Ultra Fast Menu Creator

Contenido

1. Brief description:

This Asset allows you to quickly create a game configuration menu. You can configure the options and suboptions very quickly and the script is in charge of managing the navigation by keys or with the mouse and saving the values of the options. Each time the *Back button* is pressed, preferences and changes are saved in the *PlayerPrefs*.

To change this menu you just have to modify the prefabs to your liking, modifying the background images, the color or the font and add in your main script the methods that the *PlayerPrefs* that the menu saves read.

I will explain later how to access this information.

Important! the menu does not support localization to other languages right now but I am working on it.

To use the menu, only these lines of code are needed in your scene controller. In this way, simply by pressing the *Escape* key the menu will appear:

```
public class Example : MonoBehaviour
{

    [SerializeField]
    MenuController myMenuController;

    // Update is called once per frame
    void Update()
    {
        if(Input.GetKeyDown(KeyCode.Escape))
            myMenuController.ShowMenu();
    }
}
```

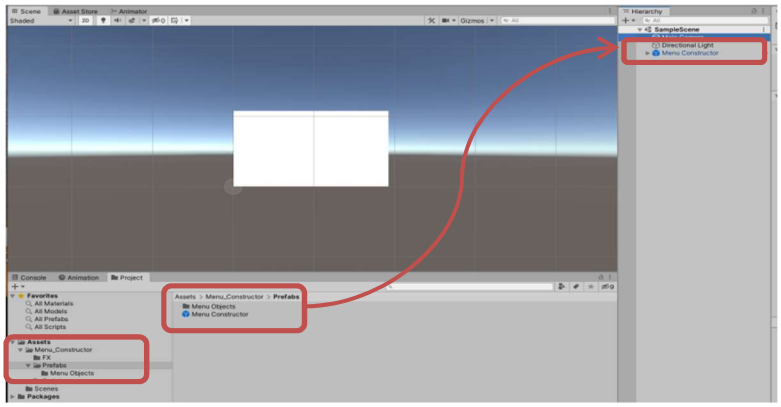
To access the strings that are defined in the *PlayerPrefs*, you just have to call to this method.

```
public string[] GetPlayerPrefsStrings(){
    return sPlayerPrefsStrings;
}
```

And finally, you can reset the *PlayerPrefs* by calling this method:

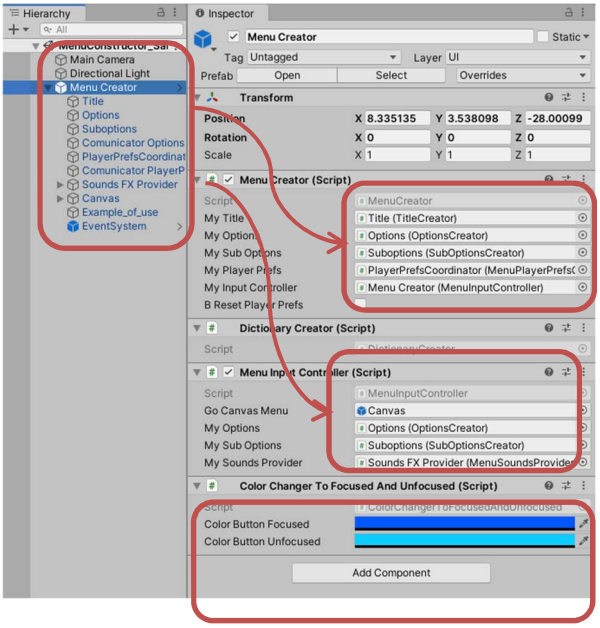
```
public void ResetMenuPlayerPrefs(){
    Debug.Log("Deleting PlayerPrefs");
    PlayerPrefs.DeleteKey(sPlayerPrefsFirstTimeSaved);
    for(int i = 0; i < aSubOptions.Length; i++){
        PlayerPrefs.DeleteKey(sPlayerPrefsStrings[i]);
    }
}
```

2. **Add a menu to the project:** It is as simple as taking the *Menu Creator prefab* located in the *Menu_Creator / Prefabs* folder and dragging it to the project. As simple as that.



The prefab is already made with a couple of options and suboptions, related to sound and graphics, so it is very easy to modify it or build your own.

