT_: It tells to Unity this layer builds an UI Label.
- INPT_: It tells to Unity this layer builds an UI Text Input.
- PANL_: It tells to Unity this layer builds an UI Panel.
- BUTT_: It tells to Unity this layer builds an UI Button.
- SCRB_: It tells to Unity this layer builds an UI Scrollbar.
- SLID_: It tells to Unity this layer builds an UI Slider.
- LIST_: It tells to Unity this layer builds an UI ScrollView.
- DRAG_: It tells to Unity this layer builds a Draggable image object.
- DROP_: It tells to Unity this layer builds a Droppable area object.

- Otherwise or empty: The layer will be treated as plane image, with no behaviour attached by Unity. Where gameObjectName will be the name of the game object that matches in Unity. and, at last, some variables with their values can be sent with the command if developer wants.

Variables

PSD2uGUI can manage the following variables (some one may not be available for some tokens) :

- w: The width of the element in the Unity UI.

- h: The height of the element int the Unity UI.

- x: Photoshop position over X-Axis.

- y: Photoshop position over Y-Axis.

- fs: Font size (for text).

- Al: Alignment (for text): It can be “l” (Left), “c” (Centered) and “r” (Right).

- b: new variable added for border creation. It must be used with layers but not with Layersets. An example could be the next sentence: LayerImageName-b=1=1=1=1 Note some different points:

  a. Due to is used by final image layers, we need to use different separator (“-“, the exported image can’t use some characters like “:” we ussualy use in the image name).

  b. Each number separated by “=” character means the pixels of the image that Unity will take for this sprite border. We can use from one number up to four.

- tag: The variable to decide the tag assigned to the target game object on Unity side.

- @: special separator for layersets which means target folder name.

The following sentences could be a well formed command examples for PSD2uGUI:

TEXT_Example text:fs=12:al=l BUTT_Button1:w=130:h=18

The following sentences could be a bad formed command examples:

TOGG Toggle:w=130 TOGG_Toggle:w= TOGG_Toggle w=130

One of the most powerfull features the new UI System has is called “border”. It is a tool with which we can create borders for symetrical images. It allows to save memory due to we don’t need an image with the real size to fill the target screen rect. It repeats the borders of the image avoiding to lose image quality.

Regarding the previous info and its requirements, when a command is set, Unity expects some layer structure that depends on the kind of the command specified. It means, if you want it to work, you have to build and design the layers like we will teach you soon.

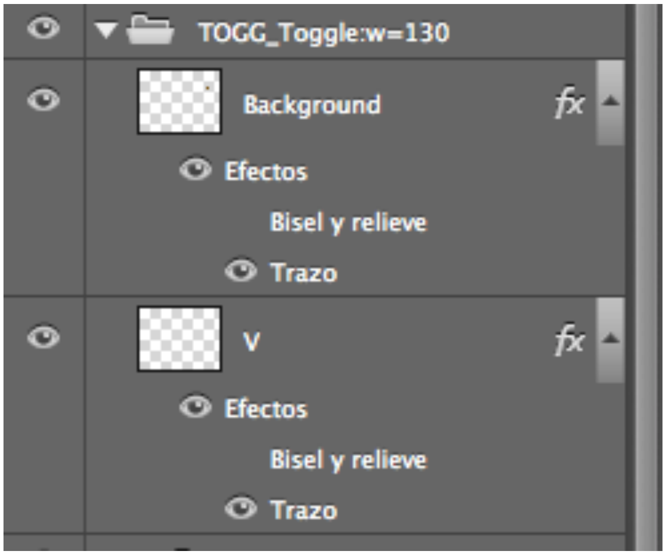
PSD LAYERS STRUCTURES

From this point the document will guide the designers telling them how they should build the UI PSD documents. **Please, be aware to keep layers sorted as shown in the images, changes will affect the final result.**

The next structures are based on image border feature, but they can be used with a complete image that fills the desired target screen rect.

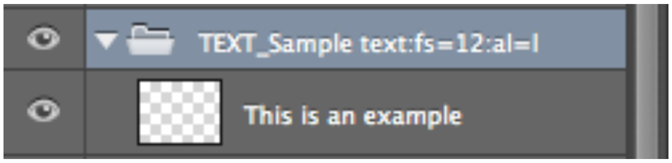
Toggle Photoshop structure

Like image shows, Toggle must have two base sublayers with the image of the background of the toggle and the image for the checkmark.



