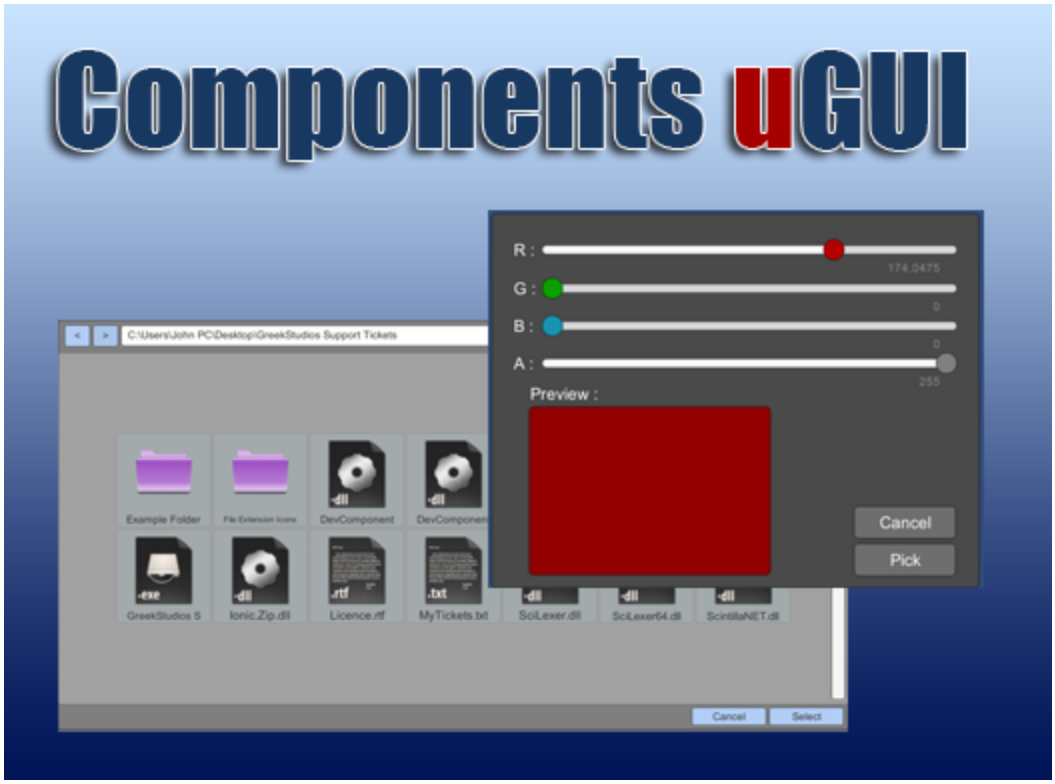


Components uGUI

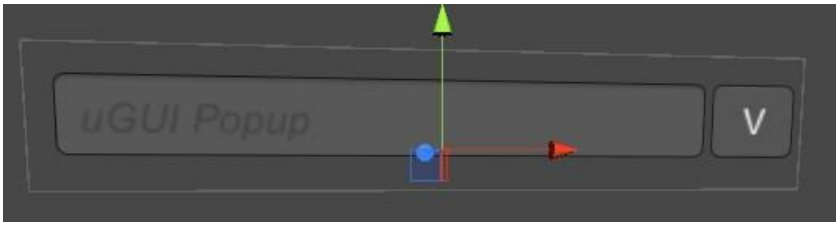
Components uGUI is a collection of some customizable components (systems) based on Unity 4.6 UI written in C# (C Sharp). The current version is 1.0 and contains four components with their editors.

Content :

Pop-up (Combo box)

