

UT.2 Introducción a la generación de interfaces gráficas de usuario con editores visuales:

Componentes contenedores básicos.

Parte C

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

- Componentes contenedores de controles:
 - Canvas. *Diapositivas 3-31*
 - Paneles. *Diapositivas 32-52*

Material Adicional a esta presentación:

- Vídeos:
 - UT2-CanvasPaneles.
 - UT2-CreacionSpritesColorMateriales

Prácticas:

- Práctica_UT2_IntroduccionInterfaceGrafica

- Los Interfaces gráficos de Usuario (a partir de ahora lo nombraremos como GUI), es lo que nos permite que haya una comunicación entre nuestra herramienta software y el usuario.
- Tradicionalmente, los GUI se han venido desarrollando en 2D. Sin embargo, la evolución de las tecnologías ha hecho que hayan evolucionado hacia el 3D, permitiendo crear GUI's en tres dimensiones para entornos de realidad virtual (a partir de este momento VR).
- En esta parte de la UT2, nos vamos a centrar en las GUI's de 2D, estudiando los siguientes elementos:
 - **Canvas:** **Entorno** principal del GUI.
 - **Paneles:** Nos sirve para crear **conjuntos** de componentes (botones, texto, listas, etc) dentro de nuestro entorno Canvas.



Ver el vídeo UT2_CanvasPaneles

Profesores: Raquel Rojo y Mario Santos.

• Canvas

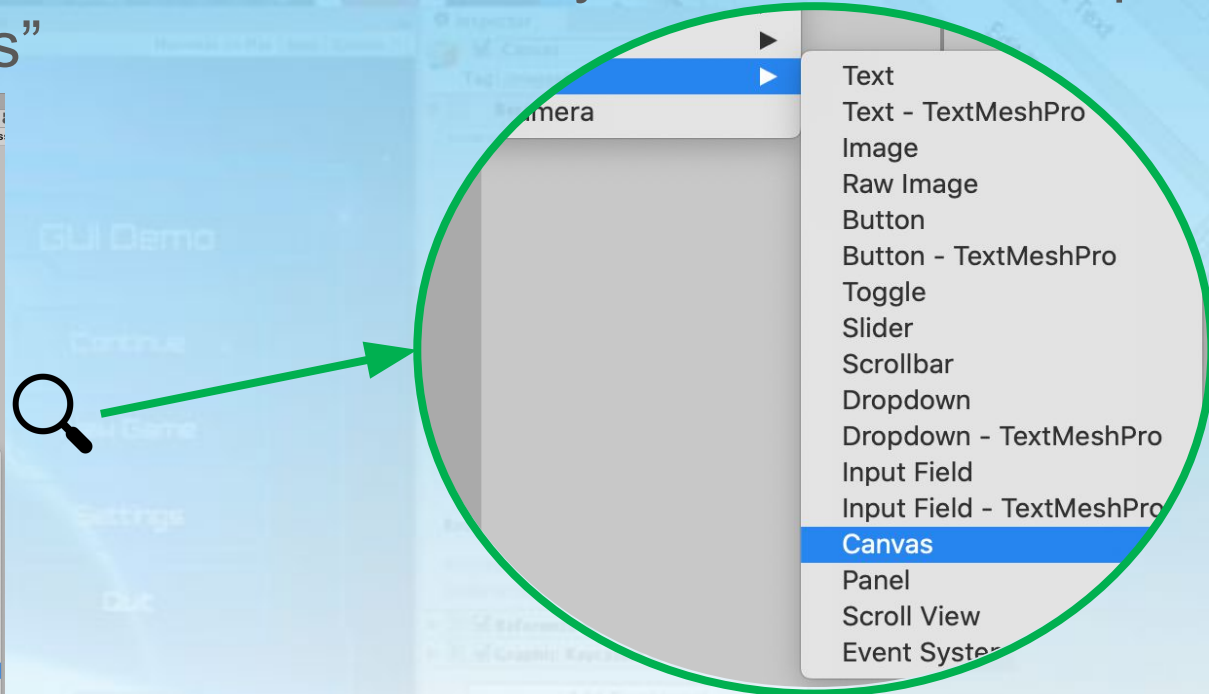
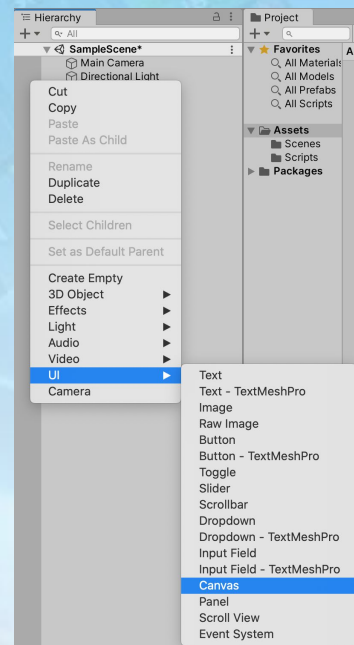
- Nos proporciona un entorno para trabajar con las GUI's.
- Para trabajar con los Canvas, haremos uso del entorno de trabajo de Unity, más concretamente, haremos uso de las ventanas:
 - Hierarchy
 - Scene
 - Game
 - Inspector



Estas ventanas están explicadas en UT2_ParteB_VentanasPrincipalesUnity.