3. General configuration:

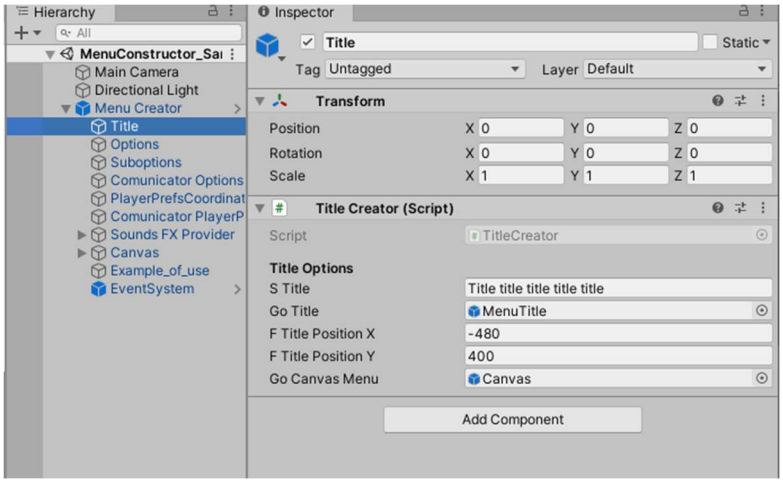


Warning! Never remove prefabs from variables as this will cause the script to stop working correctly. If you want to modify something of the prefabs it is as simple as editing the prefab and changing the parameters you want. I'll talk about this in section 9.

In the Menu Creator Game Object you will find some scripts with their references and you can change the color of the buttons when they are selected and when they are not. In this case the colors are two different blues.

You can also change the audios that sound when you click. In the case of the example I have used the same audio with a different name so that it is easy to understand. Simply by changing the sound in its audio source you can customize your own click sounds.

4. Title settings:



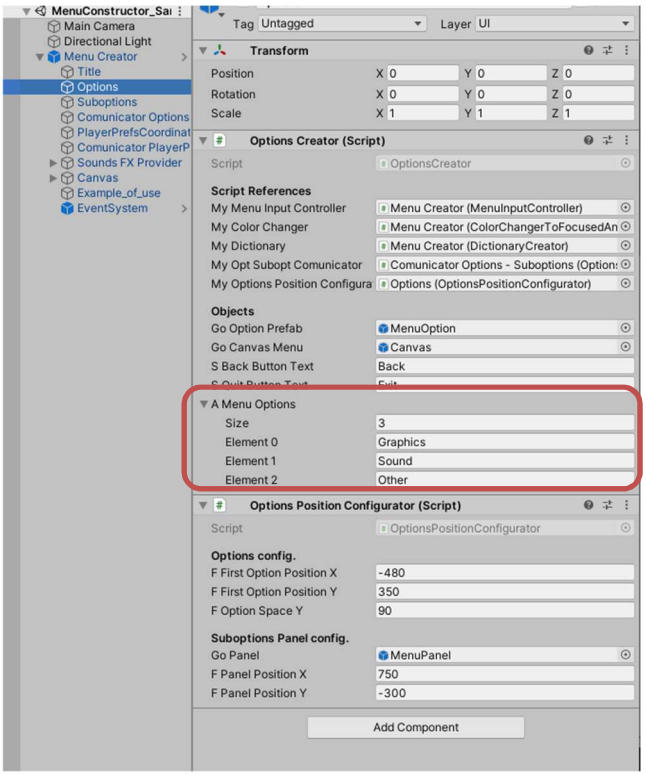
In the *Title Creator* section you can configure the name of the game and the position it will have on the screen.

Again, I want to remember that you do not have to delete the *GoMenu* prefab because it can cause the menu not to work well.

However, you can change the settings of the prefab (background image, typography, etc.) by opening and editing it.

The title position can also be adjusted by changing the coordinates.

5. Option Settings:



MenuOption is the prefab of the options. We recommend not deleting this field for the menu to work properly. If you want to modify the font you can edit the prefab but never delete it.

In addition, the position at which the options start, and the space between them, can be configured.