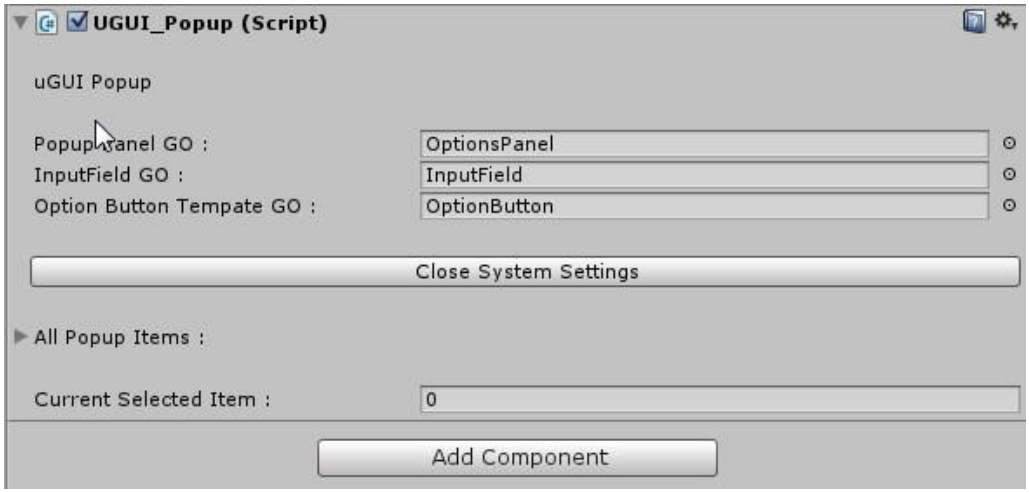
This component is a group of choices which allows users to choose one of them using the drop down menu or even type their own. In order to create a new pop-up box just drag & drop the prefab from the 'Basic uGUI Components' into your scene.

This is how it would look like :



You can modify the way it looks, the colors or the position, keep in mind that you can resize it and it would keep the position of each element.

This is the custom editor, you can click the 'System Settings' in order to view the settings which are not supposed to be changed unless if you know what are you doing. It's actually the 3 game objects, the container of the items container, the input field and the button template.



In order to add items into the pop-up box just add strings at the 'All Popup Items' array. You can set the selected item from the editor or even at runtime.

Here we will cover all the events / functions which can be called at runtime.

Functions :

AddOption (string name)

You can call this function from any script and it would add a new item into the pop-up box and it would automatically re-draw the component.

In order to call it you have to get the pop-up gameObject. Into your script, create a new variable

```
Public GameObject myPopup;
```

then, call the function like the way bellow :

```
myPopup.GetComponent<uGUI_Popup>().AddOption("My Option Name");
```

My Option Name is the name of the new item.

RemoveOption(int index, [optional] string name)

You can call this function from any script and remove the given item. There two ways of using this function. If you like to remove an item by it's index then you should leave the second parameter (string name) empty, then this function will remove the option with the given index.
(The index of the 'All Popup Items' array.

The second way is to remove an option by it's name, in case of having more than one item with the given name the function will remove the first one. Using this way you should call the function by giving on the first parameter (int index) any value (like 0) and then on the name parameter the name of the option you like.

In order to call it you have to get the pop-up gameObject. Into your script, create a new variable

```
Public GameObject myPopup;
```

then, if you want to call the function using the first way :

```
myPopup.GetComponent<uGUI_Popup>().RemoveOption(2);
```

2 is the index of the item you like to remove.

If you want to call it using the second way :

```
myPopup.GetComponent<uGUI_Popup>().RemoveOption(0,"Item Name");
```

Item Name is the name of the item you like to remove.

AddOptions(string[] names)

You can call this function from any script and add an array of new items.

In order to call it you have to get the pop-up gameObject. Into your script, create two new variables

```
Public GameObject myPopup;
```

```
Public string[] allNewItems;
```

then, call the function like the way bellow :

```
myPopup.GetComponent<uGUI_Popup>().AddOptions(allNewItems);
```

allNewItems is the string array which will be added.

RemoveOptions(int index, int count)

You can call this function from any script and delete a number of items

In order to call it you have to get the pop-up gameObject. Into your script, create a new variables

Public GameObject myPopup;

then, call the function like the way bellow :

myPopup.GetComponent<uGUI_Popup>().RemoveOptions(3,4);

the two integers are the index, count. Index is the index(position) of the first item and count is how many items will be removed after the index. So on our example it would remove item 3,4,5,6,7

Events :

OnItemSelect(int selectedItemIndex)

Called each time an item is selected.
The parameter returns the current selected item

OnPopupOpened()

Called each time the pop-up menu open.

OnPopupClosed()

Called each time the pop-up menu close.

Color Picker

This component allow users to pick colors at runtime, the design is simple and it can be easily extended.

Functions :

ResetColor()

It reset the color values to white.

GetColor()

It returns the current color.

Events :

OnPickButtonPress()

Called each time the Pick button is pressed.