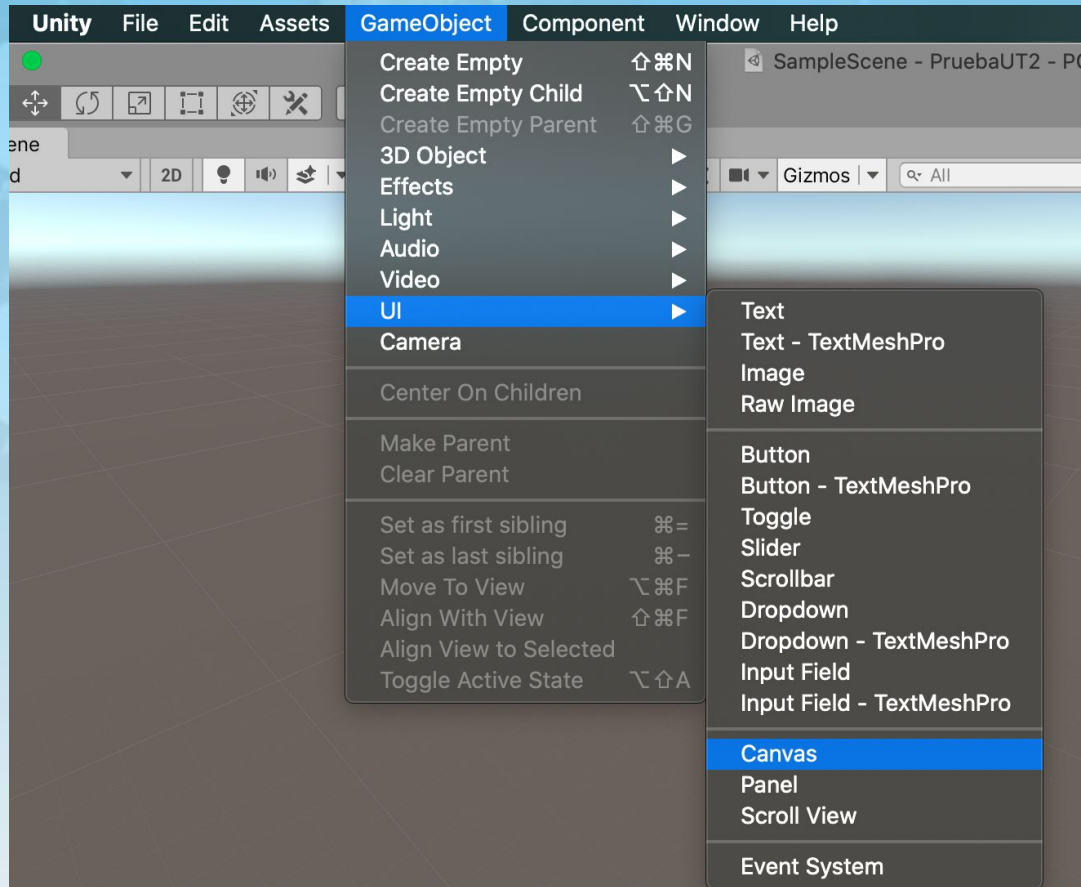
• Canvas

- Ventana Hierarchy: En esta ventana es donde debe aparecer nuestro Canvas, una vez añadido a nuestro proyecto. Tenemos dos formas de hacerlo:
- Una es sacando el menú contextual de Hierarchy, entrando en el apartado de "UI" y seleccionando "Canvas"



• Canvas

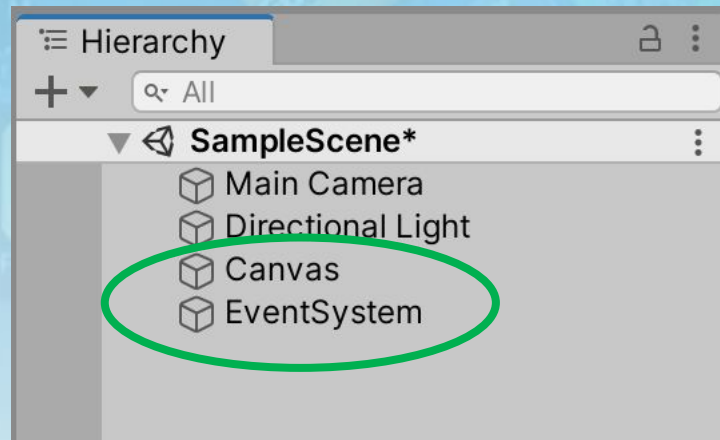
- La segunda forma es desde la barra de menús, desde el menú “GameObject”



Es interesante que el menú se denomine “GameObject”, esto es porque en Unity, no existen Objetos, existen los GameObjects y los podemos cargar desde este menú. Nosotros vamos a trabajar con todos los GameObjects del apartado “UI”.

• Canvas

- Una vez seleccionado el GameObject “Canvas”, veremos que ocurren varias cosas:
- En la ventana Hierarchy aparecerán dos GameObjects nuevos.



