



I.E.S PALOMERAS
VALLECAS

Desarrollo de Interfaces

2º Desarrollo de Aplicaciones Multiplataforma



I.E.S. VILLABLANCA

**UT.3 Introducción a la generación de interfaces gráficas de usuario con editores visuales:
Componentes básicos.**

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Contenidos

En esta Unidad de Trabajo vamos a tratar los siguientes puntos:

- Toggle. *Diapositivas 3-17.*
- DropDown. *Diapositivas 18-27.*
- Raw Image. *Diapositivas 28-38*

Material Adicional a esta presentación:

- Software:
 - UT3_Ejemplo.unitypackage

Prácticas:

- Práctica_UT3_IntroduccionInterfaceGrafica

Vídeos:

- UT3_Slider
- UT3_Toggle_ParteA
- UT3_Toggle_ParteB_Script
- UT3_Dropdown
- UT3_RawImage

Contenidos

Criterios de Evaluación:

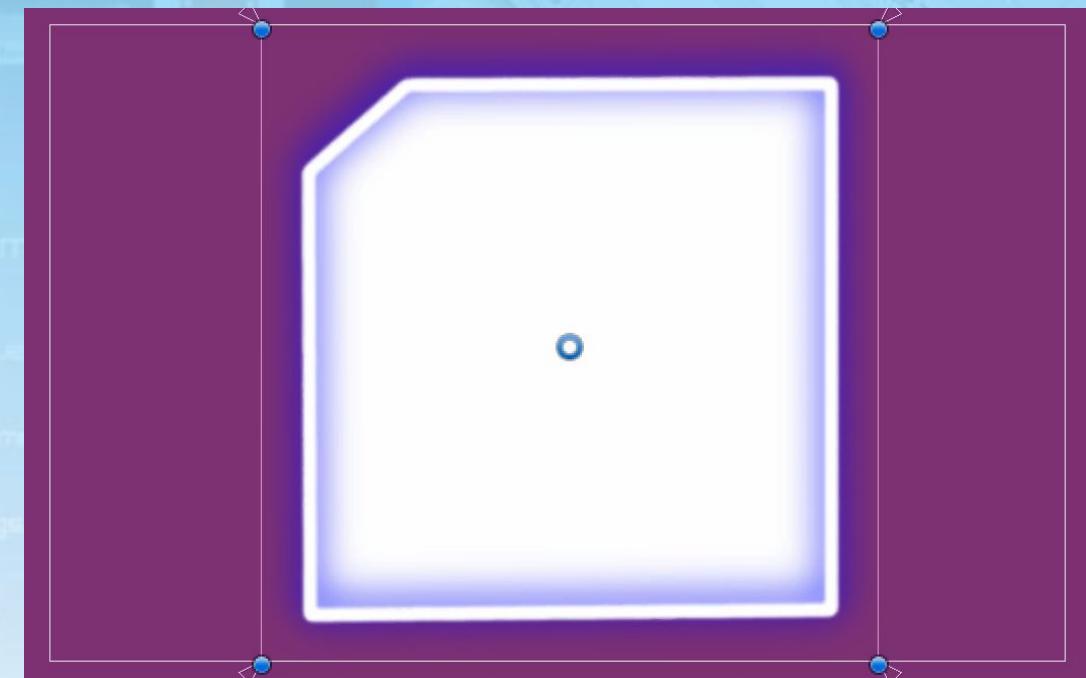
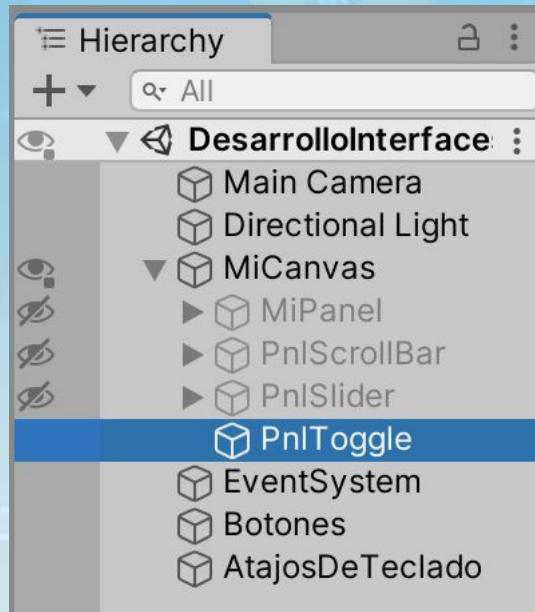
- Se ha creado un interfaz gráfica utilizando los asistentes de un editor visual.
- Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- Se ha analizado el código generado por el editor visual.
- Se ha modificado el código generado por el editor visual.
- Se han asociado a los eventos las acciones correspondientes.
- Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

Esta herramienta la vamos a utilizar para, entre un conjunto de opciones a elegir, seleccionar, ninguna, una o varias.

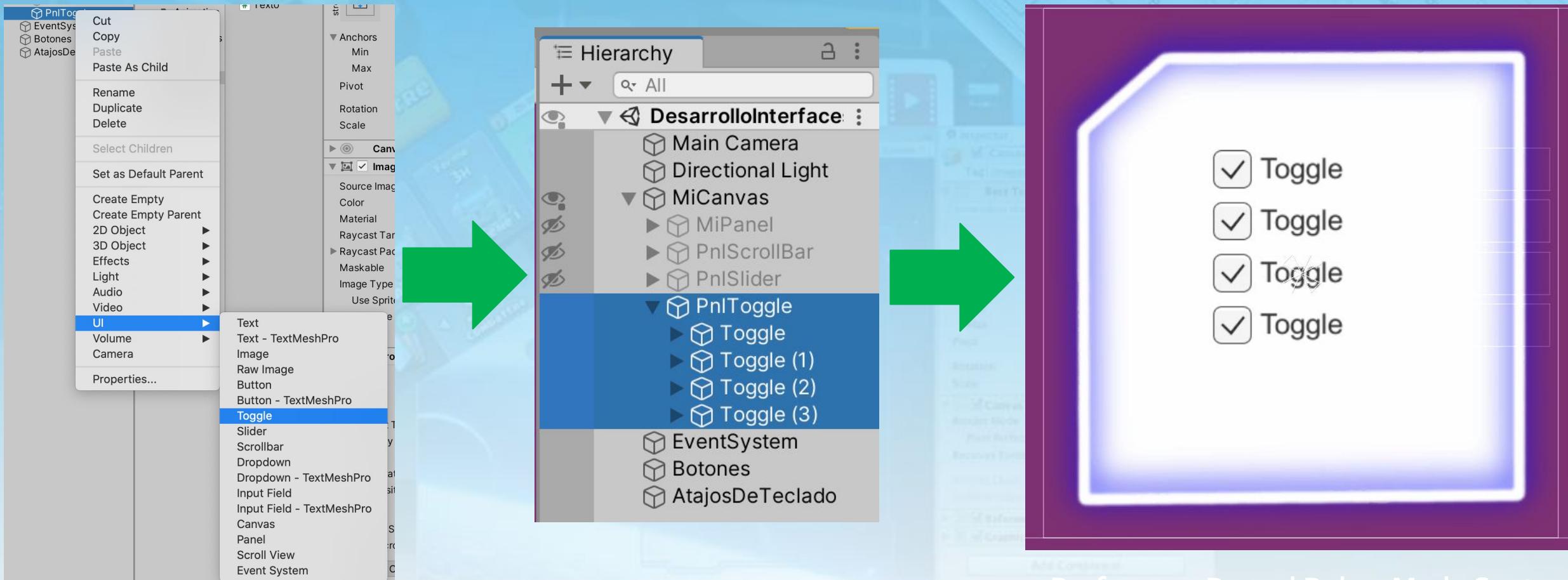
Para ver su funcionamiento, vamos a crear un nuevo panel, dentro de nuestro Canvas, que vamos a denominar “PnlToggle”, como se muestra en la imagen.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

Vamos a añadir, dentro de nuestro Panel “PnlToggle”, cuatro GameObject de tipo “Toggle” y los vamos a situar uno debajo del otro, como muestra la imagen.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

Como podemos observar, el GameObject “Toggle”, tiene dos hijos que, como hemos estado trabajando en estas UT’s, podemos personalizar a través de sus parámetros.



Fijaros que tiene un GameObject hijo denominado “Background” (este es el cuadro de selección), que a su vez tiene otro hijo “Checkmark” (la marca de selección”).

Por último nos encontramos el GameObject hijo “Label”, que es el texto que aparece a la derecha del cuadro de selección.

Vamos a personalizar, para nuestro ejemplo, el Background , el Checkmark y el texto.

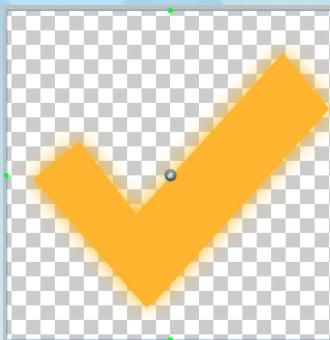
UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

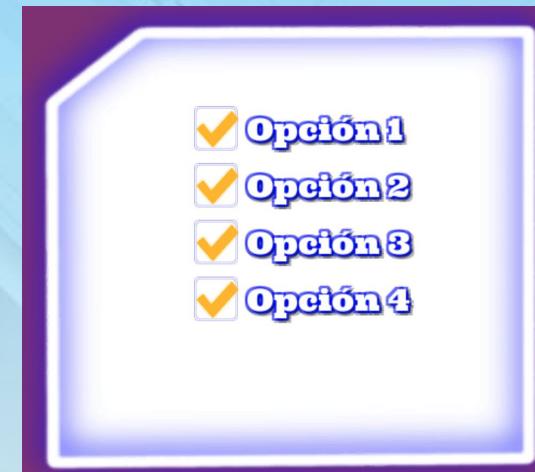
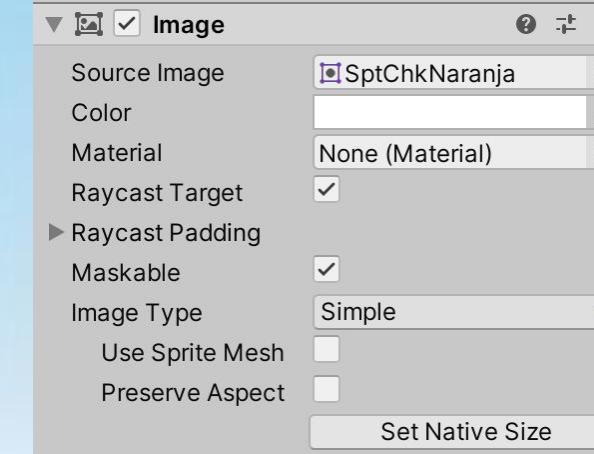
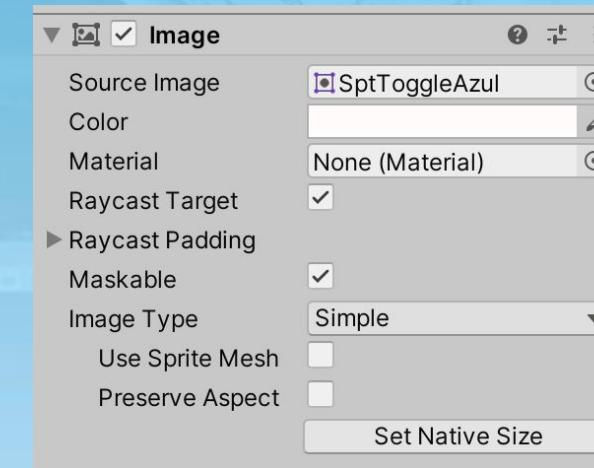
En las imágenes, se muestran los Sprites que se han creado y la ventana Inspector, para que se vea la configuración.



