

Ultra Fast Menu Creator

Contenido

1.	Descripción breve:	3
2.	Añadir un menú al proyecto:	3
3.	Configuración general:	4
4.	Configuración del título:	4
5.	Configuración de las opciones:	4
6.	Configuración de las subopciones:.....	5
7.	Añadir una nueva opción	5
8.	Añadir una nueva subopción:	6
9.	Modificar los prefabs:	7
10.	Ejemplos.....	10

1. Descripción breve:

Este Asset permite crear rápidamente un menú de configuración del juego. Se pueden configurar las opciones y subopciones de forma muy rápida y el script se encarga de gestionar la navegación por teclas o con el ratón y el guardado de los valores de las opciones. Cada vez que se pulsa el botón *Back* se guardan las preferencias y los cambios en los *PlayerPrefs*.

Para cambiar el menú tan sólo tienes que modificar los prefabs a tu gusto, modificando las imágenes de fondo, el color o la tipografía y añadir en tu script principal los métodos que lean los *PlayerPrefs* que guarda el menú.

Más adelante explicaré cómo acceder a esta información.

¡Importante! Ahora mismo el menú no soporta la localización a otros idiomas pero estoy trabajando en ello.

Para usar el menú sólo hacen falta estas líneas de código en tu controlador de escena. De esta forma, simplemente pulsando la tecla *Escape* aparecerá el menú:

```
public class Example : MonoBehaviour
{
    [SerializeField]
    MenuController myMenuController;

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

Para acceder a las cadenas que se definen en los *PlayerPrefs* sólo hay que referenciar a este método.

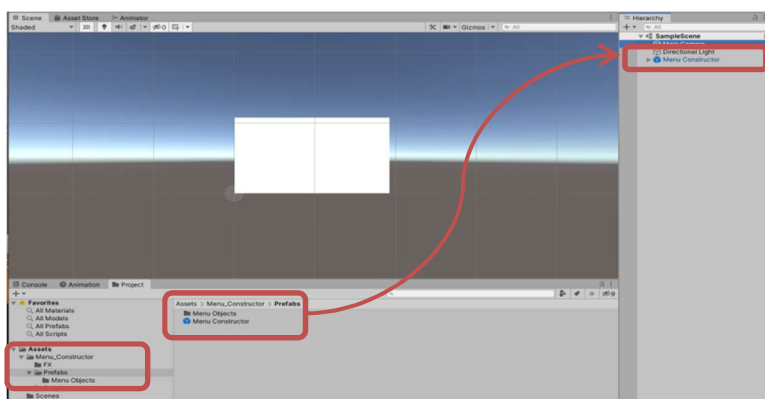
```
public string[] GetPlayerPrefsStrings(){
    return sPlayerPrefsStrings;
}
```

Y por último, se puede resetear los *PlayerPrefs* llamando este método:

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

2. Añadir un menú al proyecto:

Es tan sencillo como coger el prefab **Menu Creator** situado en la carpeta *Menu_Creator/Prefabs* y arrastrarlo al proyecto. Tan sencillo como eso.



El prefab está ya hecho con un par de opciones y subopciones, relacionadas con el sonido y los gráficos, para que sea muy sencillo modificarlo o construir el tuyo propio.