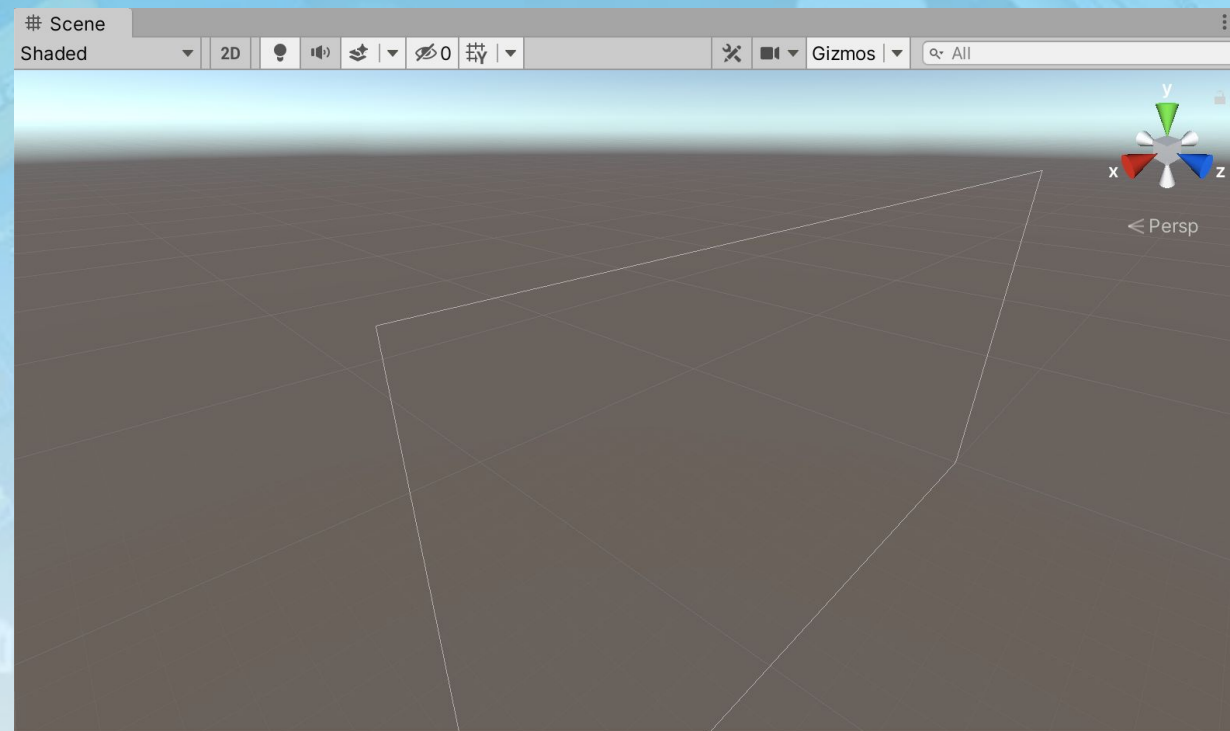
- **Canvas:** El GameObject del Canvas con el que vamos a trabajar.
- **EventSystem:** Este GameObject, siempre, se asocia al Canvas y controla todos los eventos que se producen dentro del Canvas.
- Estos dos GameObjects, pueden ser cambiados de nombre sin ningún problema.



Puede haber varios Canvas, pero solo puede haber un único EventSystem.

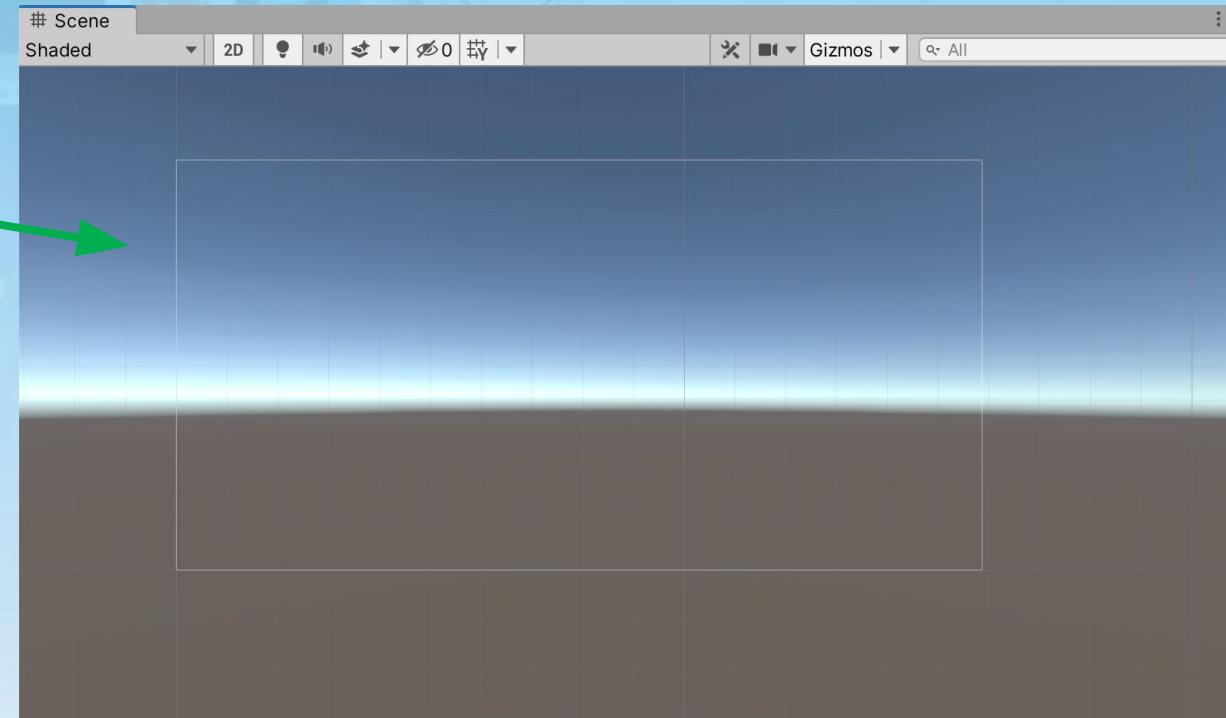
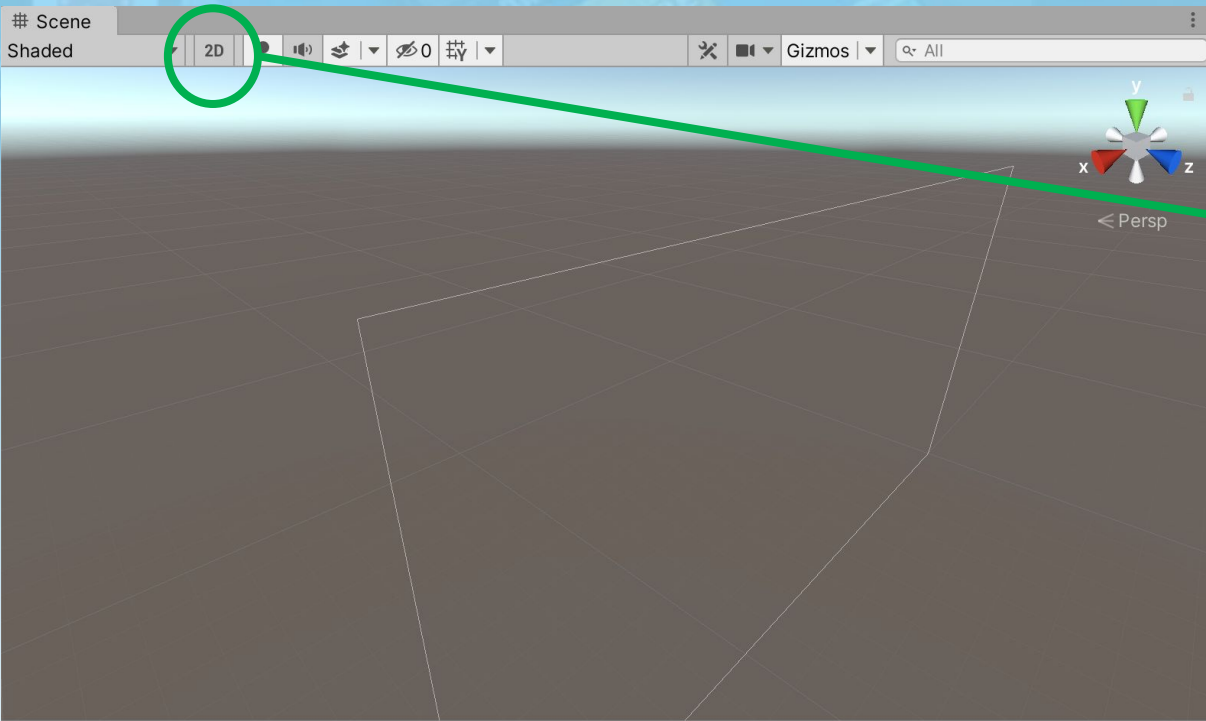
- **Canvas**