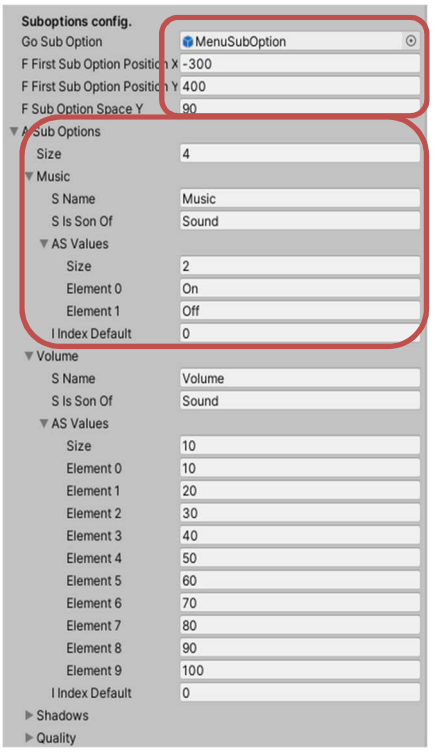
Finally, the array of options must be filled. In the case of the image, 2 options have been included: *Sound*, *Graphics and Other*. Each of these options will create a new *GameObject* and the children related to it will be created, which we will define in the sub options configuration, in the section 6.

It is very important to write the options well because the same text string has to be used to create the sub options.

For example: if we define the option *Graphics* and later use *Graphic*, the sub options will not be grouped in the correct place.

6. Sub Option Settings:

This section is the most complex but even so, it can be completed in a few minutes.

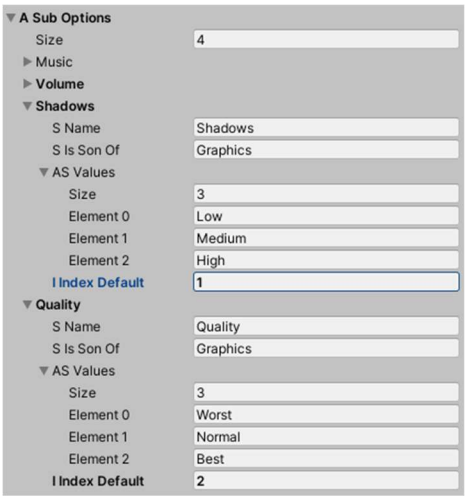


First, the number of total suboptions is defined. Suboptions always have the same structure:

- *sName*: the name of the suboption. In this case, his name is *Music*.
- *sIsSonOf*: is the name of the parent option. It has to match exactly one of the options defined in point 5. In this case, *Music* is a son sub option of *Sound.*
- *aSValues*: is the array of values that this suboption can have. In this case the values are *On* and *Off.*
- *iIndexDefault*: is the value of the default sub option that we want to define. In the image, the default value is 0, which corresponds to element 0 of *aSValues*, that is: *On*.
  It is important to note that this value corresponds to the positions of the array, so its first value will be 0 and the last value will be the size of the array minus one. In this case, from 0 to 1

The next sub option is *Volume* and it is also a child of the *Sound* option. In this case it has 10 values that go from 10 to 100 and the default option is 0 (Reminder: this value can go from 0 to 9, the size of the array minus 1).

Finally, the following image shows by way of example how the other two Shadows options are configured*:*



- Son of *Graphics*
- It has 3 values: *Low*, *Medium* y *High*
- Default value: *0*.

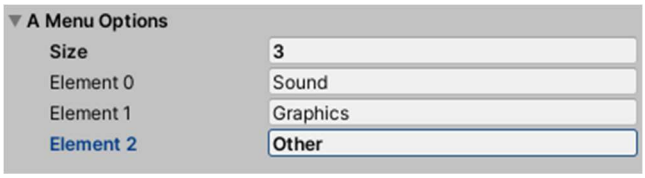
Quality:

- Name: *Quality*.
- Son of *Graphics*
- It has 3 possible values: *Worst*, *Normal*, *Best*

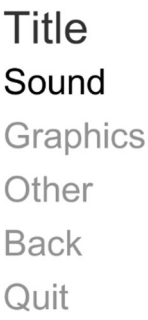
- Default index is 2: *Best*

7. Add a new option

It is very simple. Simply, increase the size of the *AMenuOptions* array. In the image you can see how now the size is 3 and the new option is called *Other*

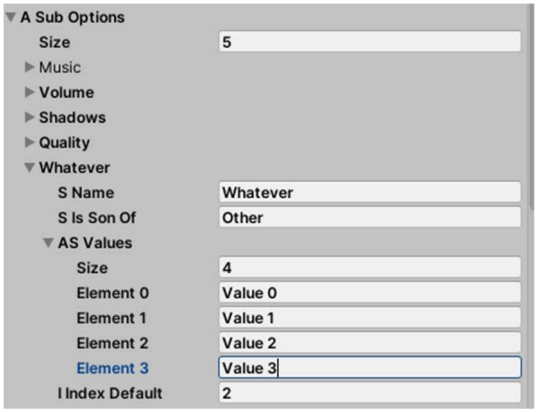


The menu now looks like this:

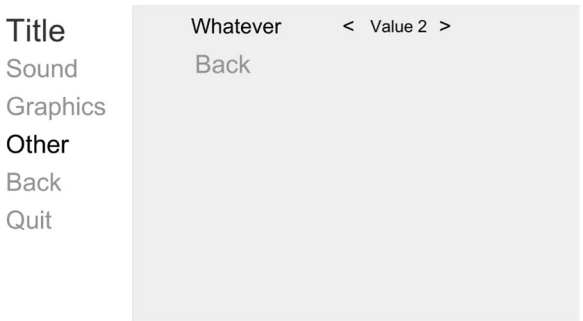


8. Add a new suboption:

In this case, we increase the size of the *ASubOptions* array and we will see that a new sub option appears. Fill in the name, whose son is, in this case, the *Other* option. It will have the 4 values that are seen on the screen and the default index will be 2 (that is, the value *Value* 2)

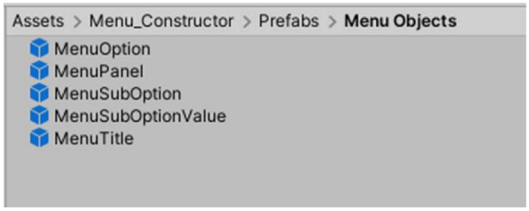


The result is as follows:

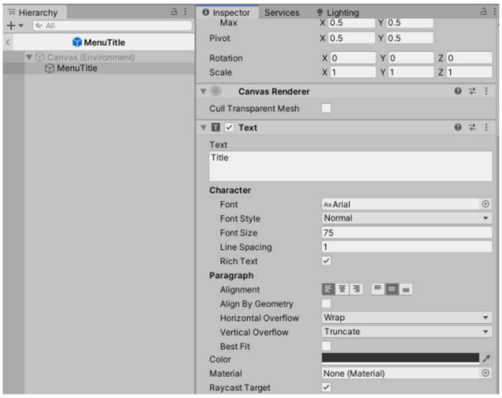


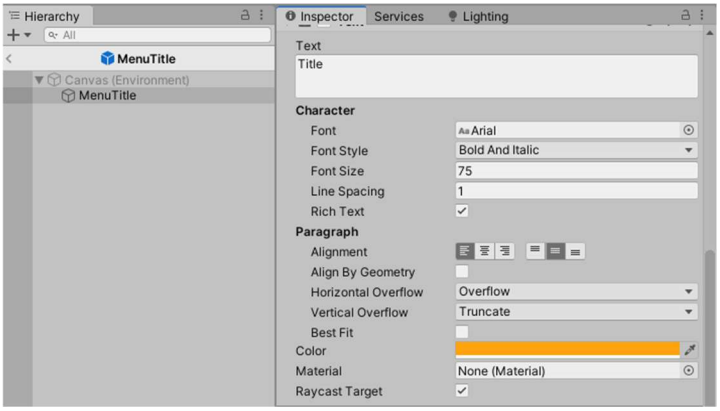
9. Modify the prefabs:

If we want to change the appearance of the menu we can modify the values of the prefabs. These are located in the *Assets folder> Menu_Creator> Prefabs> Menu Objects*

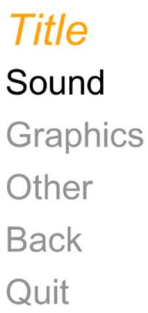


If we select the *MenuTitle* prefab we can access its components. In this case we are going to change the font color and style.