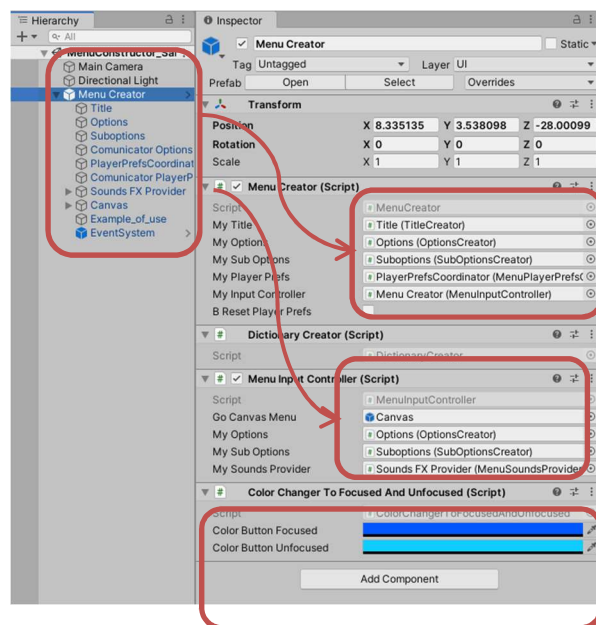
3. Configuración general:

Al seleccionar el game object Menu Creator en la jerarquía, veréis que aparecen los campos de configuración en el inspector.

Warning! No quitéis nunca los prefabs de las variables ya que eso producirá que el script deje de funcionar correctamente. Si queréis modificar algo de los prefabs es tan sencillo como editar un prefab y cambiar los parámetros que queráis.

En la objeto Menu Creator encontraréis diferentes scripts con sus referencias y además, se puede cambiar el color de los botones cuando están seleccionados y cuando no lo están. En este caso los colores tienen dos tonos de azul.

Además, se pueden cambiar los audios que suenan al hacer click. En el caso del ejemplo he usado el mismo audio con diferente nombre para que sea fácil de entender. Simplemente cambiando el sonido en su audio source podréis personalizar vuestros propios sonidos de click.



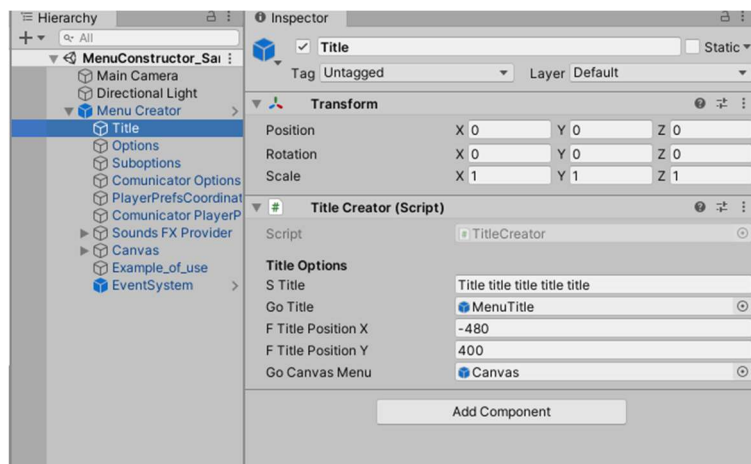
4. Configuración del título:

En la sección *Title Creator* se puede configurar el nombre del juego y la posición que tendrá en la pantalla.

De nuevo quiero recordar que no hay que borrar las referencias a los prefabs porque puede ocasionar que el menú no funcione bien.

Sin embargo, sí se puede cambiar la configuración del prefab (imagen de fondo, tipografía, etc.) abriéndolo y editándolo. Más adelante explicaremos esto.

La posición del título también se puede ajustar cambiando las coordenadas.

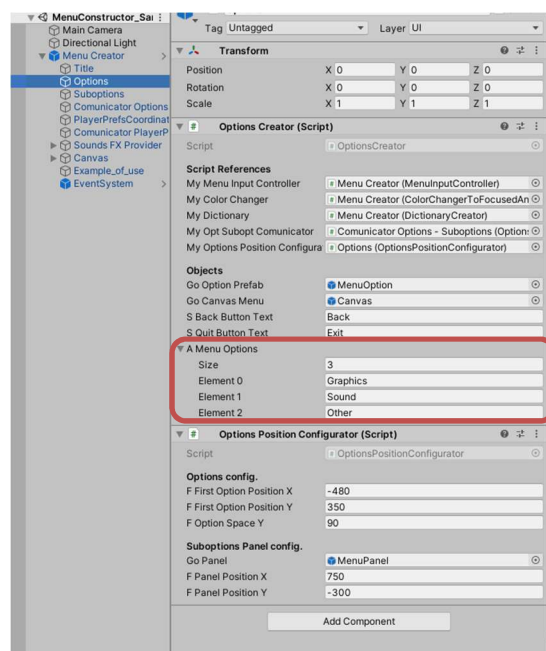


5. Configuración de las opciones:

MenuOption es el prefab de las opciones. Recomendamos no borrar este campo para que el menú funcione correctamente. Si se quiere modificar la tipografía se puede editar el prefab pero nunca borrarlo.