- Veremos que en la ventana de “Scene”, aparece un cuadro que nos marca el área de trabajo del Canvas.



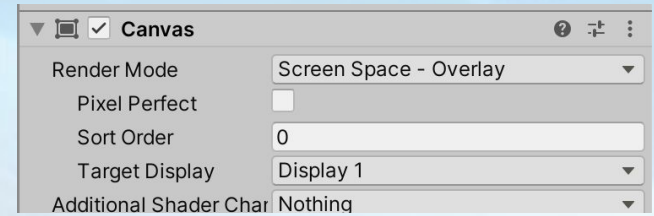
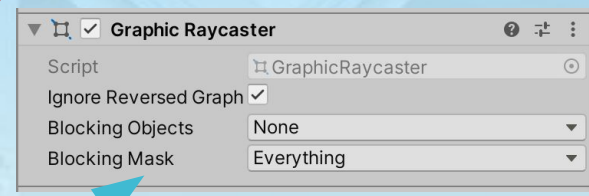
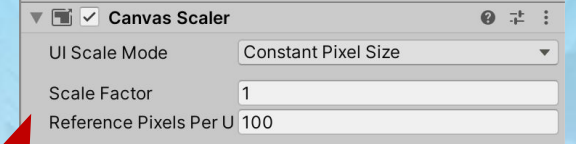
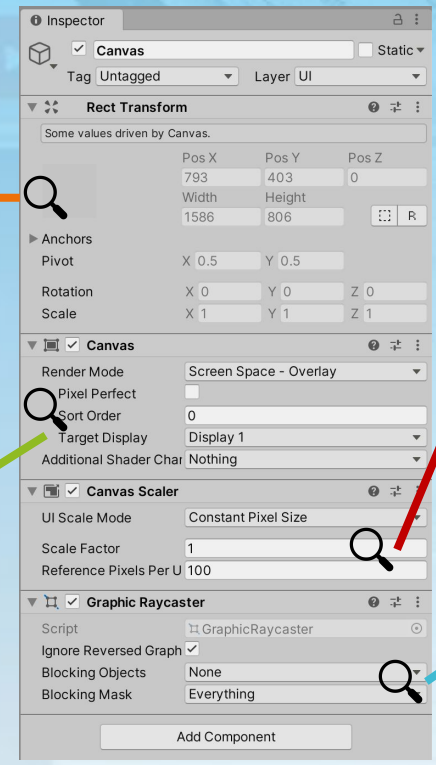
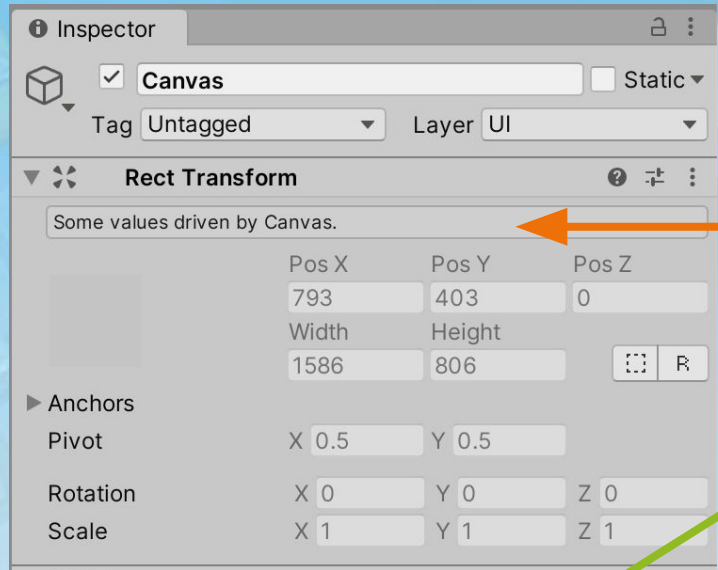
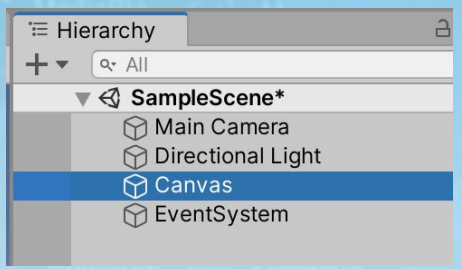
• Canvas