SptToggleAzul



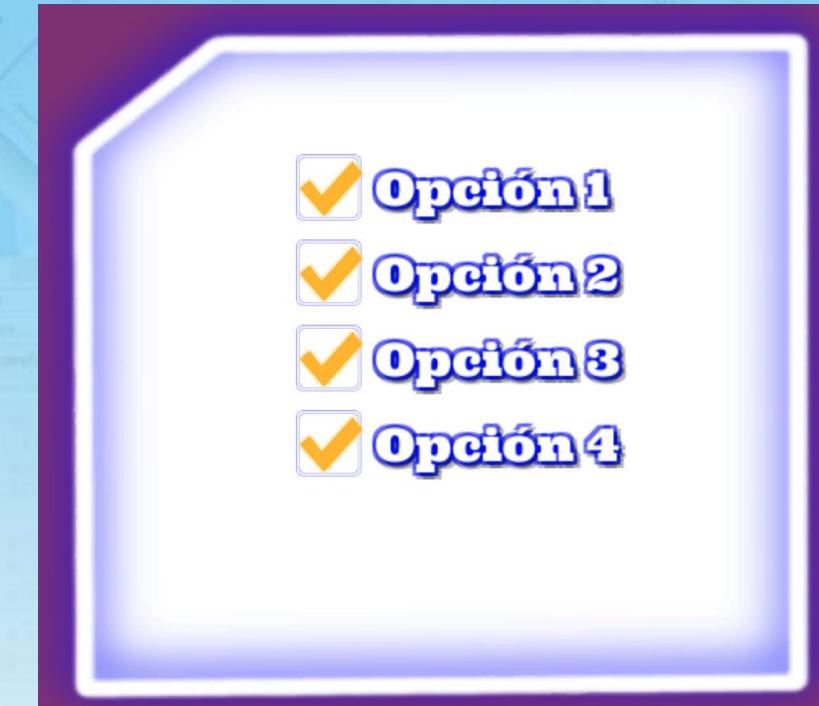
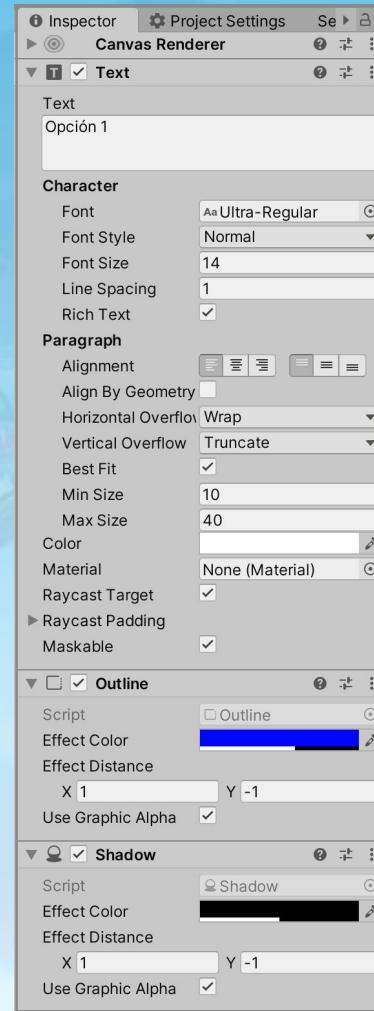
SptChkNaranja



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

Y la configuración para el texto.



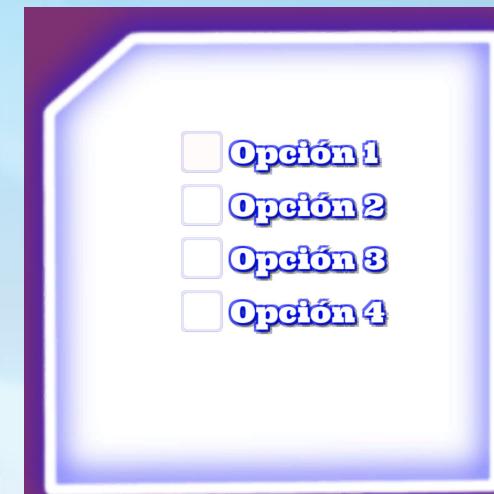
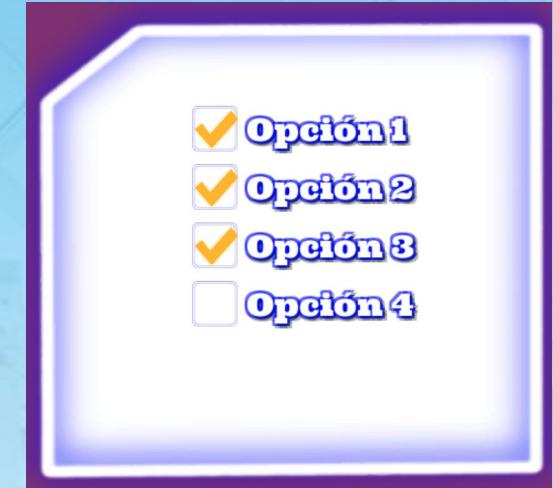
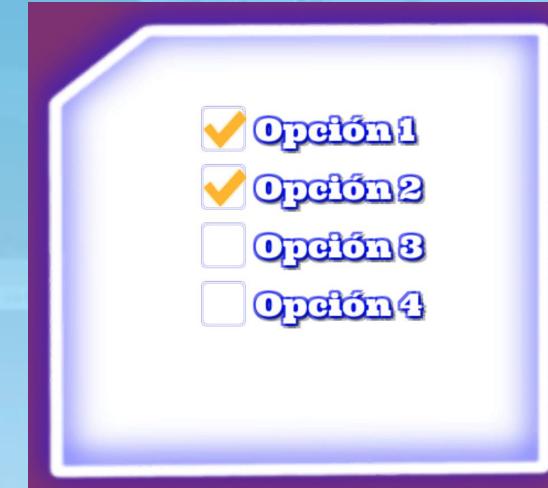
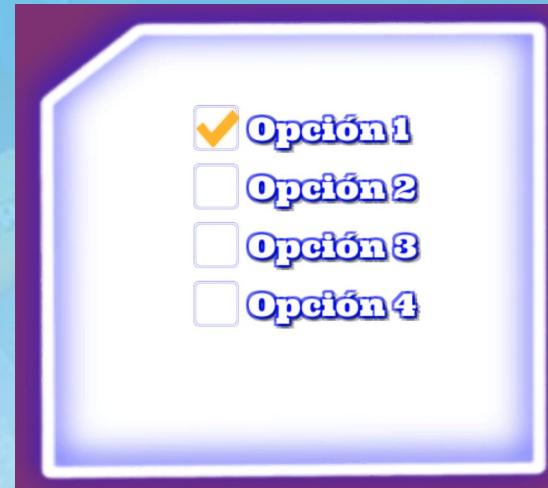
Recordar que todas las texturas van en el paquete del proyecto.

Profesores: Raquel Rojo y Mario Santos.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

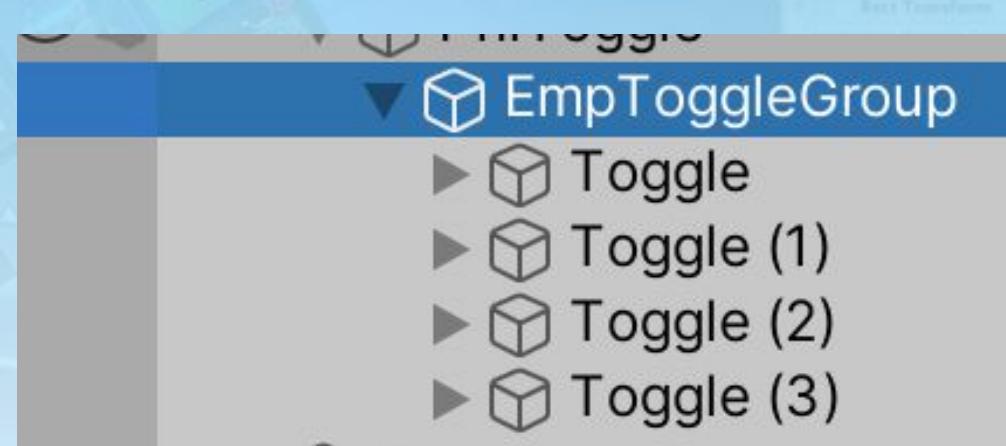
En este caso, podemos seleccionar una, dos, todas o ninguna opción.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

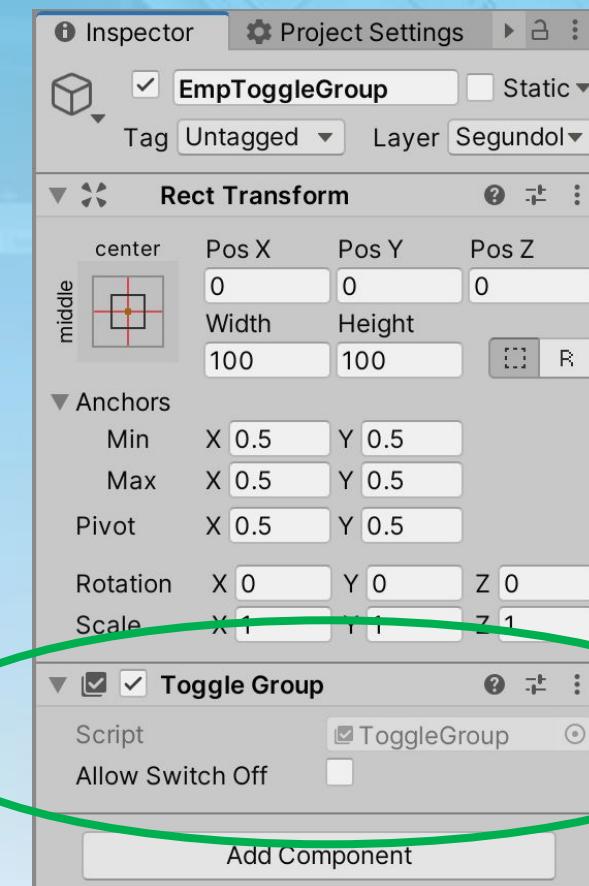
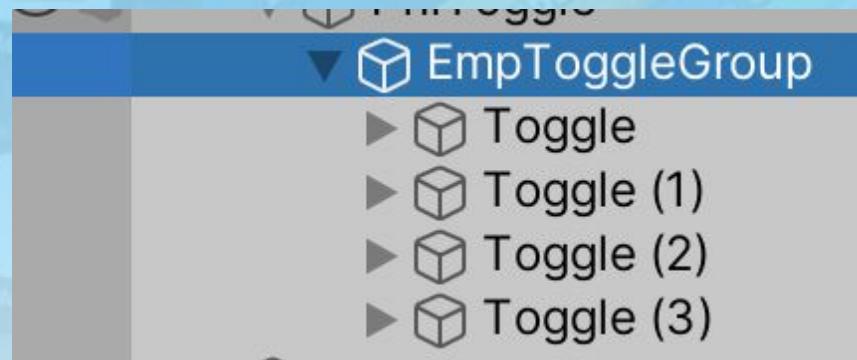
- Ahora, lo que nos interesa es que únicamente haya una única opción activa, solo y siempre, una única opción.
- Para conseguir esto, debemos agrupar los GameObjects de tipo “Toggle”.
- Para hacer esto, vamos a insertar dentro de nuestro “PnlToggle” un GameObject de tipo “Empty” y le vamos a llamar “EmpToggleGroup” o como queramos.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

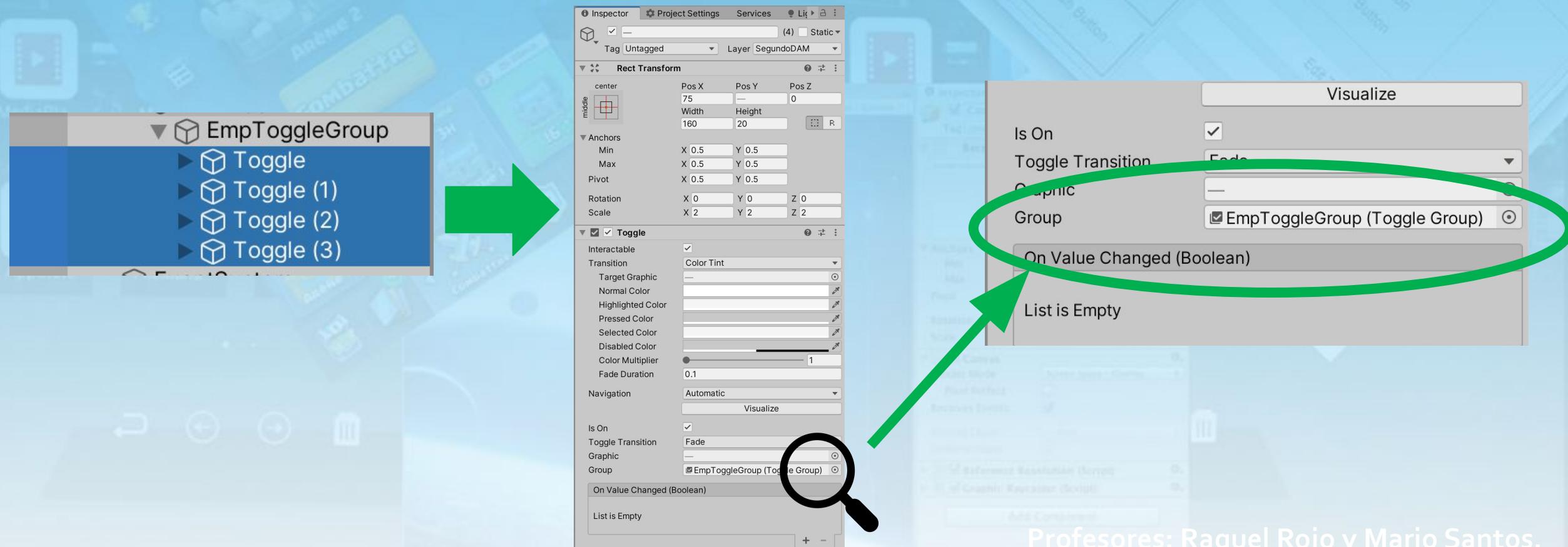
- En la ventana del "Inspector" del GameObject de tipo "Empty" y que hemos denominado "EmpToggleGroup", vamos a insertar el componente de tipo "Toggle Group".