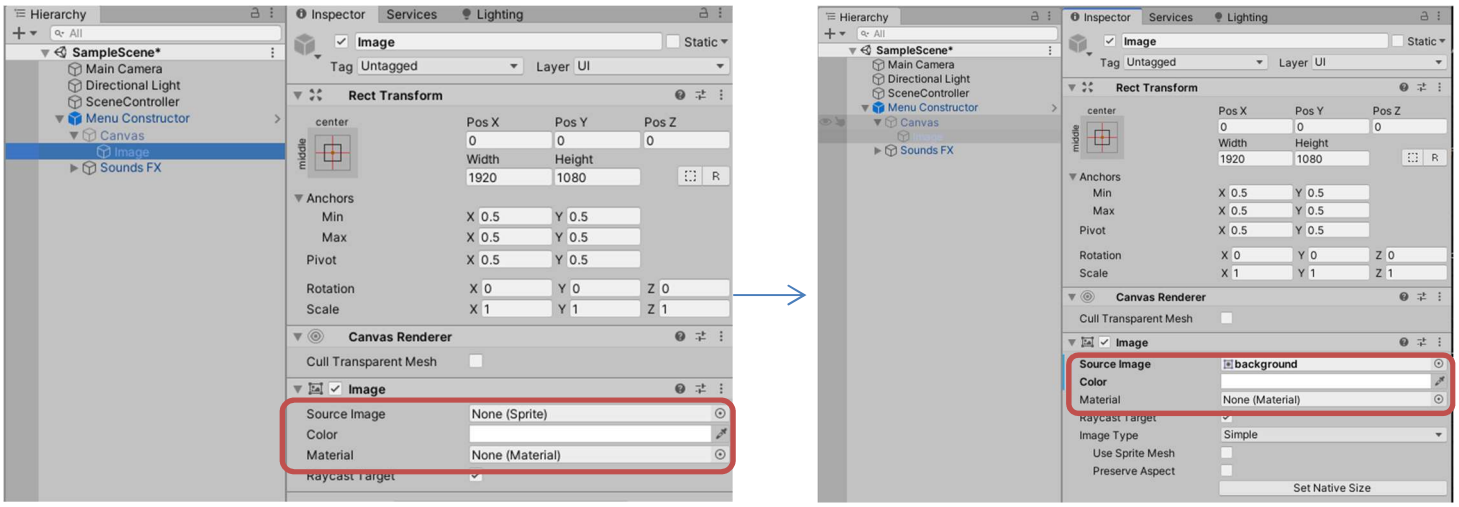




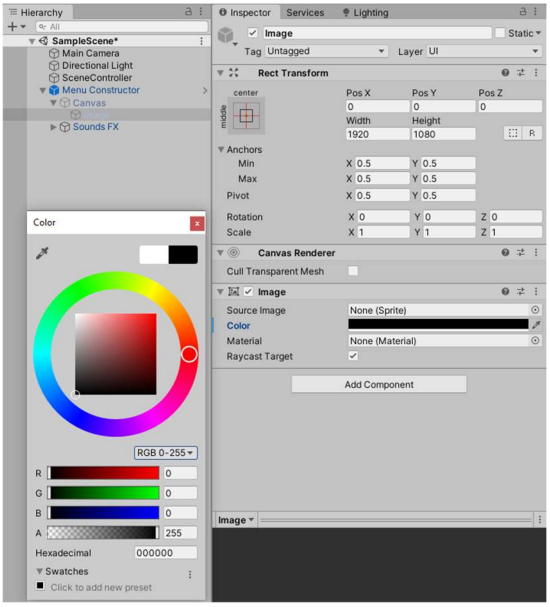
The result is as follows:



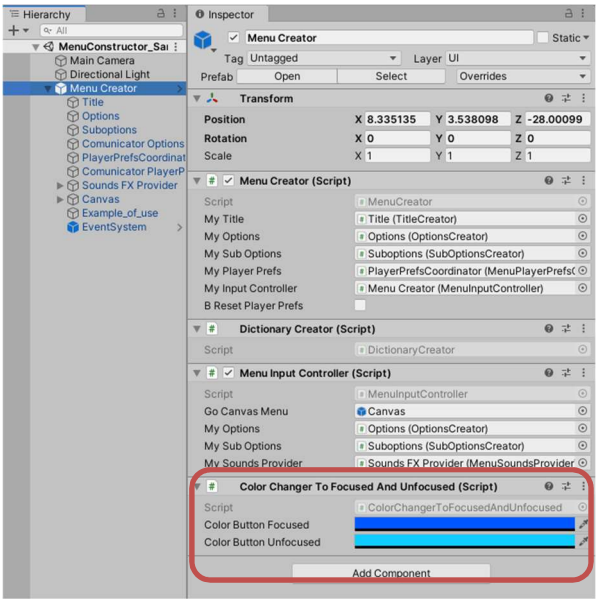
Next we are going to add a background:



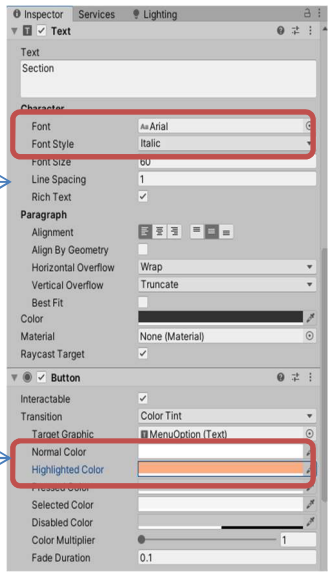
The *Canvas* object has an *Image* child that allows you to change the background color or add an image. You just have to modify these values with the ones that best suit our game. In our case we are going to change the background color to black.