- Si pulsamos sobre el botón “2D”, la vista se fija sobre los ejes X e Y y veremos mejor el espacio del Canvas.



Canvas

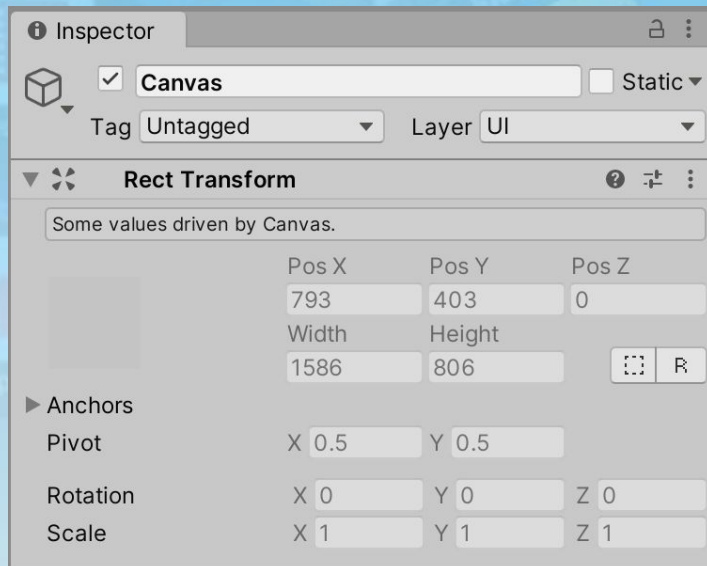
- Si seleccionamos el GameObject "Canvas", veremos todas sus propiedades en la ventana Inspector.



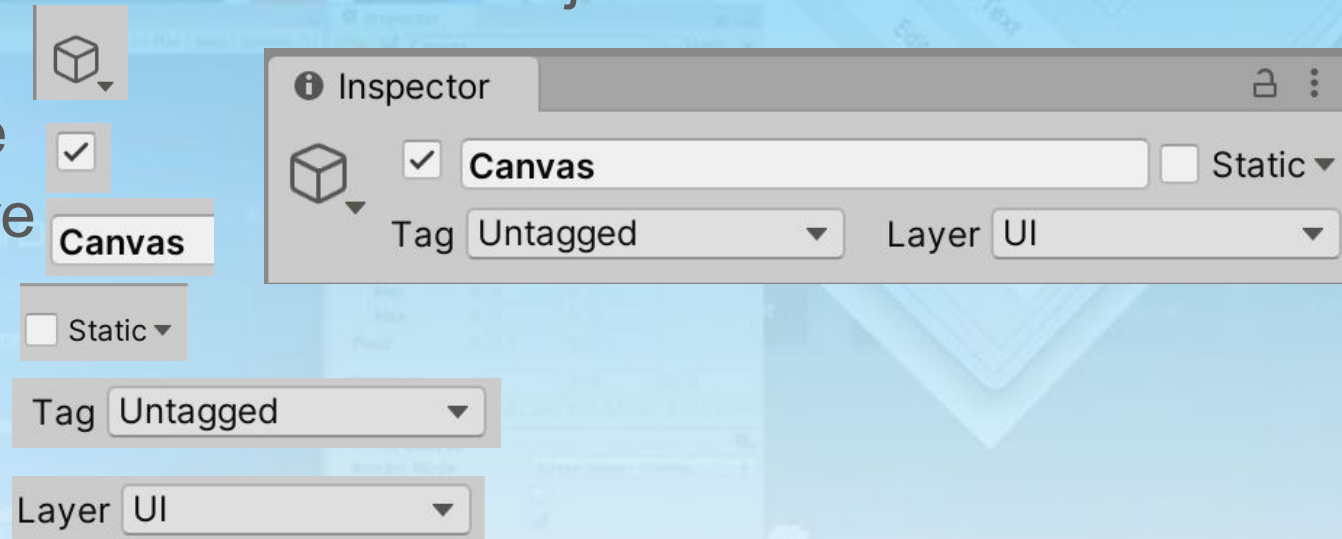
• Canvas

- Veamos las distintas propiedades del GameObject "Canvas".