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

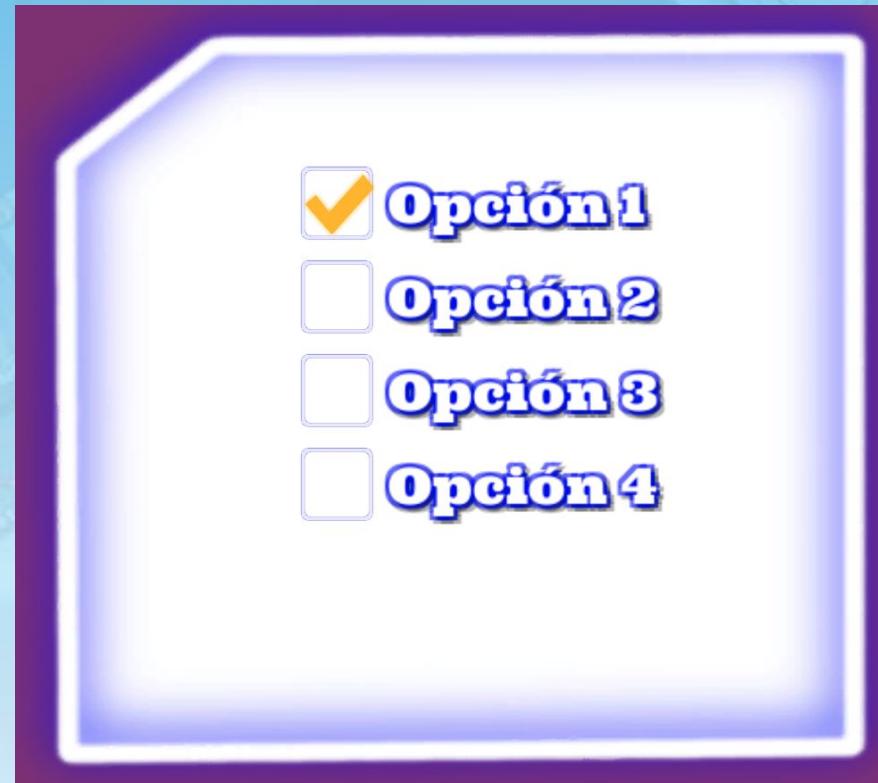
- A continuación vamos a seleccionar todos los GameObjects de tipo “Toggle” que queremos agrupar y vamos a insertar el GameObject de tipo “Empty” y que hemos denominado “EmpToggleGroup”, dentro del parámetro de “Group”.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

- A partir de este momento, únicamente puede haber un único, y solo uno, toggle activo.



[Ver el vídeo UT3_Toggle_ParteA](#)

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

Veamos ahora un poco de Script. Vamos a añadir un GameObject de tipo Text denominado “TxtValorToggle” en el que vamos a mostrar la opción seleccionada. Este GameObject cuelga directamente del GameObject “Canvas”.

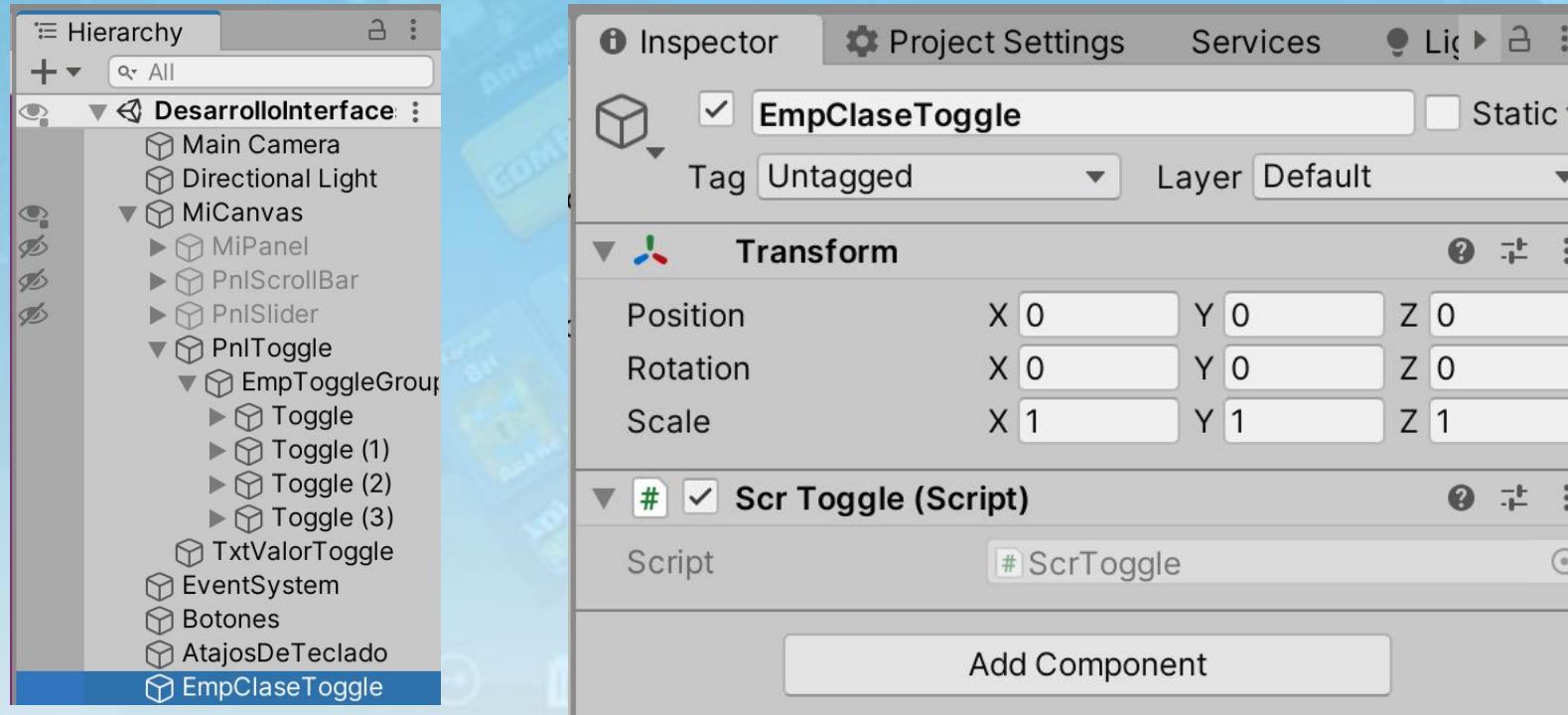


[Ver el vídeo UT3_Toggle_ParteB_Script](#)

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

Añadimos en el proyecto (**no** dentro del Canvas), otro GameObject de tipo “Empty” que vamos a denominar “EmpClaseToggle” y al que vamos a asociar un Script que usaremos como clase para trabajar con los eventos del componente “Toggle”.



```
/*
 * Date: 12 de Septiembre de 2020.
 * Class: ScrToggle.
 * Description: Visualizamos la opción seleccionada
 * de un conjunto de Toggle's.
 * Author: Raquel Rojo y Mario Santos.
 * Palomeras Vallecas - Villablanca
 */
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI; //No olvidar esta librería

public class ScrToggle : MonoBehaviour
{
    private GameObject txtValorToggle;

    // Start is called before the first frame update
    void Start()
    {
        //Apuntamos al Componente de Text, en el que va
        //a aparecer la opción seleccionada.
        txtValorToggle = GameObject.Find("TxtValorToggle");
    }

    //Creamos una función pública.

    public void VisualizarOpcion(string valor)
    {
        //Visualizamos en nuestro componente Text, la opción.
        txtValorToggle.GetComponent<Text>().text = valor;
    }
}
```



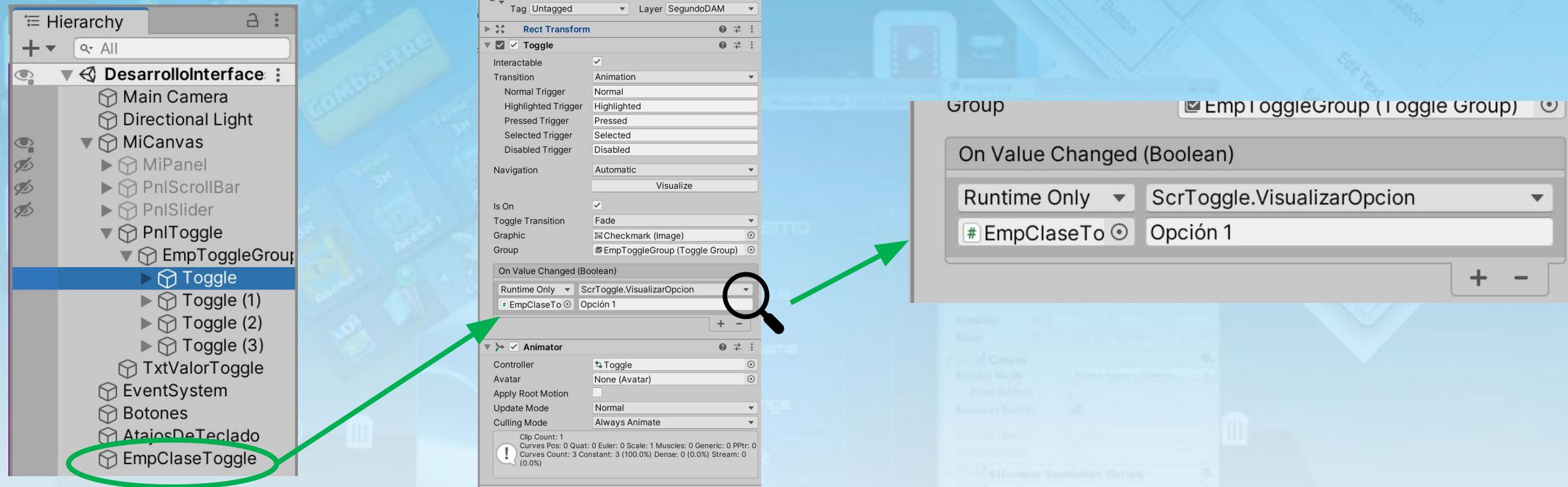
[Ver el vídeo UT3_Toggle_ParteB_Script](#)

Profesores: Raquel Rojo y Mario Santos.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Toggle

Una vez creado el Script y asociado al GameObject “EmpClaseToggle”, se lo vamos a asociar a cada uno de los Toggle, de igual forma a como hemos trabajado con los botones y con los Sliders.



[Ver el vídeo UT3_Toggle_ParteB_Script](#)

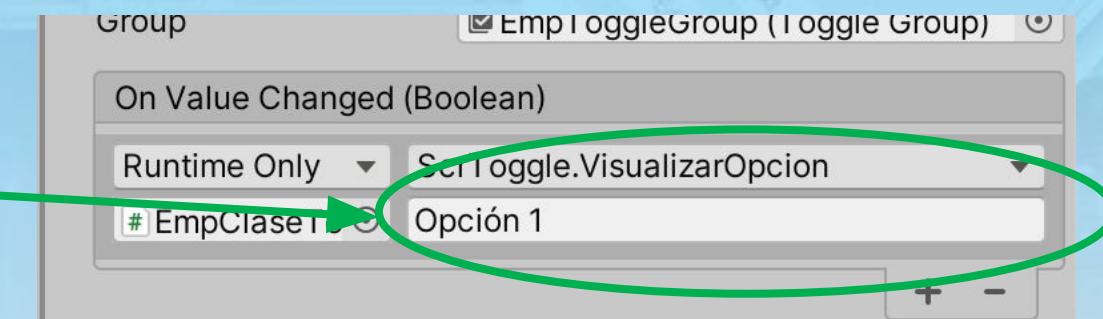
Profesores: Raquel Rojo y Mario Santos.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales:
Componentes básicos.

Toggle

Fíjate que la función pública “VisualizarOpcion” que hemos creado en el script “ScrToggle”, tiene un parámetro de entrada, que es lo que le pasamos desde la configuración, en este caso “Opción 1”.