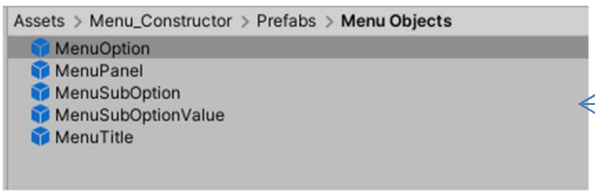
Also, we are going to modify the color of the buttons when they are in focus and when they are out of focus. To do this, within the *Menu Creator* object we change the colors as seen in the image.



Note that the *Coior Button Focused* and *Color Button Unfocused* are defined in the *Menu Controller*, not in the button properties. However, the color to highlight the button when the mouse is hovered over it, is defined in these properties.



And finally, we are going to change the font style and color when the mouse passes over it, in the options and sub options by opening the *MenuOption* and M*enuSubOption* prefabs, located in *Assets> Menu_Creator> Prefabs> Menu Objects.*



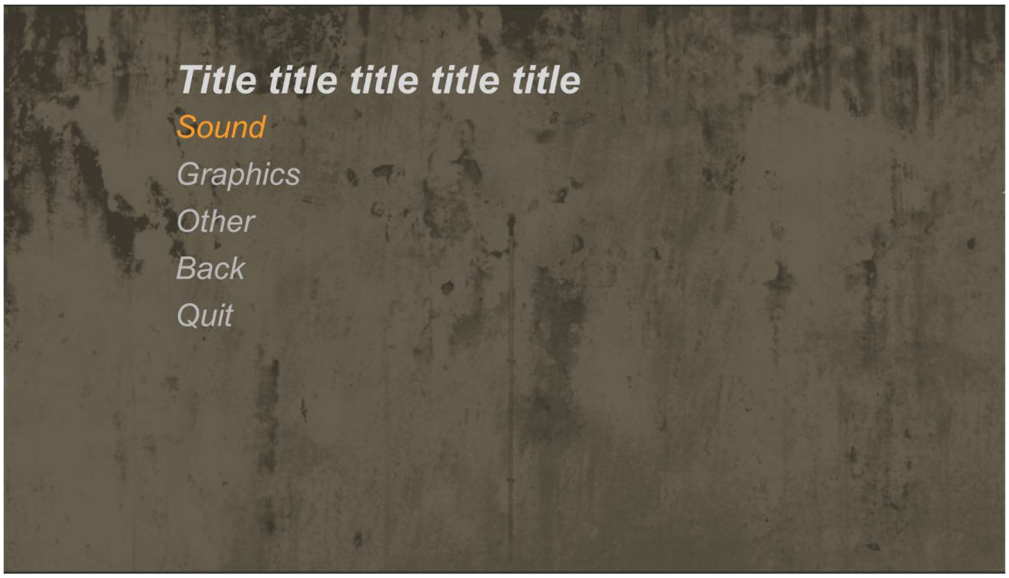
The other characteristics of the *Text* and *Button* components could also be changed.

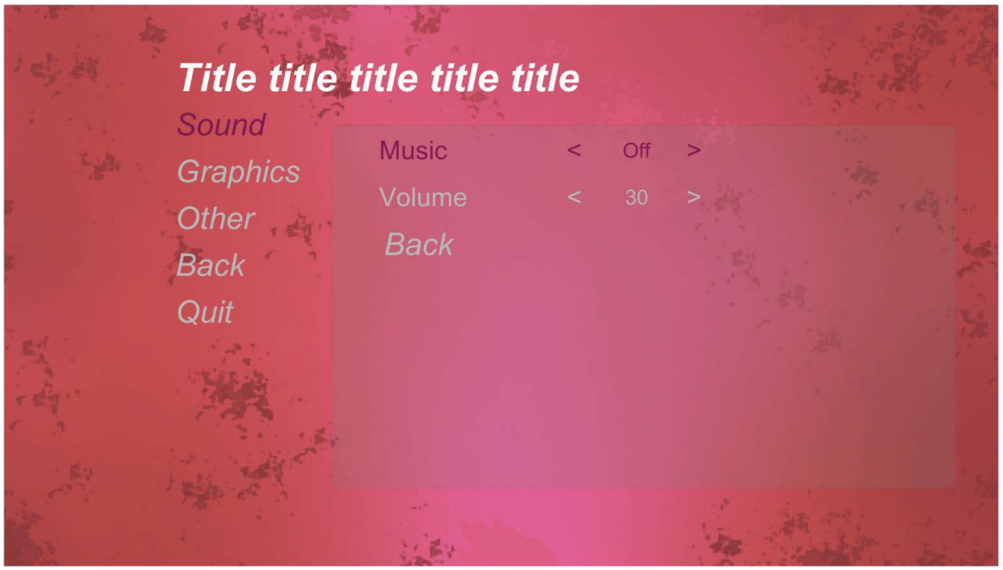
In the same way, if we edit the *MenuSubOption* prefab, you can change the font, size, style and color when you mouse over the sub options.