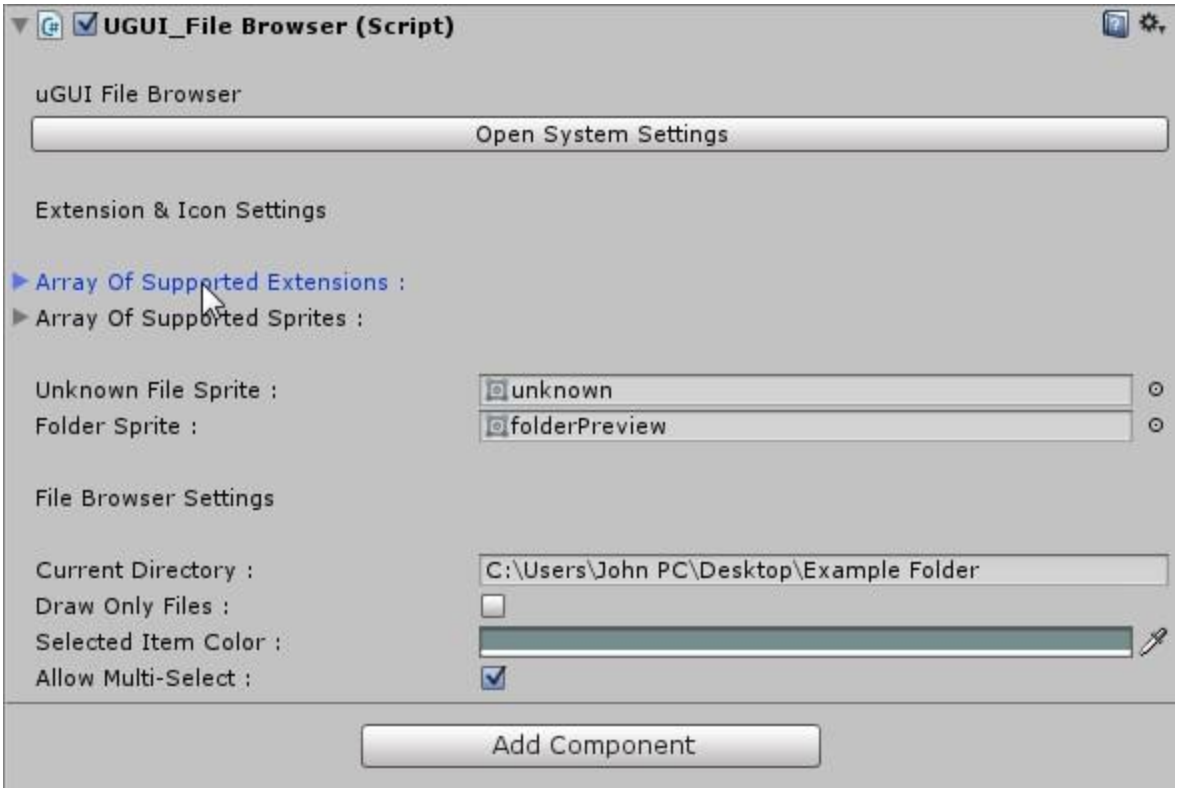
OnCancelButtonPress()

Called each time Cancel buttons is pressed.

OnColorChangeEvent()

Called each time the color is changed.

File Browser



With file browser you can allow users to browse through the local disk and select file(s). Multi-select is supported, it also renders directories which can be disabled by checking the 'Draw Only Files'

The Current Directory is the folder which will load.

You can modify the selected color of the files and allow multi-select.

To support more extensions just add their name at the 'Array Of Supported Extensions' and then at the same index add the icoun you like.

Note that you can find a .psd file (photoshop file) with the background of the file type icon in case you like the same style.

Functions :

ForceRefresh()

It refresh the system so it would render new changes on the directory.

NavigateTo(string directory)

It loads the given directory.

Events :

OnItemSelect(string selectedItem)

Called each time an item is selected/unselected. The parameter returns the selected item name.

OnDirectoryExpand(string directory)

Called each time user click and expand a new directory. The parameter returns the directory path.

OnDraw()

Called each time the system draws file(s)/folder(s).

The Folder Dialog works the same, it just returns directories instead of files and the system does not draw any files. You may see the script's comments for more.

If you have any question , please email : paraskevlos@yahoo.gr