Además se puede configurar la posición en la que comienzan las opciones y el espacio entre ellas.

Por último, hay que rellenar el array de opciones. En el caso de la imagen, se han incluido 2 opciones: *Sound*, *Graphics* y *Other*. Cada una de estas opciones creará un nuevo *GameObject* y se crearán los hijos relacionados con él, que definiremos en la configuración de las subopciones, en el siguiente punto.



Es muy importante escribir bien las Opciones porque se usará la misma cadena de texto para crear las subopciones.

Por ejemplo: si definimos la opción “Graphics” y posteriormente usamos “Graphic”, las subopciones no se agruparán en el lugar correcto.

6. Configuración de las subopciones:

En primer lugar, se define el número de subopciones totales. Las subopciones tienen siempre la misma estructura

- sName: el nombre de la subopción. En este caso su nombre es *Music*
- sIsSonOf: es el nombre de la opción padre. **Tiene que coincidir exactamente con una de las opciones definidas en el punto 5.** En este caso, *Shadows* es una subopción hija de la opción *Graphics*.
- aSValues: es el array de valores que podrá tener esta subopción. En este caso los valores son *Low*, *Medium*, *High* y *Ultra*.
- iIndexDefault: es el valor de la subopción por defecto que queremos definir. En la imagen, el valor por defecto es 2, lo que corresponde al elemento 2 de *ASValues*, es decir: *High*.

Es importante tener en cuenta que este valor corresponde a las posiciones del array por lo que su primer valor será 0 y el último, el tamaño del array menos uno. En este caso, de 0 a 1.

La siguiente subopción es *Volume* y también es hija de la opción *Sound*. En este caso tiene 10 valores que van de 10 a 100 y la opción por defecto es el 0 (Recordatorio: este valor puede ir de 0 a 9, el tamaño del array menos 1).

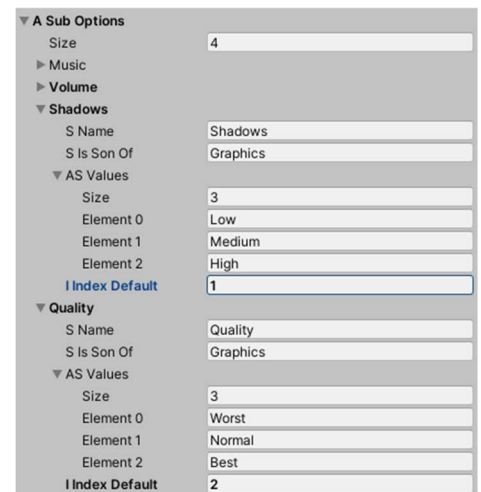
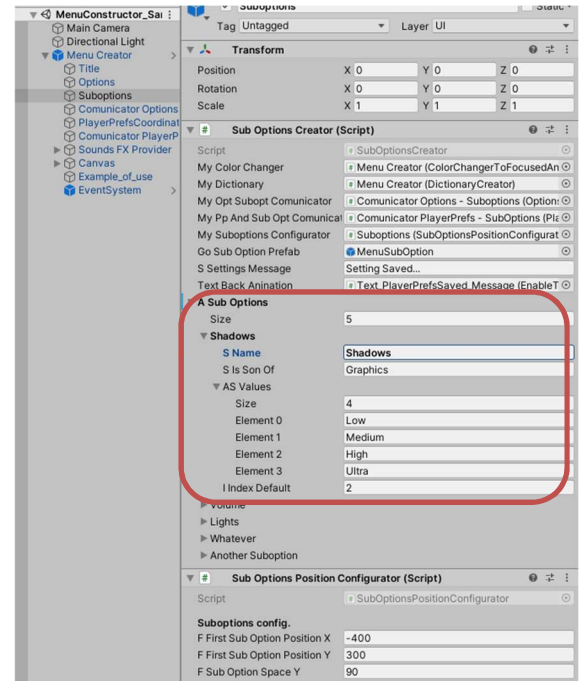
Por último, en la siguiente imagen se muestra a modo de ejemplo cómo se pueden configurar o cambiar rápidamente otras opciones

Shadows:

- hija de *Graphics*
- con 3 valores: *Low*, *Medium* y *High*
- Valor por defecto: 0.