- Lo primero que nos encontramos son unas propiedades comunes a todos los GameObjects



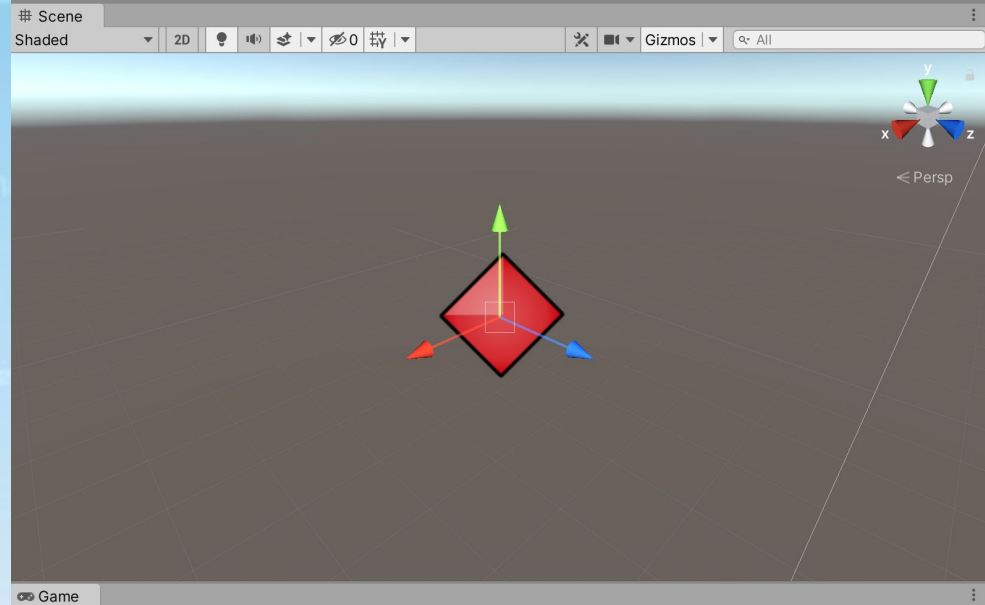
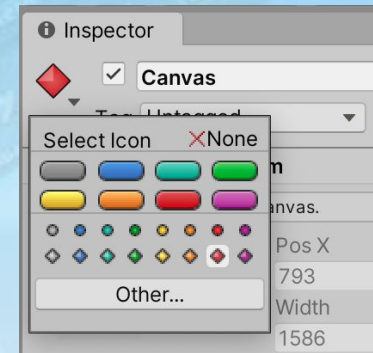
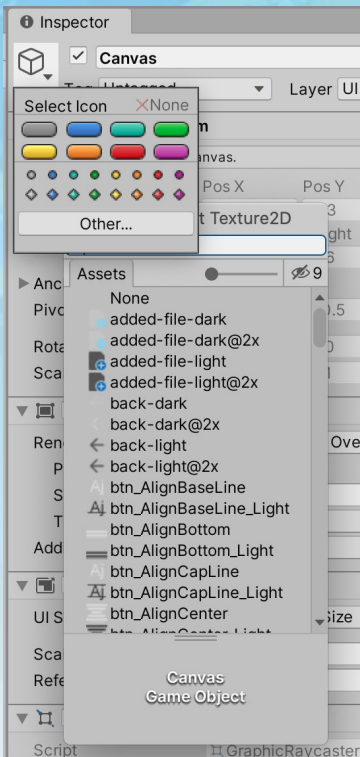
- Gizmo
- Enable
- Nombre
- Static
- Tag
- Layer



• Canvas

• Gizmos.

- Es la representación gráfica del GameObject en la ventana "Scene", podemos seleccionar entre varias opciones (formas, colores, personalizadas,...).

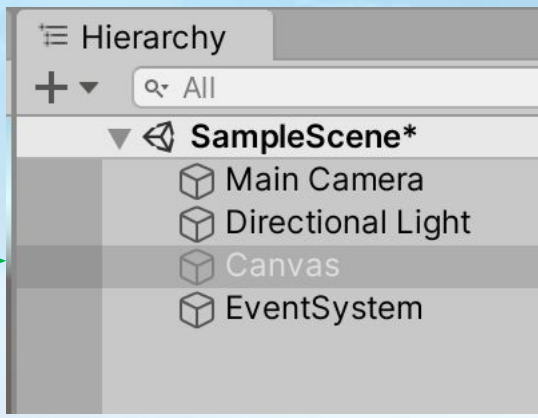
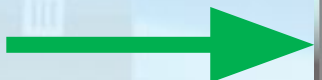
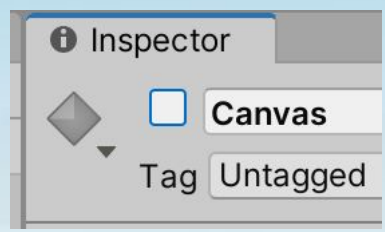
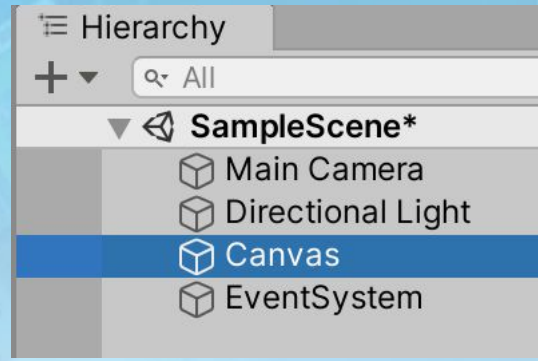
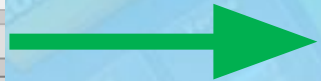
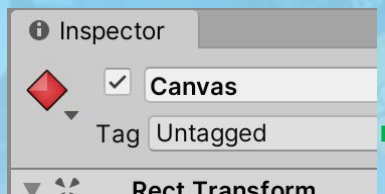


Canvas y Paneles

• Canvas

• Enabled.