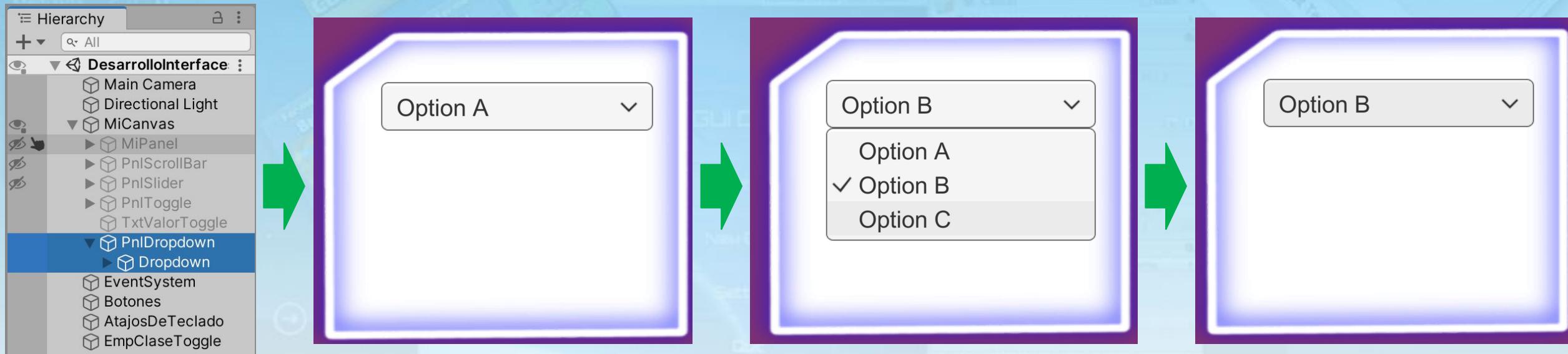
```
//Creamos una función pública.  
  
public void VisualizarOpcion(string valor)  
{  
    //Visualizamos en nuestro componente Text, la opción.  
    txtValorToggle.GetComponent<Text>().text = valor;  
}
```



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Dropdown

El GameObject “Dropdown” es una lista desplegable, con una serie de opciones, de las cuales, únicamente podremos seleccionar una. La opción seleccionada, se podrá leer siempre en la parte superior del Dropdown. Para empezar a trabajar con este componente, vamos a crearnos un panel nuevo en el Canvas, que vamos a denominar “PnlDropdown” y le vamos a emparentar un GameObject hijo de tipo “Dropdown”.

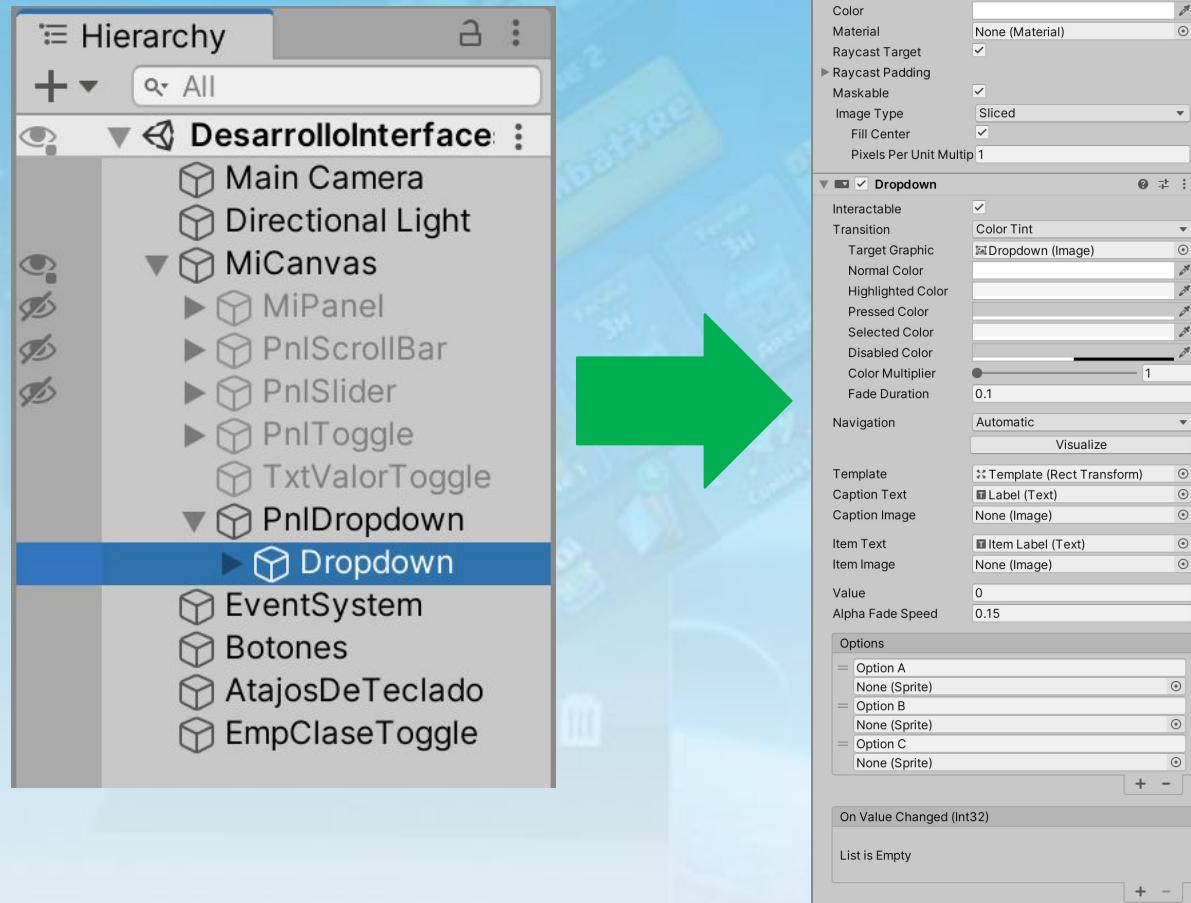


UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales:

Componentes básicos.

Dropdown

Veamos más en detalle cómo queda la ventana Hierarchy después de introducir el GameObject Dropdown, y en la ventana “Inspector”, para ver las distintas propiedades.



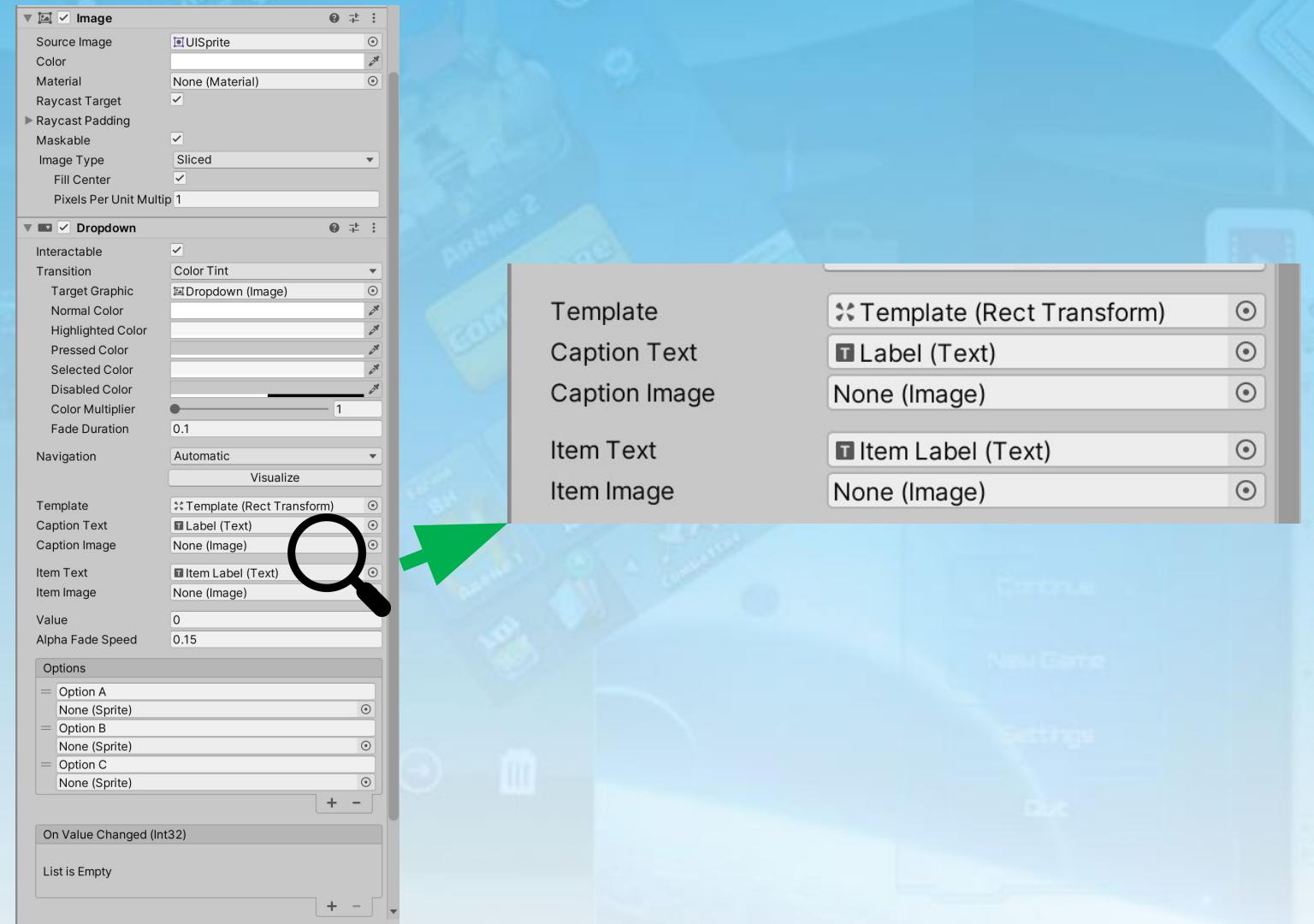
Como podemos observar en la ventana “Inspector”, los componentes que contiene ya los hemos visto casi en su totalidad.

Tiene un componente “Image” con el que hemos estado trabajando desde la U.T.2 y que es el que vamos a utilizar para personalizar nuestra lista de selección.

Tiene el componente “Dropdown” que lo primero que nos muestra es la transición que queremos configurar. Este aspecto también ha sido ya trabajado en la U.T.2 y en esta U.T.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Dropdown



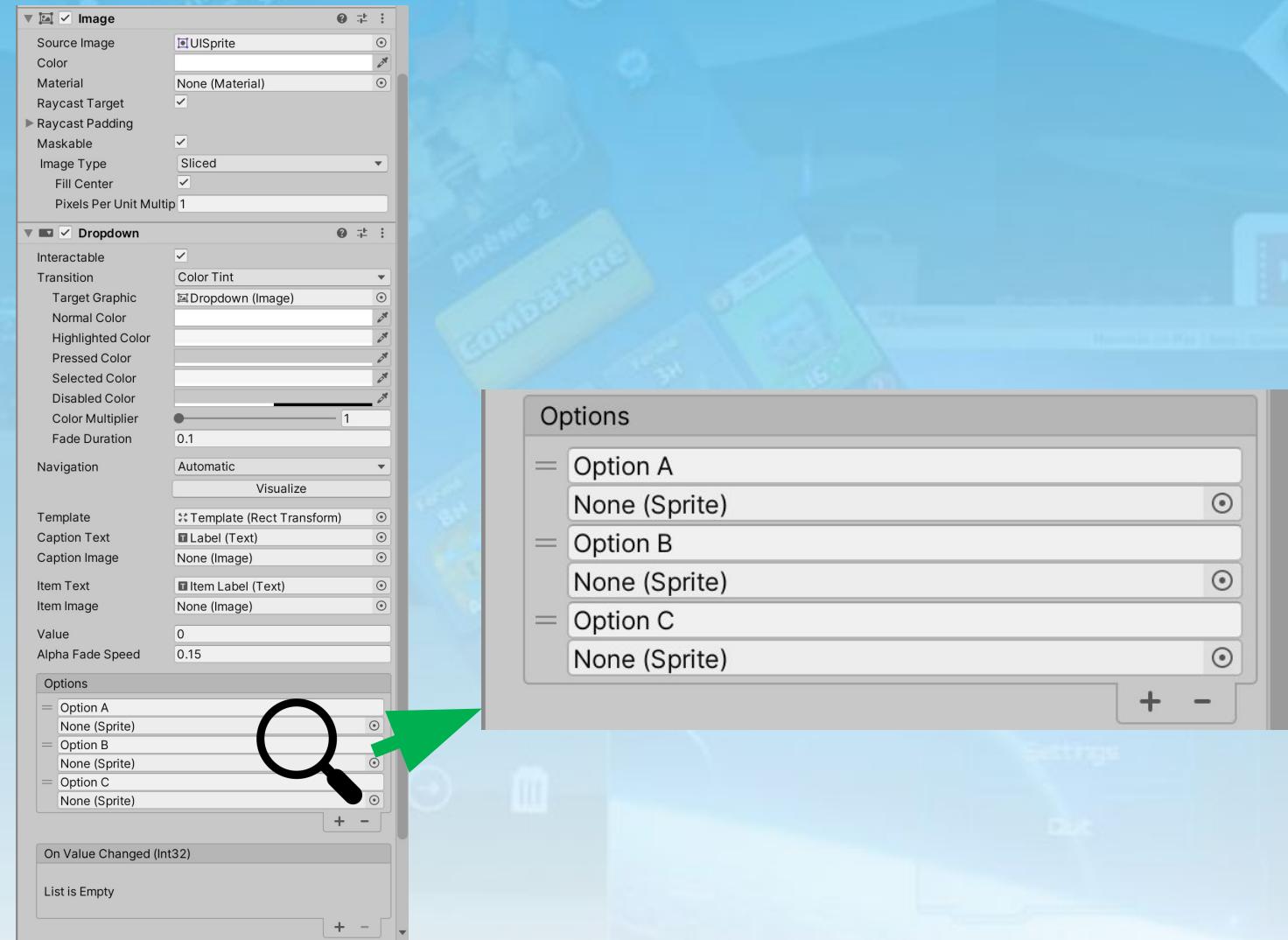
Estos parámetros son los que vinculan el GameObject padre “Dropdown” con los GameObjects hijos “Label” y “Template”.