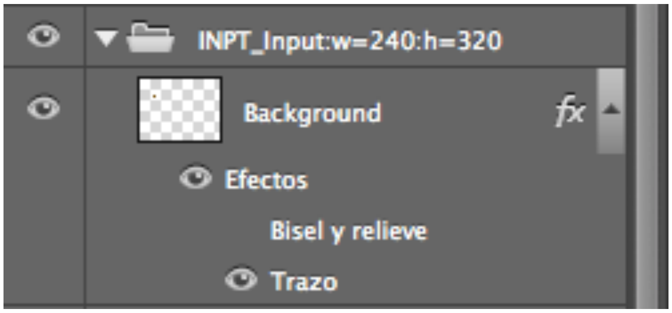
Text Photoshop structure

The image shows a Label must be built with one base sublayers which name will have the text to be shown. It will never be affected by w, h, x and y variables. Width is calculated using the nunmber of characters to show and the font size. The height is computed using font size too.



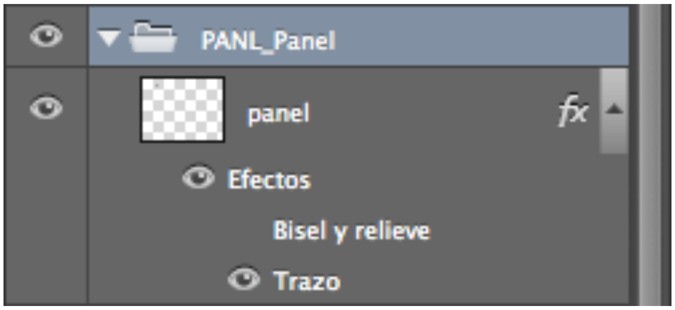
Input Photoshop structure

As Text layers, Input layers also be composed by one base sublayer but for the background instead of the text. The background área will be the área where the final user can input text. X and Y variables don't affect Input due to is placed using photoshop coordinates.



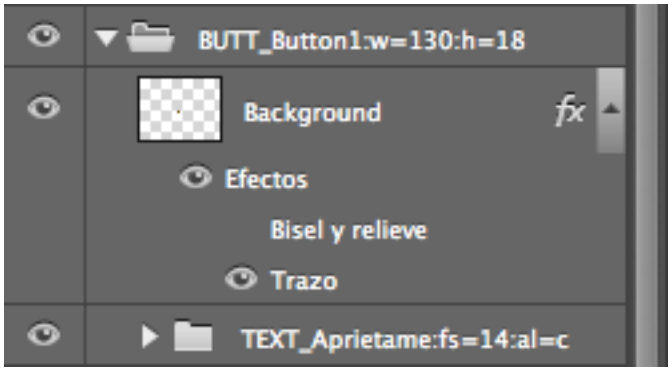
Panel Photoshop structure

Built with one base sublayer with the background that matches with the panel background. The variables don't affect the panel, it will always fill the entire screen by default.



Button Photoshop structure

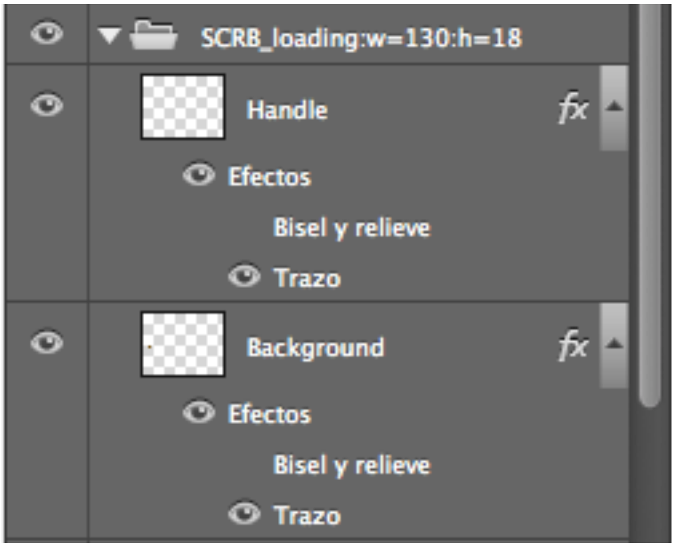
The button is one of the complex structure. It has a base sublayer for the background and a second sublayer as layerset for the button label.



- BUTT_Button1:w=130:h=18
  - Background fx
    - Efectos
      - Bisel y relieve
      - Trazo
  - TEXT_Aprietame:fs=14:al=c

Scrollbar Photoshop structure

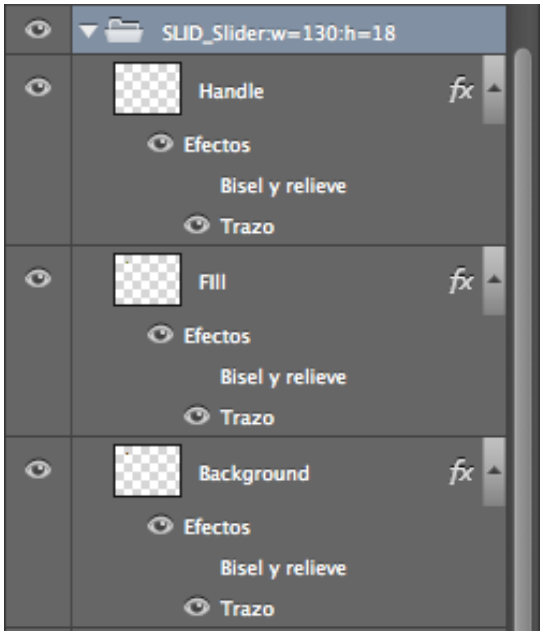
Other of the complex structures, scrollbar are built with two base sublayers, the background and the handler to make scroll. It can be horizontal or vertical, depending on the diference between width and height. If width is greater or equal than height, then it becomes horizontal scrollbar. It becomes vertical otherwise.