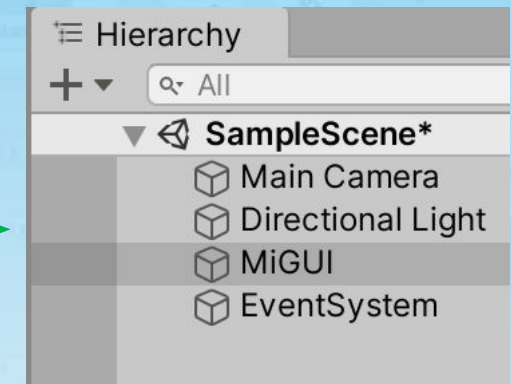
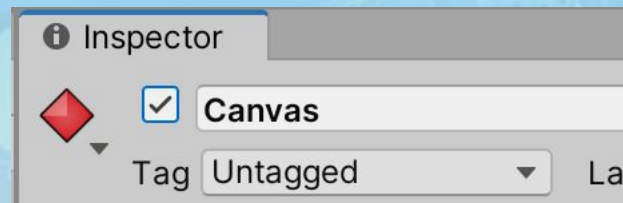
- El objeto está cargado en escena y está habilitado/deshabilitado.



- **Canvas**

- **Name.**

- Desde el que pondremos un nombre al GameObject o le dejaremos el que viene por defecto, en la imagen cambiamos el nombre “Canvas” por “MiGUI”.



Canvas y Paneles

- **Canvas**

- **Static.**

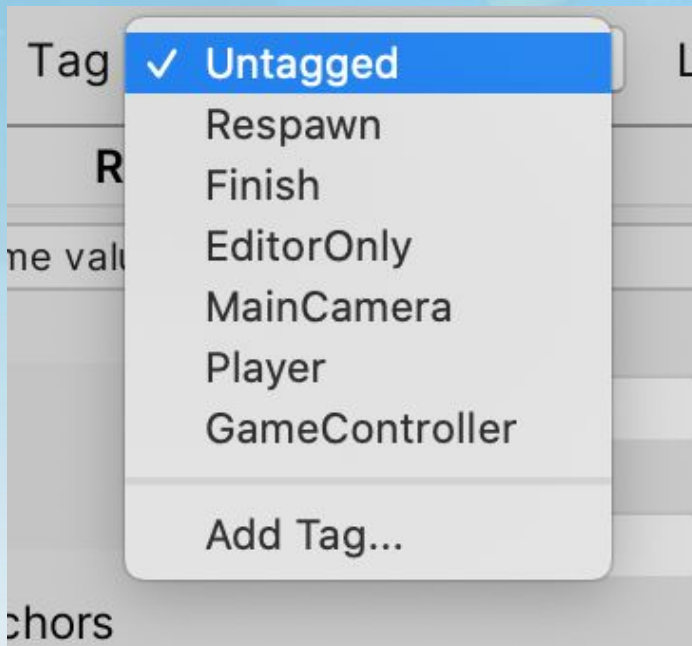
- Esta opción nos sirve para optimizar el proyecto. Si este GameObject no se va a desplazar o cambiar de forma, seleccionamos esta opción.



• Canvas

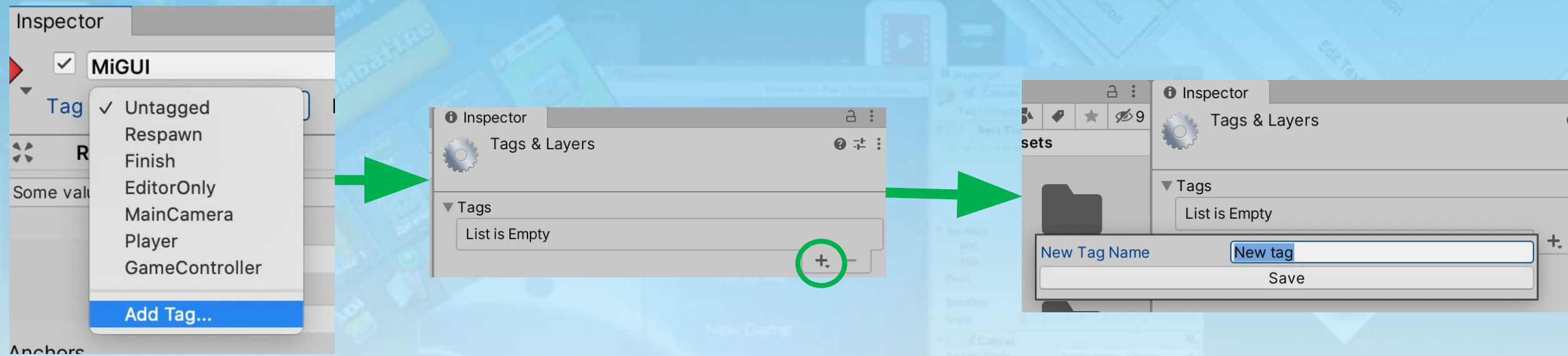
• Tag.