Esto nos permitiría hacer cosas como cambiar el sitio de visualización de la opción seleccionada, pero, por ahora, no lo vamos a modificar.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales:

Componentes básicos.

Dropdown



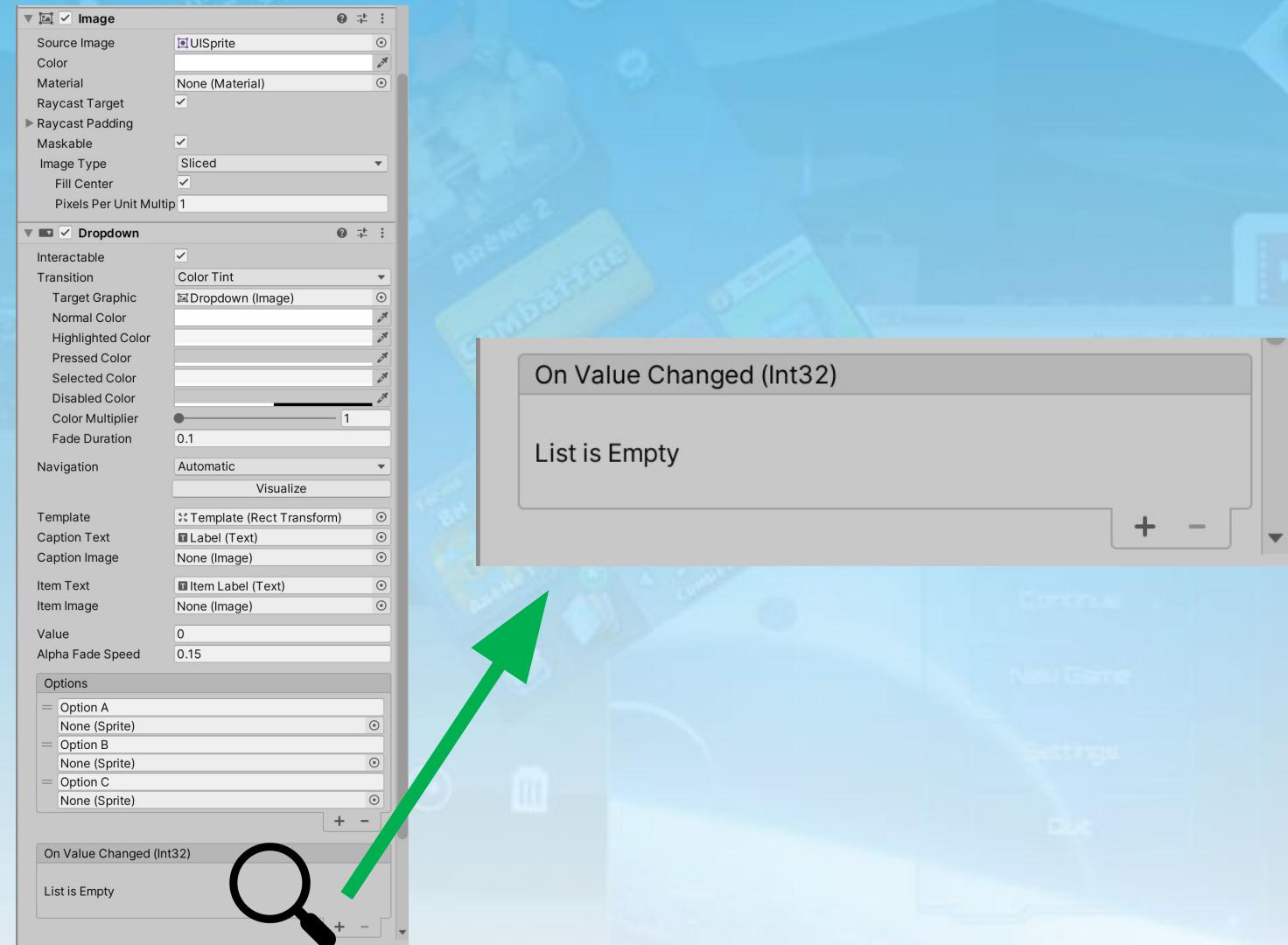
El siguiente parámetro que nos encontramos es el de “Options”, que es en el que vamos a escribir las diferentes opciones que le queremos mostrar al usuario.

Fijaros que a cada opción se le puede asociar un Sprite, lo que es interesante, cara al diseño.

Por defecto, aparecen 3 opciones, pero se pueden incrementar o decrementar con los botones situados en la esquina inferior derecha “+” y “-”.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Dropdown



Y el último parámetro que nos encontramos del componente “Dropdown” es el gestor de eventos, en este caso, el “On Value Changed”, en el que vamos a cargar una función pública.

Para más información sobre estos parámetros entrar en el siguiente link.

<https://docs.unity3d.com/2020.2/Documentation/Manual/script-Dropdown.html>

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Dropdown

Vamos a personalizar nuestro Dropdown para que se ajuste más a nuestra interface. Se han creado en Gimp los siguientes Sprites y se ha personalizado como se observa en las imágenes.



SptDropdown

SptAlemania



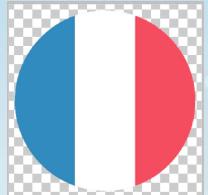
SptEspaña



SptChkNaranja



SptFrancia



SptInglaterra



SptCheckMark

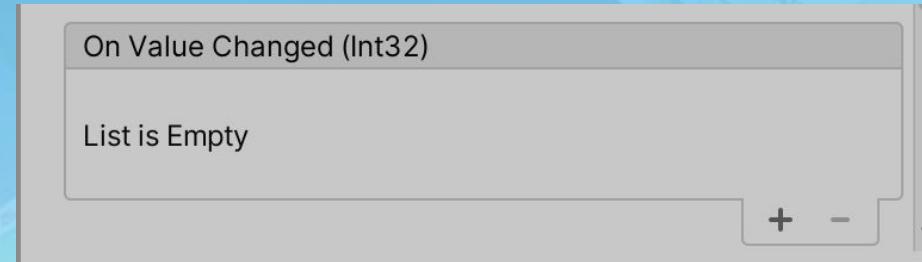


[Ver el vídeo UT3_DropDown](#)

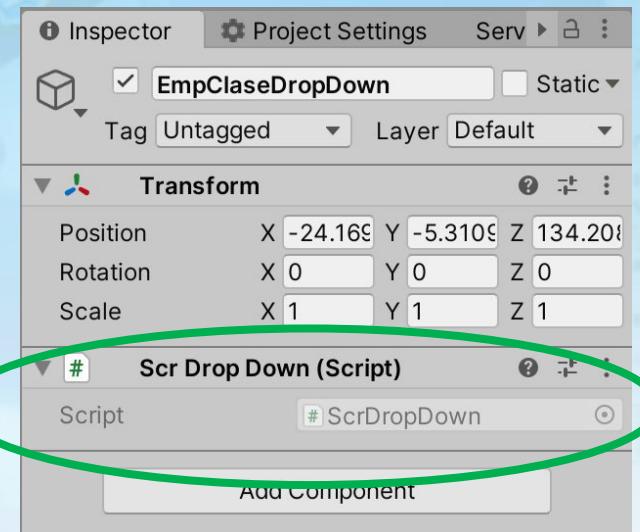
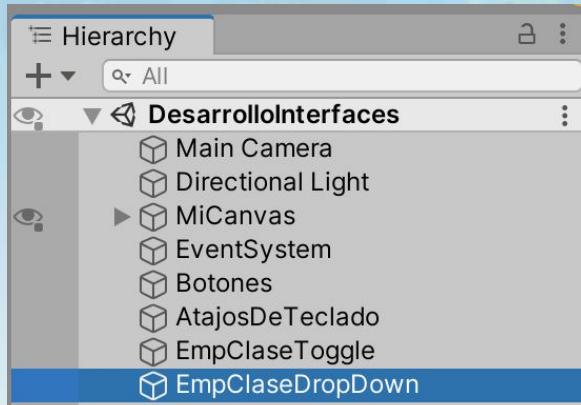
UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Dropdown

Para comprobar el gestor de eventos que nos proporciona el componente Dropdown,



Hemos añadido un GameObject de tipo Empty que hemos denominado “EmpClaseDropdown”. Dentro de este GameObject hemos asociado un Script denominado “ScrDropDown”.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales:

Componentes básicos.

Dropdown

El Script, no contiene ningún comando nuevo ni nada con lo que no hayamos trabajado.

```
/*
 * Date: 15 de Septiembre de 2020.
 * Class: ScrDropDown.
 * Description: Desde este script, trabajamos con el
 * valor que nos devuelve el DropDownList
 * Author: Raquel Rojo y Mario Santos.
 * Palomeras Vallecás - Villablanca
 */
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

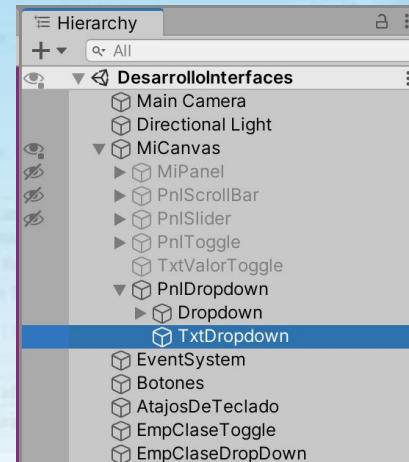
public class ScrDropDown : MonoBehaviour
{
    //Declaro dos punteros privados a los GameObjects.
    private GameObject ptrDropdown;
    private GameObject ptrTxtDropdown;

    //Creamos la función pública a la que vamos a llamar
    //desde el Dropdown.

    public void ValorDropBox()
    {
        ptrDropdown = GameObject.Find("Dropdown");
        ptrTxtDropdown = GameObject.Find("TxtDropdown");

        if (ptrDropdown.GetComponent<Dropdown>().value == 0)
        {
            ptrTxtDropdown.GetComponent<Text>().text = "España";
        }
        if (ptrDropdown.GetComponent<Dropdown>().value == 1)
        {
            ptrTxtDropdown.GetComponent<Text>().text = "Inglaterra";
        }
        if (ptrDropdown.GetComponent<Dropdown>().value == 2)
        {
            ptrTxtDropdown.GetComponent<Text>().text = "Francia";
        }
        if (ptrDropdown.GetComponent<Dropdown>().value == 3)
        {
            ptrTxtDropdown.GetComponent<Text>().text = "Alemania";
        }
    }
}
```

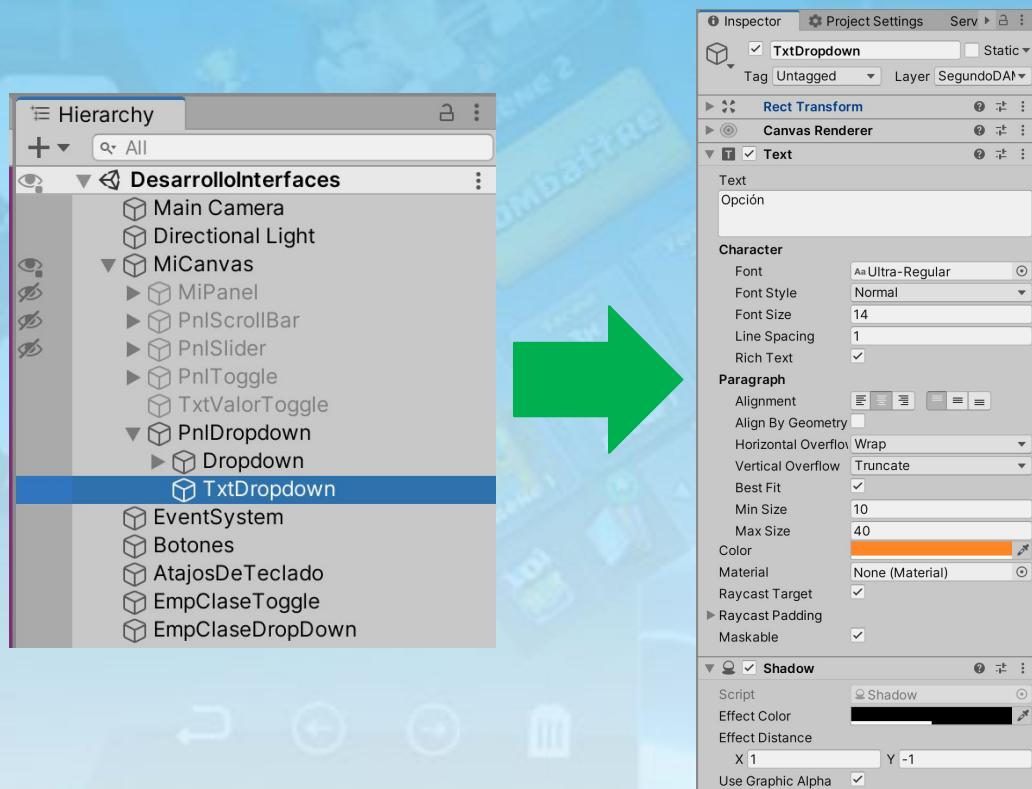
Además de añadir el GameObject de tipo Empty en el proyecto, vamos a insertar un GameObject de tipo Text dentro de nuestro panel “PnlDropBox” y lo vamos a denominar TxtDropbox.