Quality:

- Nombre: *Quality*.
- Es hija de *Graphics*
- Tiene 3 valores posibles: *Worst*, *Normal*, *Best*
- El índice del valor por defecto es 2, es decir, *Best*



7. Añadir una nueva opción

Es muy sencillo. Simplemente hay que aumentar el tamaño del array *AMenuOptions*. En la imagen podemos ver como ahora el tamaño es 3 y la opción nueva se llama *Other*

▼ A Menu Options	
Size	3
Element 0	Sound
Element 1	Graphics
Element 2	Other

El menú ahora se ve así:

Title
Sound
Graphics
Other
Back
Quit

8. Añadir una nueva subopción:

En este caso aumentamos el tamaño del array *ASubOptions* y veremos que aparece una nueva subopción. Rellenamos el nombre, de quién es hija (en este caso de la opción *Other*). Tendrá los 4 valores que se ven en pantalla y el índice por defecto será 2 (es decir, el valor *Value 2*)

▼ A Sub Options	
Size	5
▶ Music	
▶ Volume	
▶ Shadows	
▶ Quality	
▼ Whatever	
S Name	Whatever
S Is Son Of	Other
▼ AS Values	
Size	4
Element 0	Value 0
Element 1	Value 1
Element 2	Value 2
Element 3	Value 3
I Index Default	2

El resultado es el siguiente:

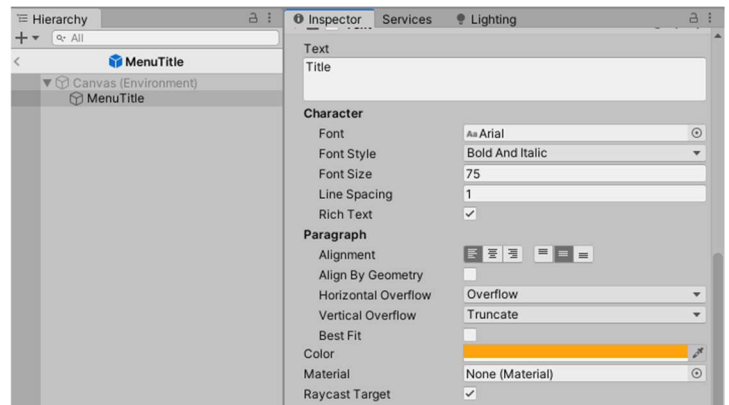
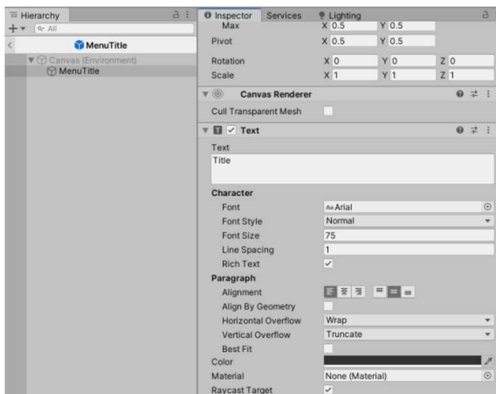
Title	Whatever	< Value 2 >
Sound	Back	
Graphics		
Other		
Back		
Quit		

9. Modificar los prefabs:

Si queremos cambiar el aspecto del menú podemos modificar los valores de los prefabs. Estos se encuentran en la carpeta *Assets>Menu_Creator>Prefabs>Menu Objects*



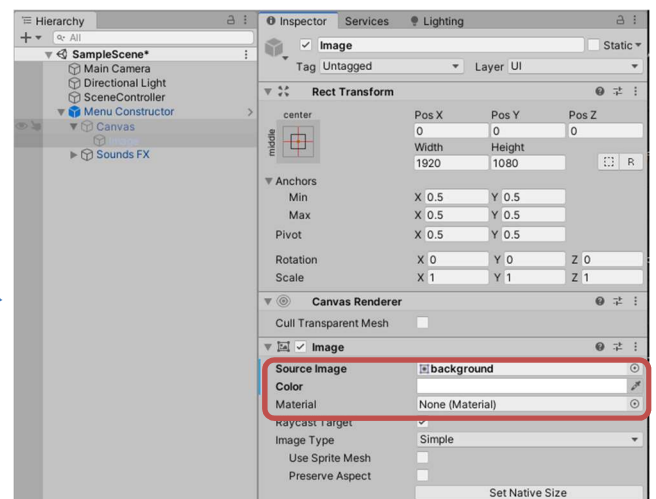
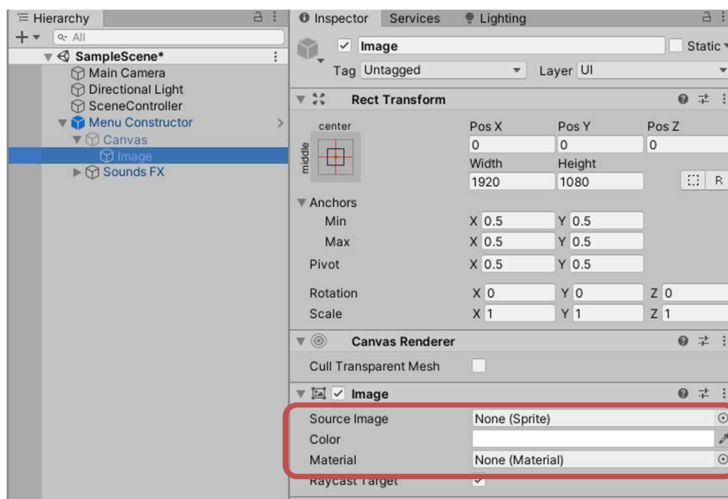
Si seleccionamos el prefab *MenuTitle* podemos acceder a sus componentes. En este caso vamos a cambiar el color de la fuente y el estilo.