- SCRB_loading:w=130:h=18
  - Handle fx
    - Efectos
      - Bisel y relieve
      - Trazo
  - Background fx
    - Efectos
      - Bisel y relieve
      - Trazo

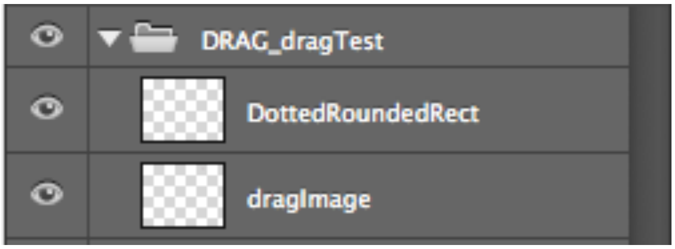
Slider Photoshop structure

It seems to be the same than scrollbar but is built using one more base sublayer image for filling the progress bar. As scrollbar does, if width is greater or equal than height, it becomes horizontal slider. Vertical otherwise.



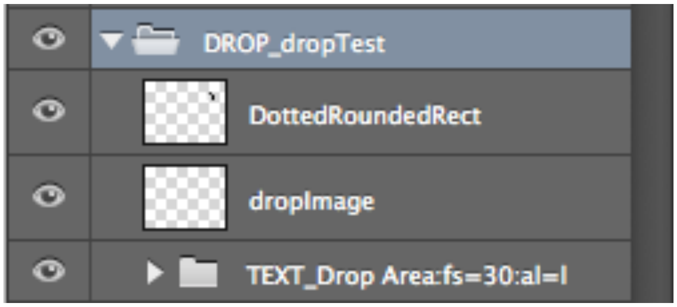
Drag item Photoshop structure

A drag ítem consists of two base sublayers: One of them allows to denote the area where the second image will be into. It uses DragItem script to perform the drag operation.



Drop item Photoshop structure

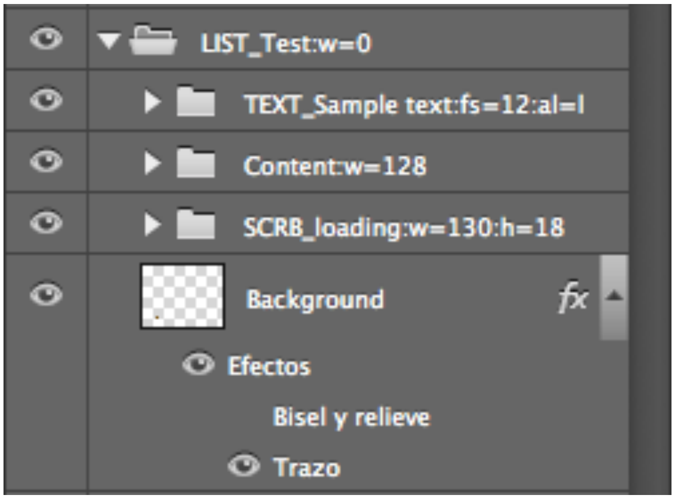
A drop ítem is built using the same two base sublayers, in addition to a text label layerset. It can't be draggable, and the size of "dropImage" denotes the área where the user can drop ítems. It uses DropItem script to perform the drop operation. You can also choose the size of the drop área by adding "w" and "h" variables to DROP Layerset name. If you do, the image size corresponding with the variable will be ignored.



DROP_dropTest
DottedRoundedRect
dropImage
TEXT_Drop Area:fs=30:al=l

Scrollview List Photoshop stucture

An scrollview is an UI component with we can do scroll around a set of images and their descriptors. It is built using a label as description or title, using a set of other componets as buttons, drag&drop ítems or raw images under Content layer, and an scrollbar. We can touch it with inertia like lists in a cellphone. In the example provided comes as an style sheet the user can modify later on Unity Engine.



LIST_Test:w=0
TEXT_Sample text:fs=12:al=l
Content:w=128
SCRB_loading:w=130:h=18
Background fx
Efectos
Bisel y relieve
Trazo

CONTACT

If you have any questions, suggestions or your are so kind to report us any error could happen, send us an email to dreamset.studios@gmail.com, our team will answer you as son as posible.