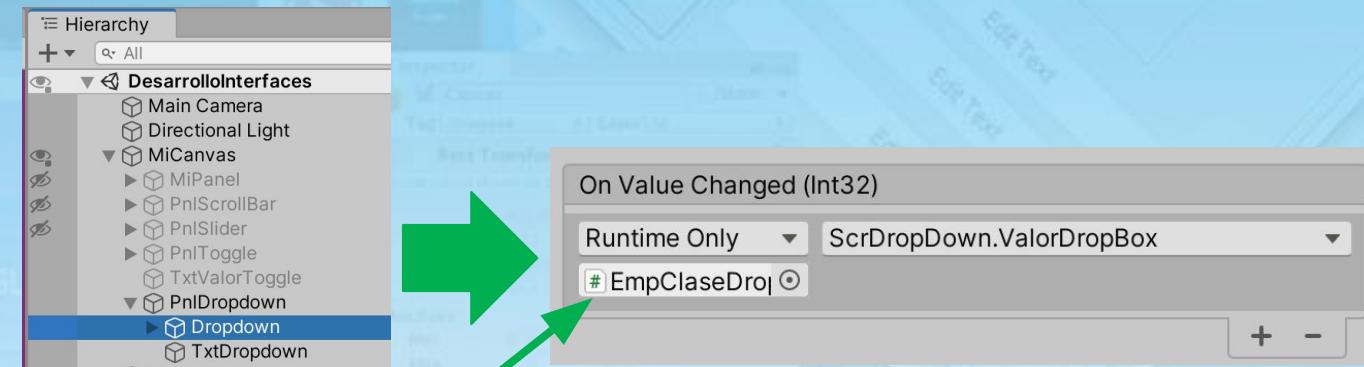
UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Dropdown

La configuración de este GameObject es la que muestra la imagen.



Ahora, nos queda cargar la función pública que hemos creado en nuestro Script “ScrDropDown”, dentro del Componente Dropdown.



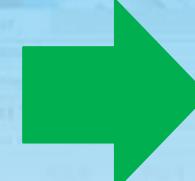
[Ver el vídeo UT3_DropDown](#)

Profesores: Raquel Rojo y Mario Santos.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales:
Componentes básicos.

Dropdown

Y lo que obtenemos es lo que nos muestran las imágenes.



Todo lo podéis ver en el paquete del proyecto.

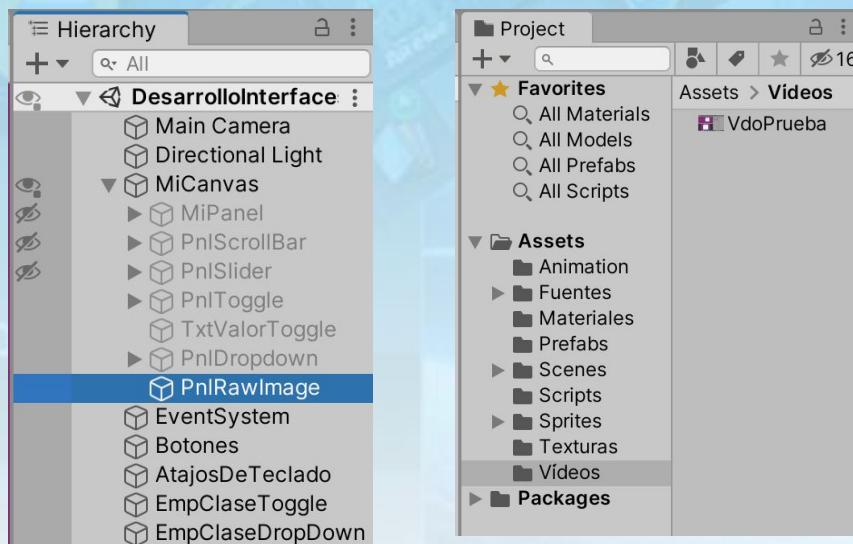
UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

Este último componente o GameObject de UI, nos sirve para visualizar sobre él texturas, Sprites o vídeos, a diferencia del GameObject de tipo Image, que únicamente nos permite cargar Sprites.

Veamos cómo cargamos un vídeo y lo reprodujimos en nuestra escena.

Vamos a crear un panel nuevo que vamos a denominar “PnlRawImage”. Así mismo, vamos a crear una carpeta nueva en nuestro proyecto que vamos a denominar “Vídeos” y vamos a guardar en ella el vídeo que queremos visualizar.



El vídeo subido al proyecto se denomina “VdoPrueba” y está en formato .mp4.

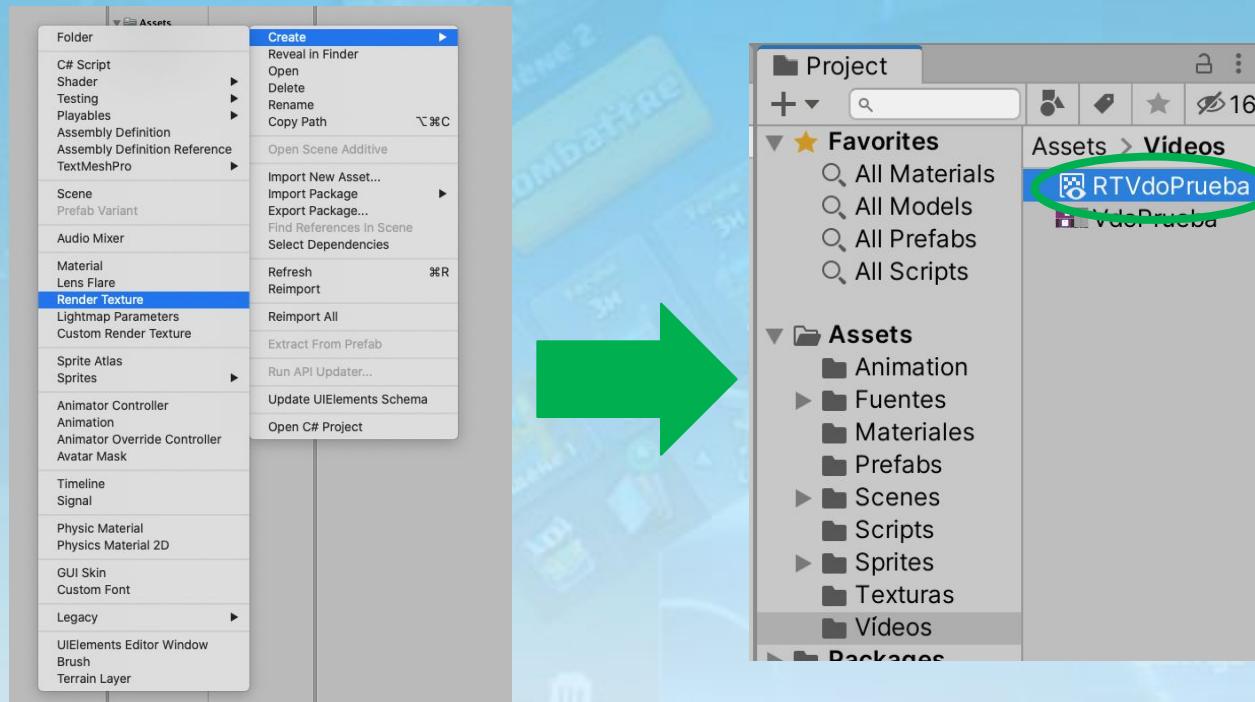


Recordar que todo este material se encuentra en el paquete de proyecto que acompaña a esta presentación.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

Una vez cargado el vídeo, vamos a crear una “Render Texture”. Se le puede llamar igual que el nombre del vídeo, pero puede ser cualquier otro nombre.

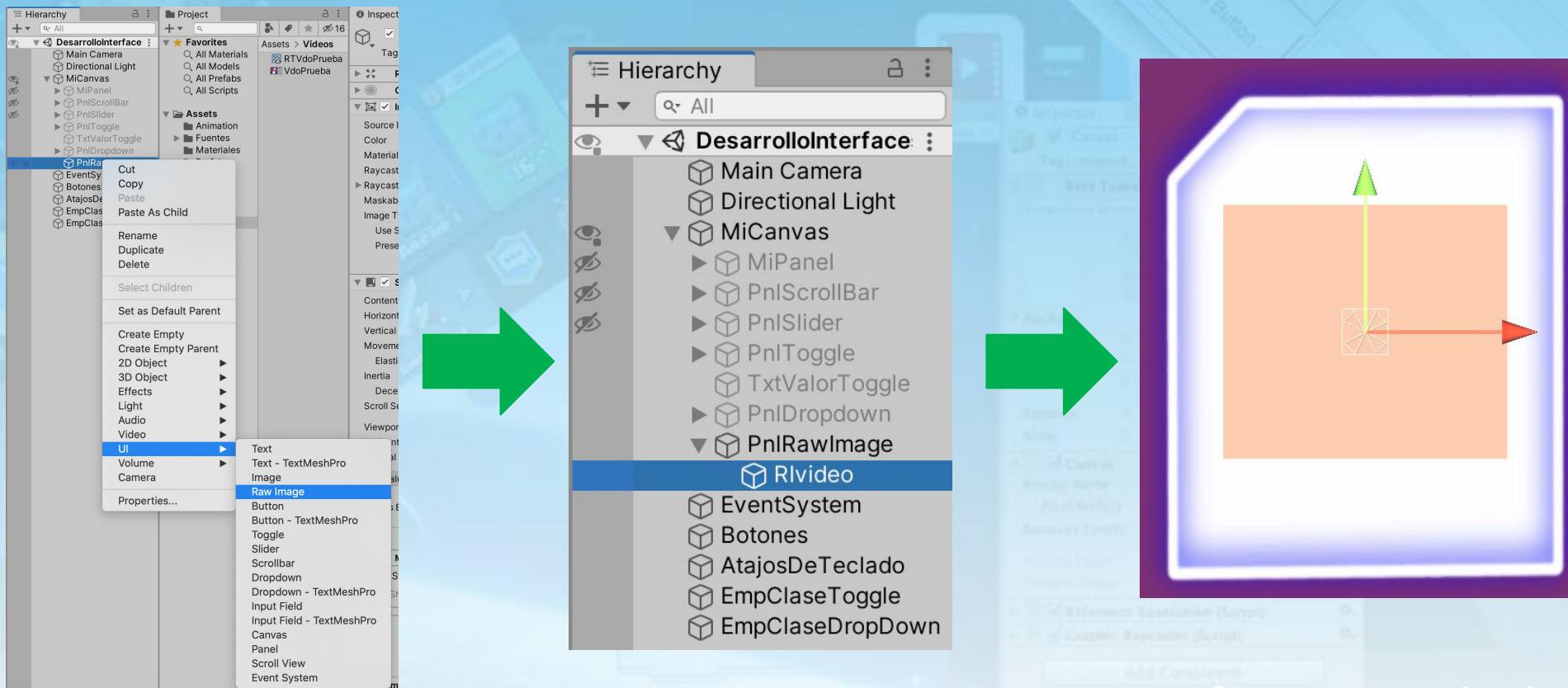


En este caso, creamos la RenderTexture, dentro de la carpeta Vídeos y la denominamos “RTVdoPrueba”.

UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

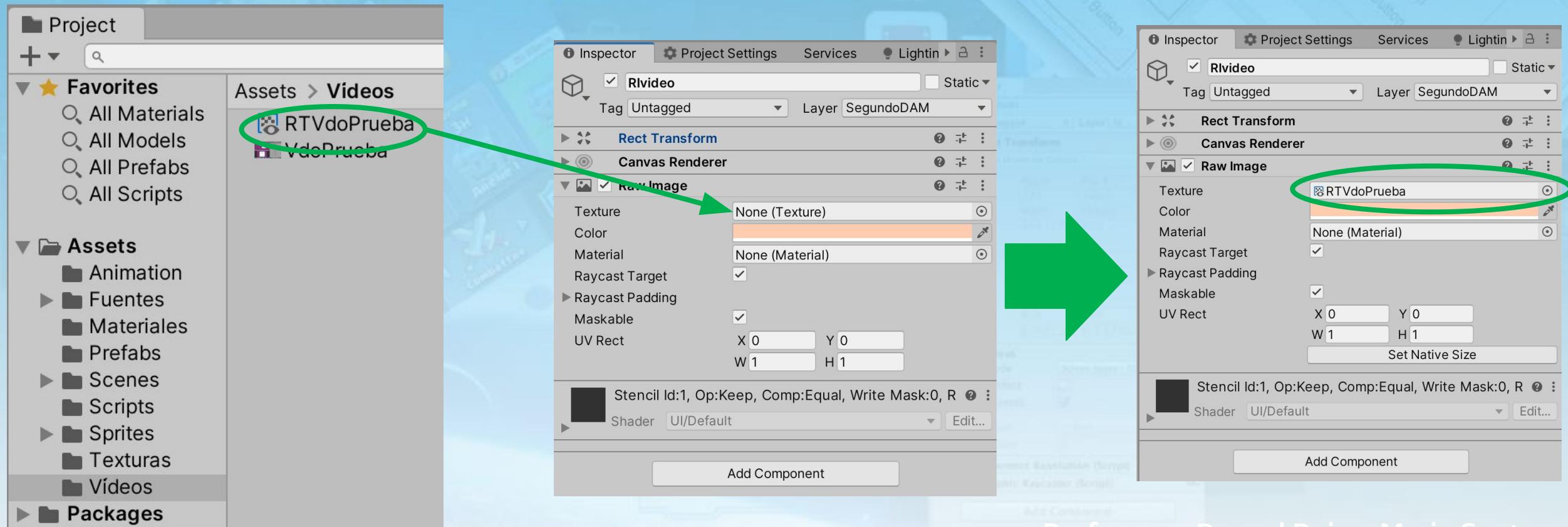
Ahora introducimos un GameObject o componente de tipo RawImage dentro de nuestro panel “PnlRawImage”. A este GameObject le vamos a denominar “Rvideo”.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

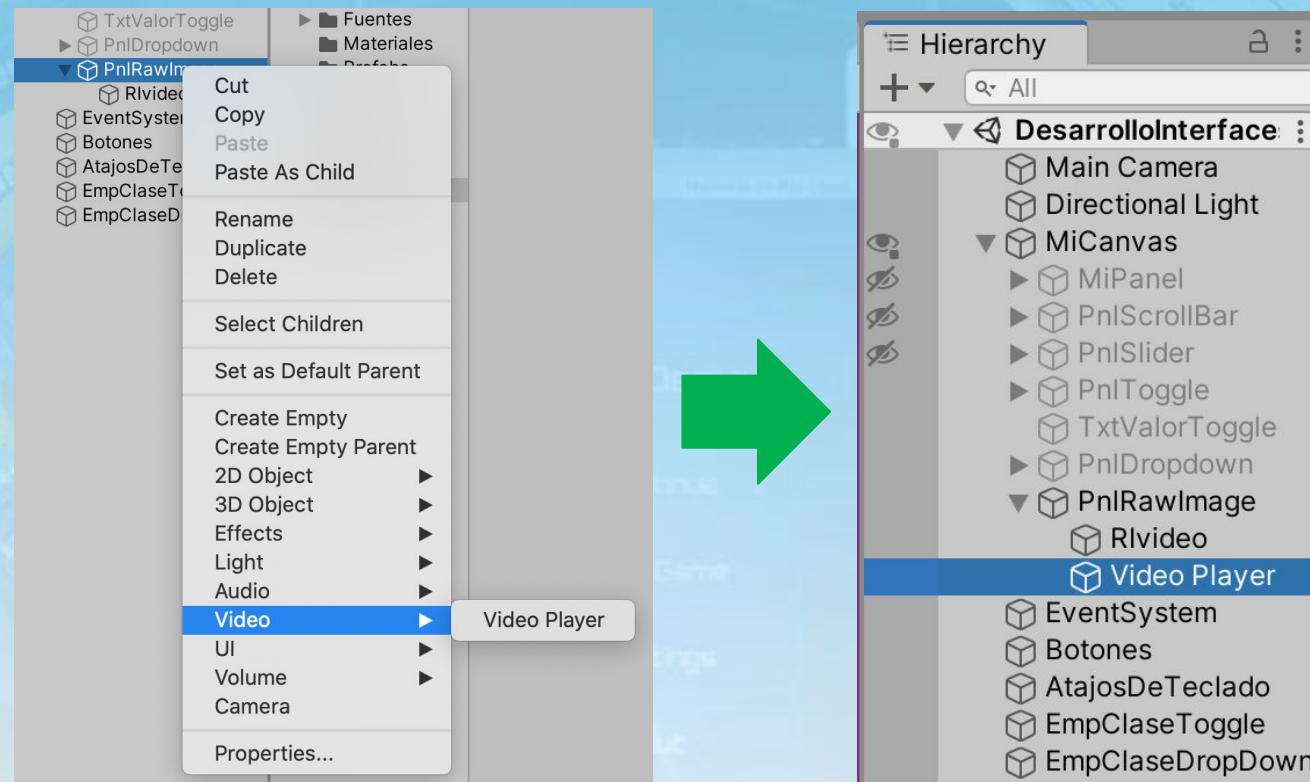
En la ventana “Inspector” del GameObject “Rvideo”, vamos a insertar el “Render Texture” RTVdoPrueba, que hemos creado con anterioridad, en el apartado “Texture”, como se muestra en la figura.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

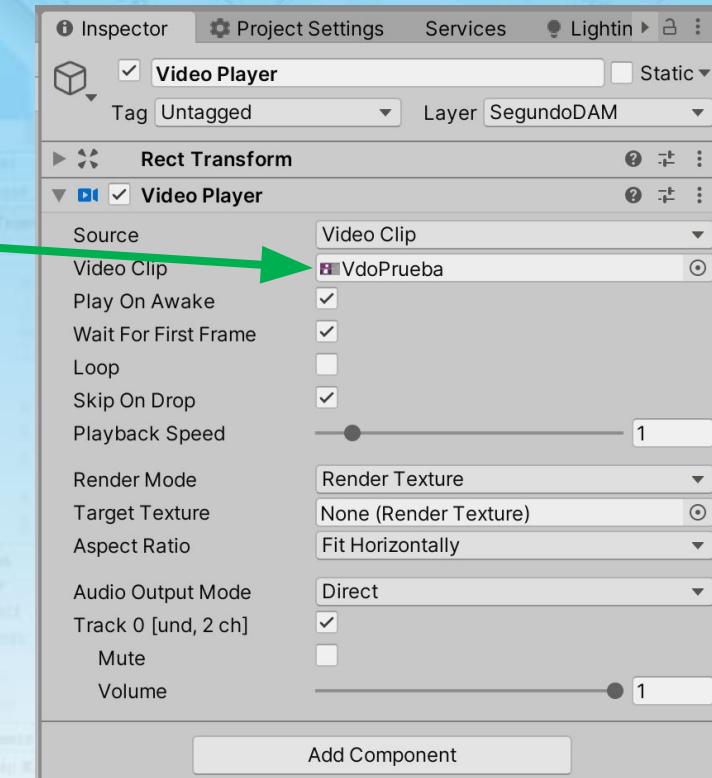
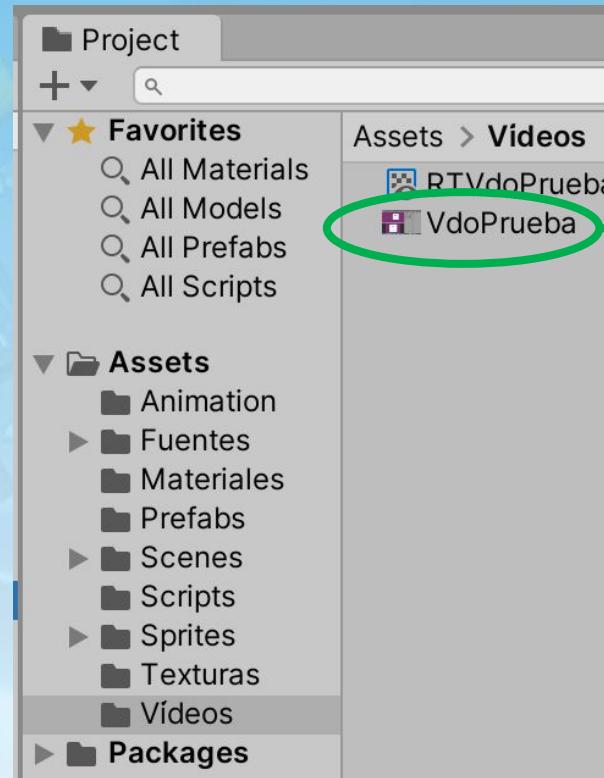
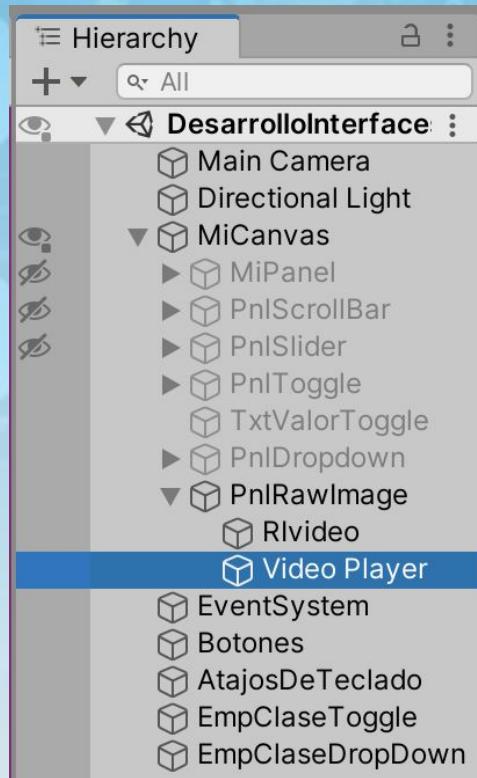
Lo siguiente que vamos a hacer es introducir un GameObject de tipo “VideoPlayer” dentro del Panel “PnlRawImage”.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

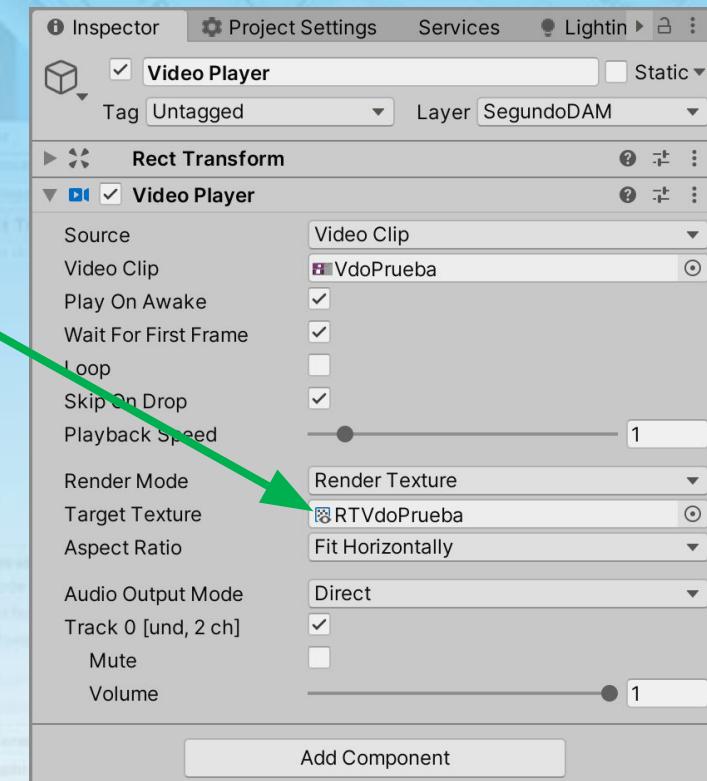
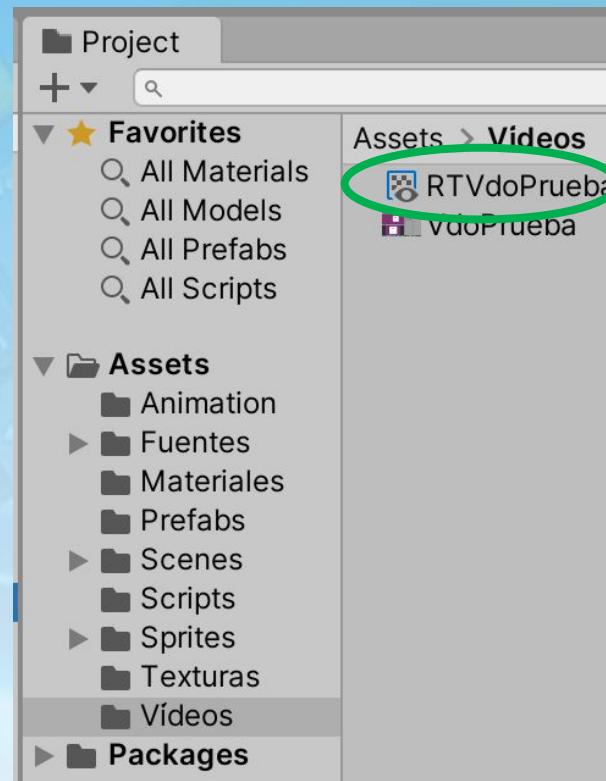
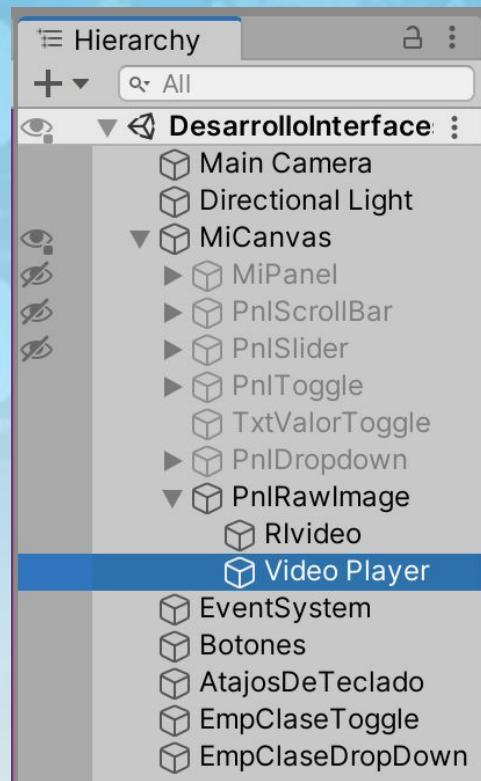
Ahora, vamos a ir a la ventana “Inspector” del GameObject “Video Player” y vamos a cargar el vídeo “VdoPrueba”, que hemos cargado en nuestro proyecto, en el apartado de “Video Clip”.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