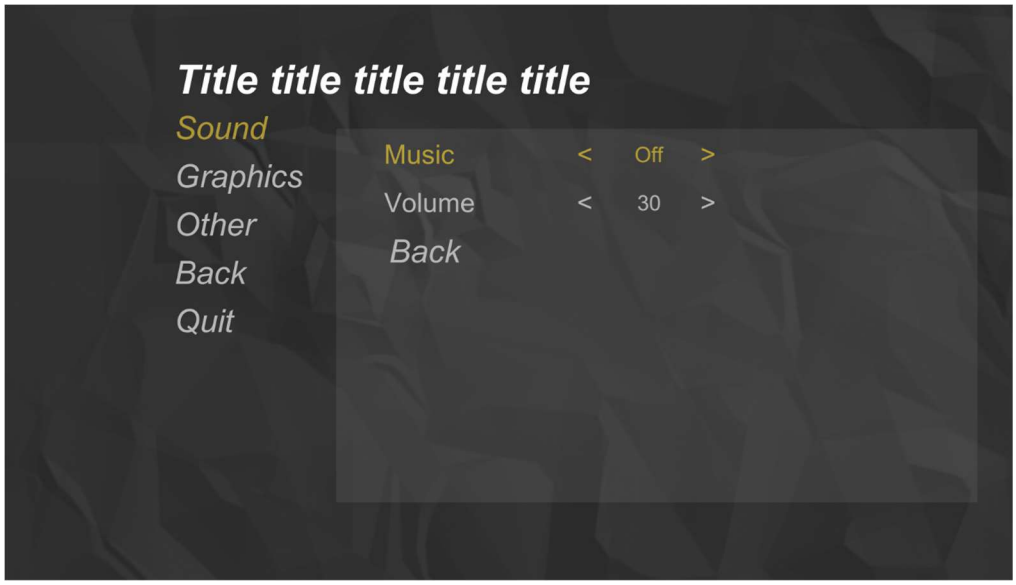
And finally, the *MenuSubOptionValue* prefab corresponds to the list of values that will appear in each sub option and in the symbols "<" and ">".

10. Examples

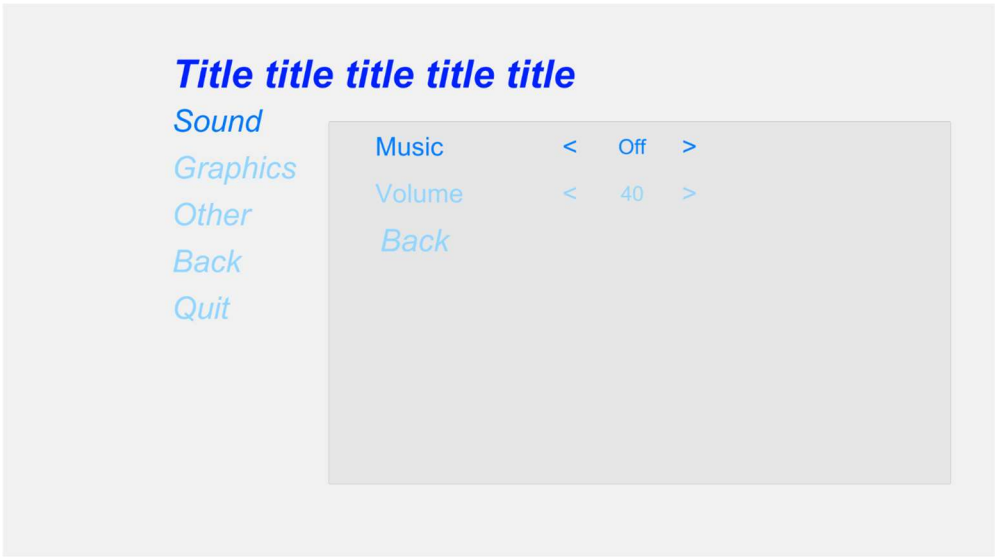
These are some examples, with a background image, that can be done in no time (sorry, I'm not an artist):