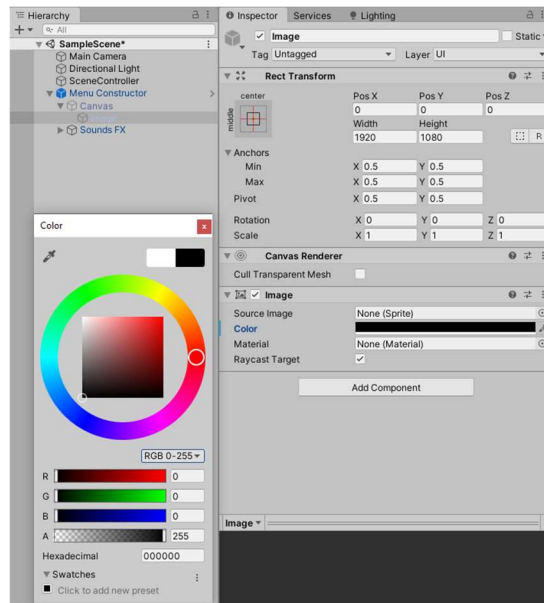
El resultado es el siguiente:

Title
Sound
Graphics
Other
Back
Quit

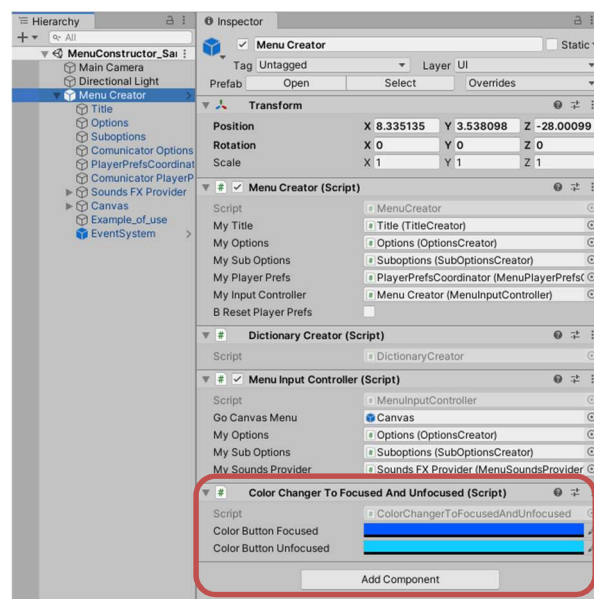
A continuación vamos a añadir un fondo:



El objeto *Canvas* tiene un hijo *Image* que permite cambiar el color de fondo o añadir una imagen. Simplemente hay que modificar estos valores con los que mejor se adapten a nuestro juego. En nuestro caso vamos a cambiar el color de fondo a negro.

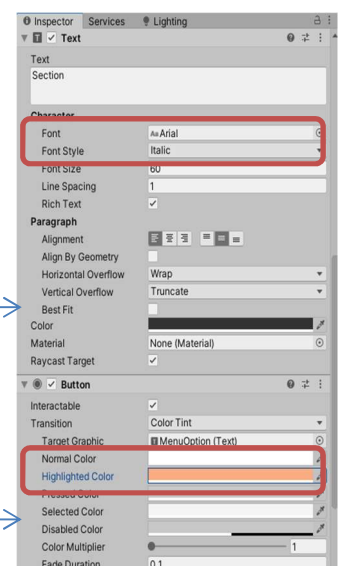
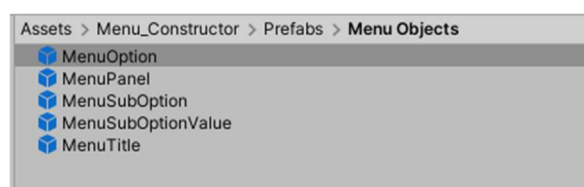


Además, vamos a modificar el color de los botones cuando están en el foco y cuando están fuera de foco. Para ello, dentro del objeto *Menu Creator* cambiamos los colores como se ve en la imagen.



Hay que tener en cuenta que el *Color Button Focused* y *Color Button Unfocused* se define en el *Menu Controller*, no en las propiedades del botón. Sin embargo, el color para que se resalte el botón cuando se pasa el ratón por encima, sí que se define en esas propiedades.

Y por último, vamos a cambiar el estilo de letra y el color cuando el ratón pasa por encima, en las opciones y subopciones abriendo los prefabs *MenuOption* y *MenuSubOption*, situados en *Assets>Menu_Creator>Prefabs>Menu Objects*.