Cargamos también en el apartado “Target Texture” la “Render Texture” RTVdoPrueba



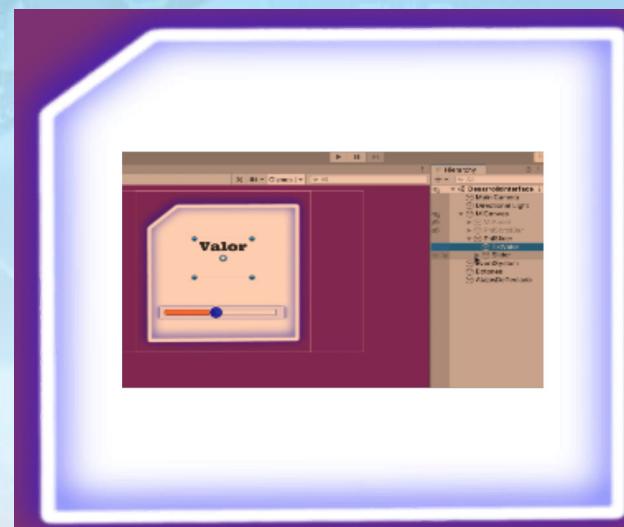
UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

Si se pulsa “Play”, ya se podrá observar el vídeo en la escena. Debemos ajustar el GameObject “Raw Image” al tamaño de la Imagen.

Para hacer esto, debemos saber el tamaño del vídeo y modificar el componente “Transform” del GameObject “Raw Image” con esos valores u otros derivados de su tamaño real.

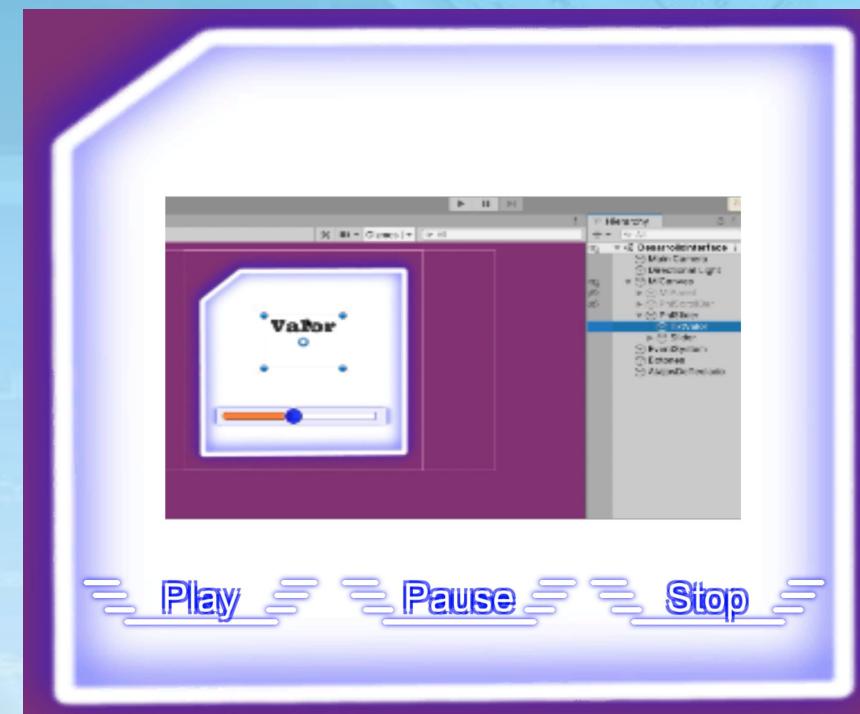
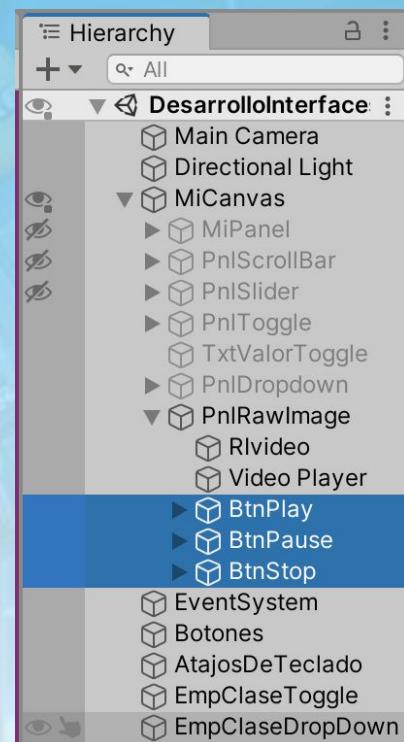
Y por otro lado, desde el componente Video Player, del GameObject “Video Player”, debemos cambiar el parámetro “Aspect Ratio” y poner el valor Stretch.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

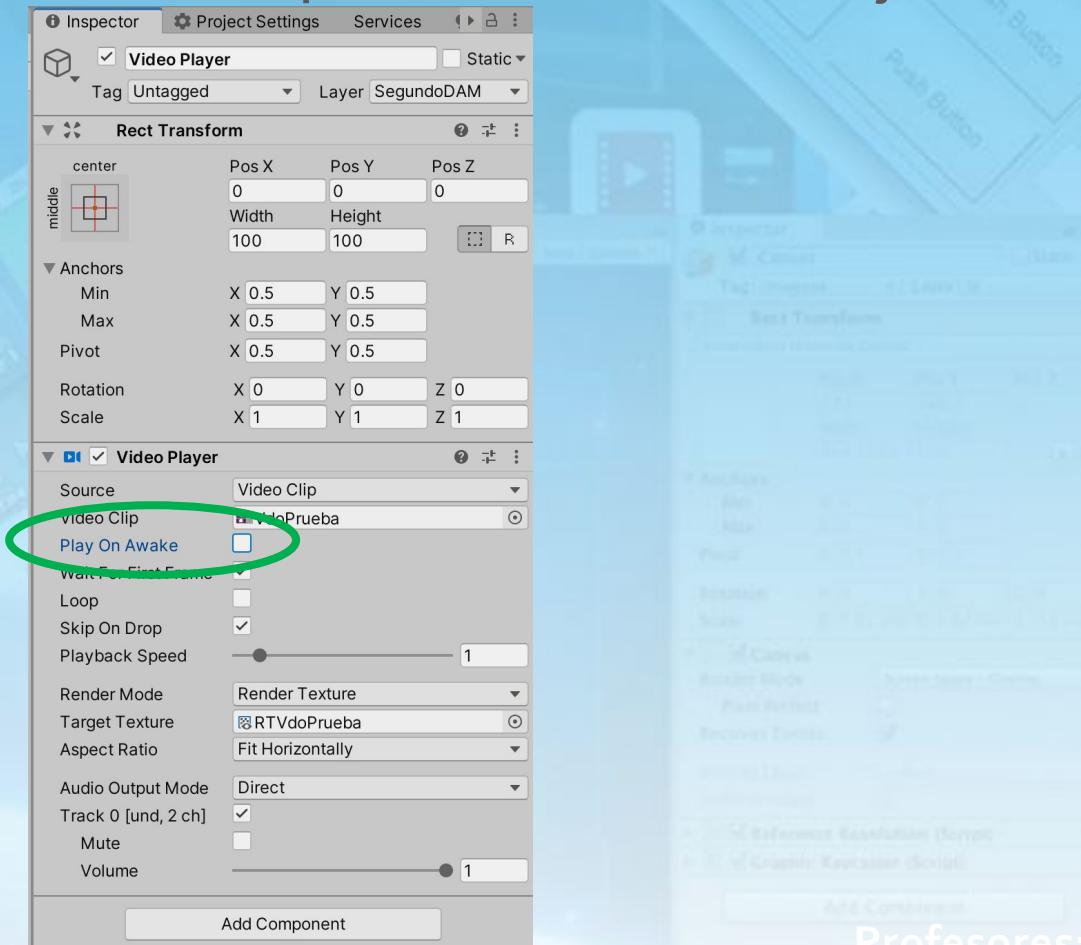
- Ahora, vamos a insertar tres botones en nuestro panel PnlRawImage, para controlar el “Play”, “Pause” y “Stop” de nuestro vídeo.



UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales: Componentes básicos.

Raw Image

Vamos a desactivar que el vídeo se ejecute nada más empezar y que esté en bucle. Esto lo hacemos desde la ventana "Inspector" del GameObject "Video Player".

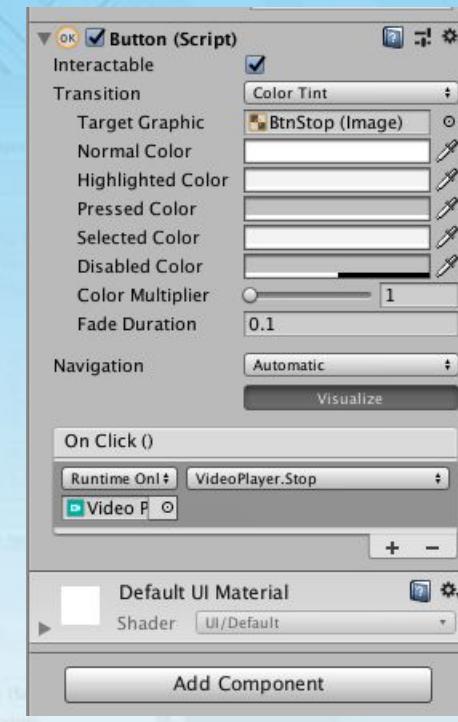
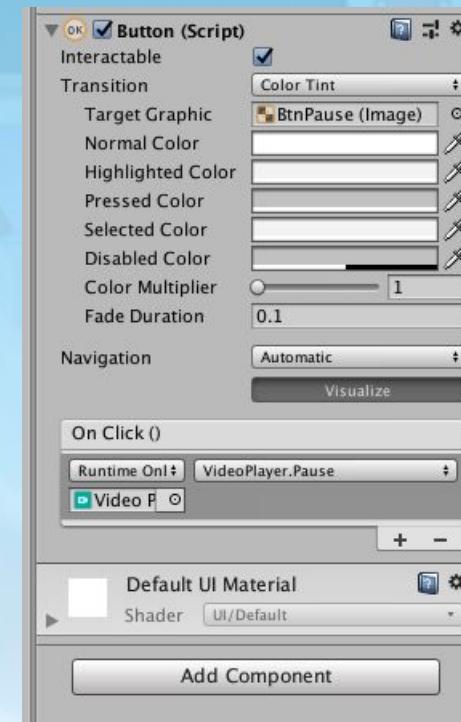
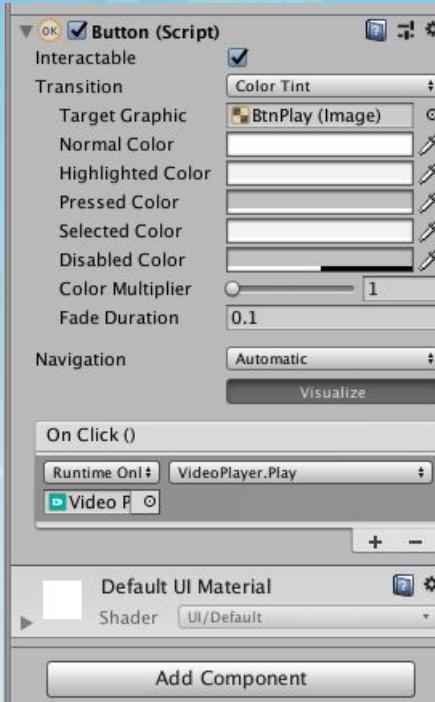


UT3.- Introducción a la generación de interfaces gráficas de usuario con editores visuales:

Componentes básicos.

Raw Image

Ahora vamos a generar el evento `OnClick()` en cada uno de los botones. En el apartado del “Object”, vamos a arrastrar el `GameObject` “Video Player”, y en el `BtnPlay`, cargamos el método “Play”, en el `BtnPause`, el método “Pause” y en el `BtnStop`, el método “Stop”.



[Ver vídeo UT3_RawImage](#)

Profesores: Raquel Rojo y Mario Santos.