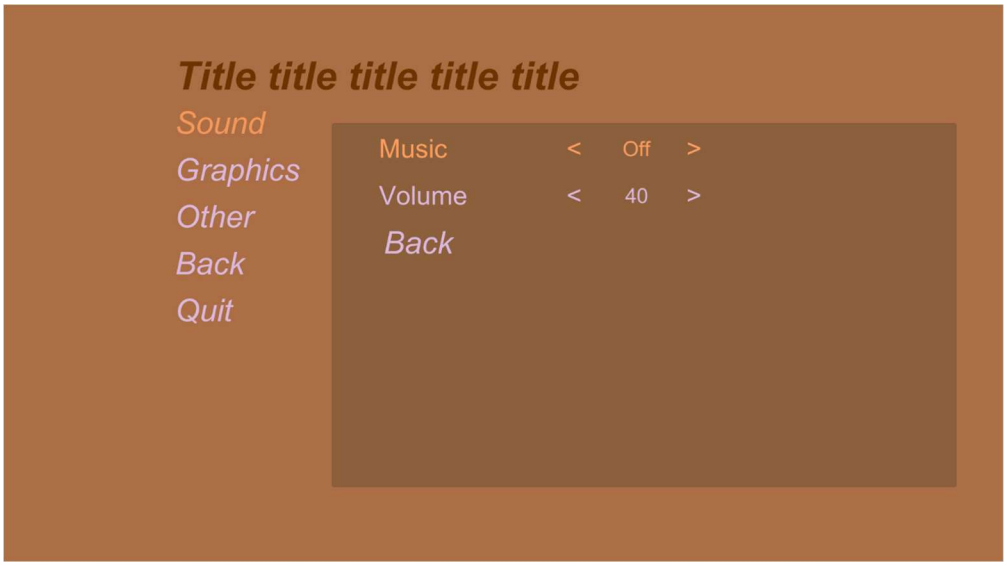


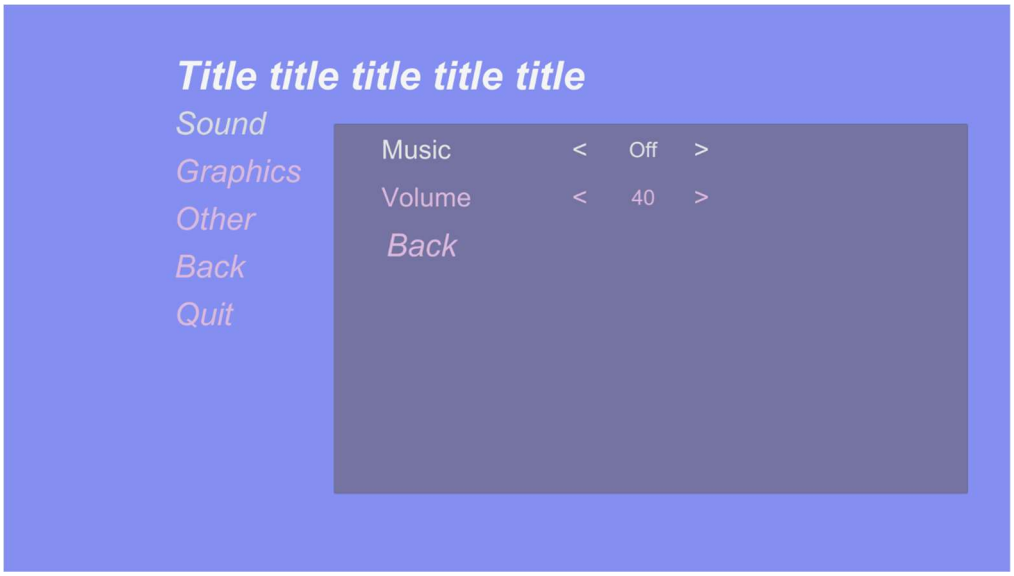




And these other examples have no background image:








Title title title title title
Sound
Graphics
Other
Back
Quit
Music < Off >
Volume < 40 >
Back