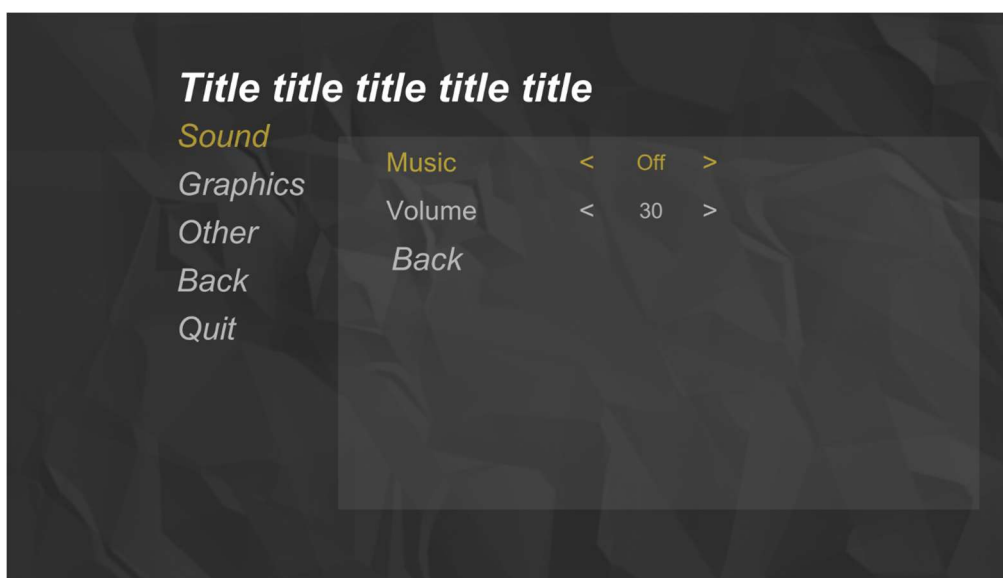
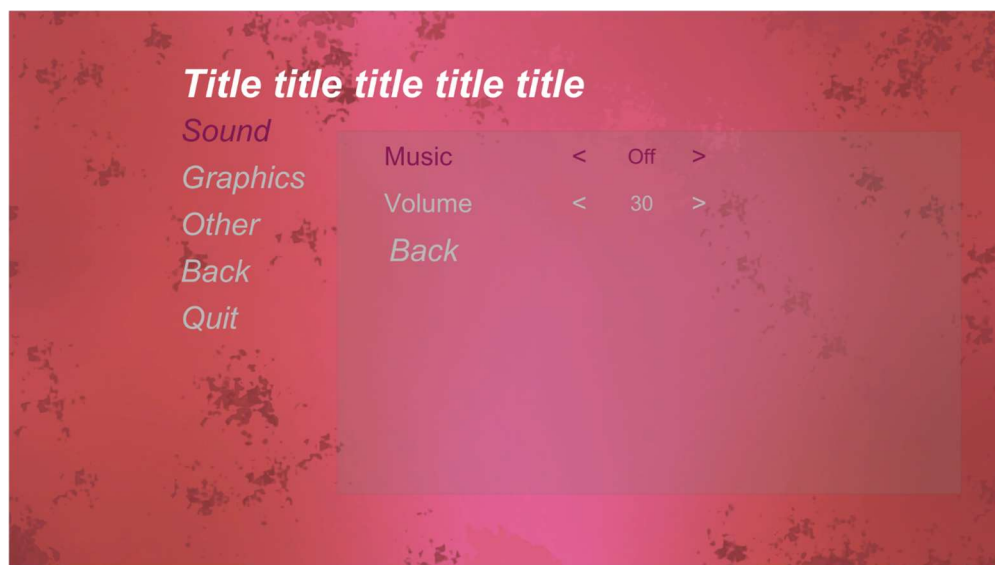
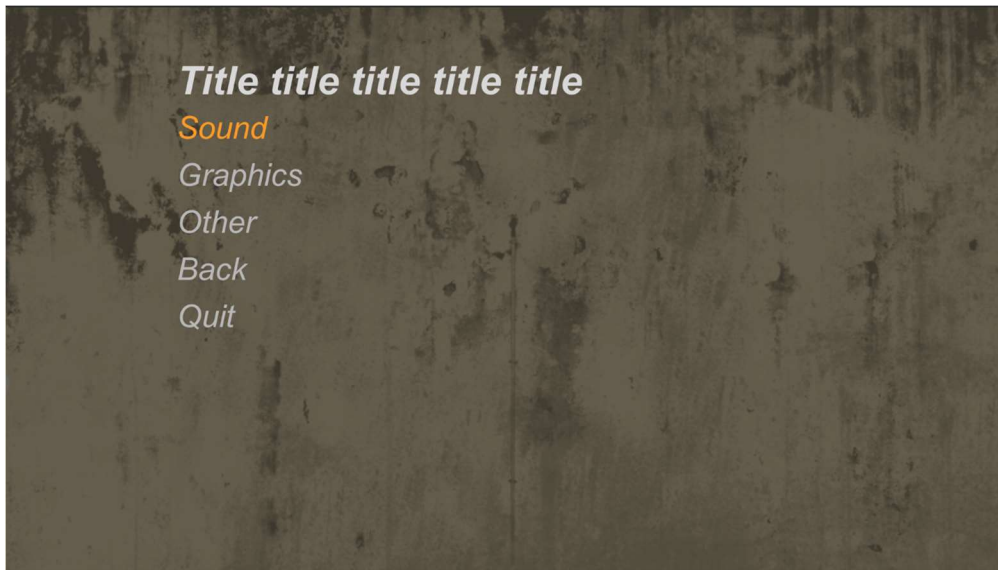
También se podrían cambiar el resto de características de los componentes *Text* y *Button*.

De la misma forma, si editamos el prefab *MenuSubOption* se puede cambiar el tipo de letra, el tamaño, el estilo y el color cuando se pasa el ratón por encima de las subopciones.

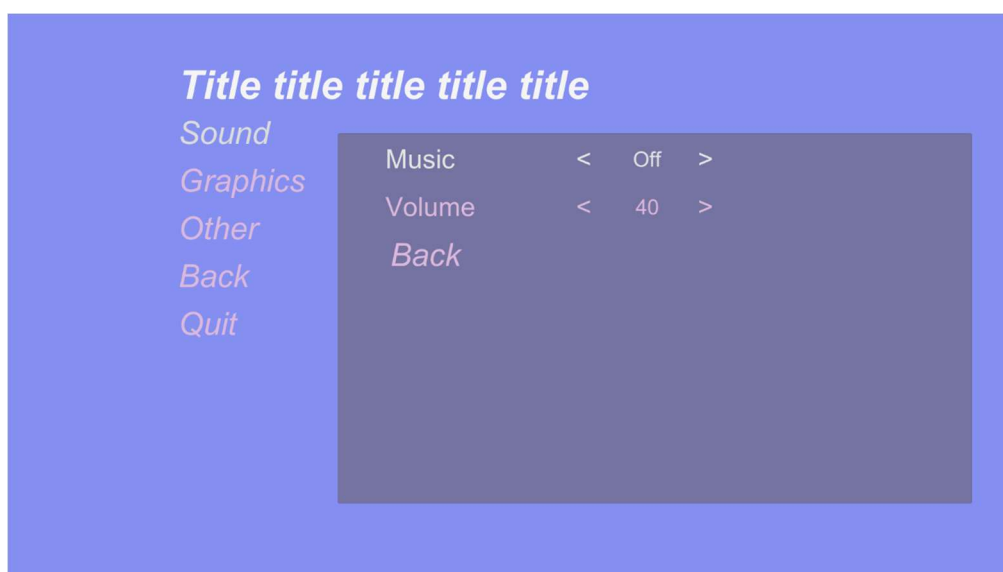
Y por último, el prefab *MenuSubOptionValue* corresponde a la lista de valores que aparecerán en cada subopción y en los símbolos “<” y “>”.

10. Ejemplos

Estos son algunos ejemplos, con una imagen de fondo, que se pueden hacer en muy poco tiempo (lo siento, no soy artista):



Y estos otros ejemplos no tienen ninguna imagen de fondo:



Title title title title title

Sound

Graphics

Other

Back

Quit

Music < Off >

Volume < 40 >

Back