- A los GameObject's se les puede añadir una etiqueta, para poder hacer operaciones desde C#, para agrupar GameObject's, etc. Si desplegamos la lista de Tag, veremos lo siguiente:



Por defecto los GameObject's no tienen etiqueta asignada. Podemos observar en la imagen que Unity nos proporciona etiquetas por defecto (Respawn, Finish, EditorOnly, MainCamera, Player y GameController). Es interesante la última opción **Add Tag...** Esto nos permite crear una etiqueta personalizada.

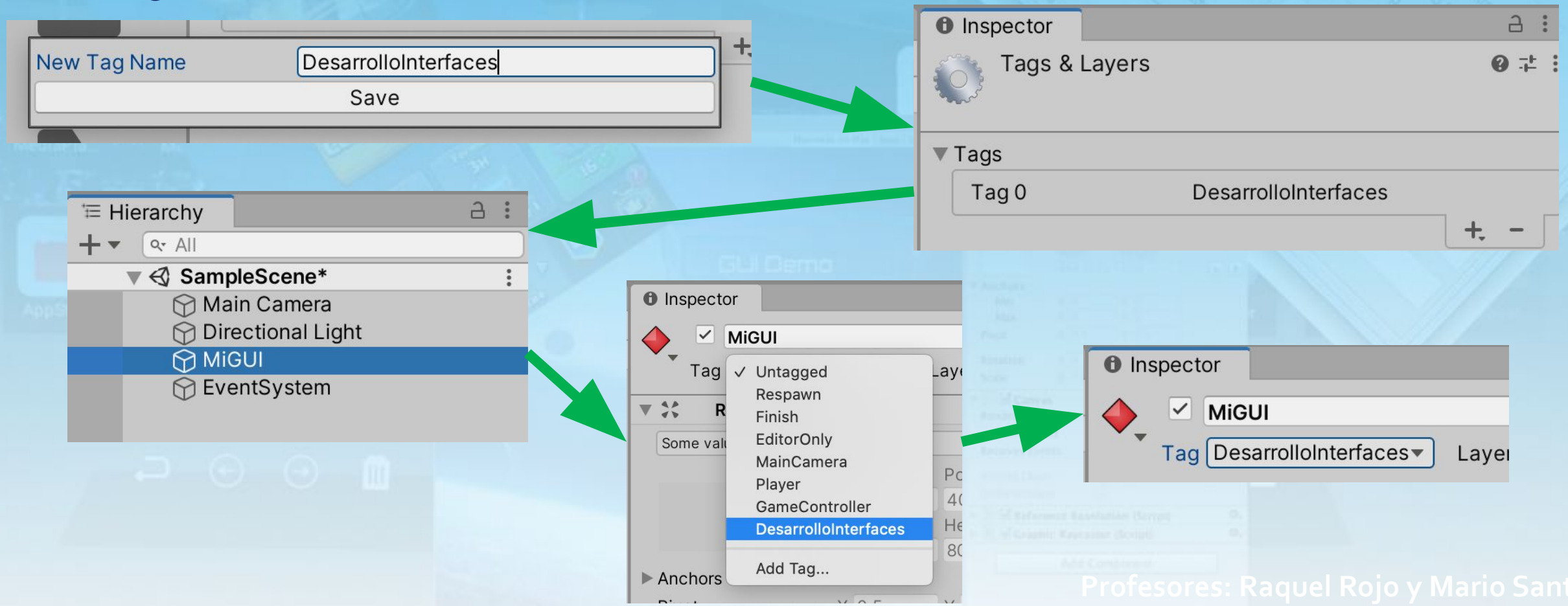
- Canvas
- Tag.



Vamos a crear una nueva etiqueta que se denomine “DesarrolloInterfaces” y se la vamos a insertar a nuestro GameObject “MiGUI”.

Canvas y Paneles

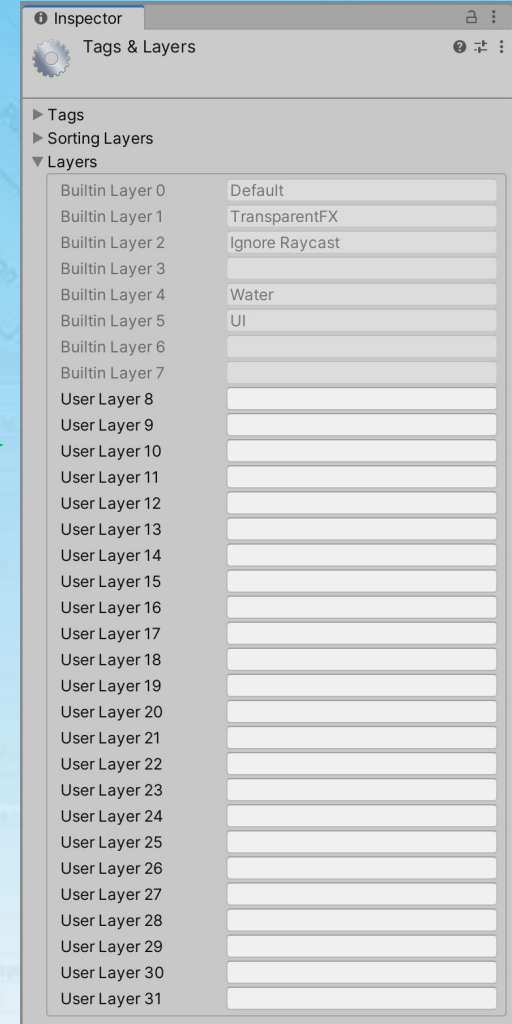
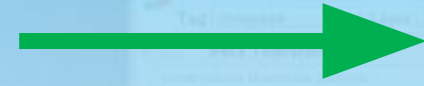
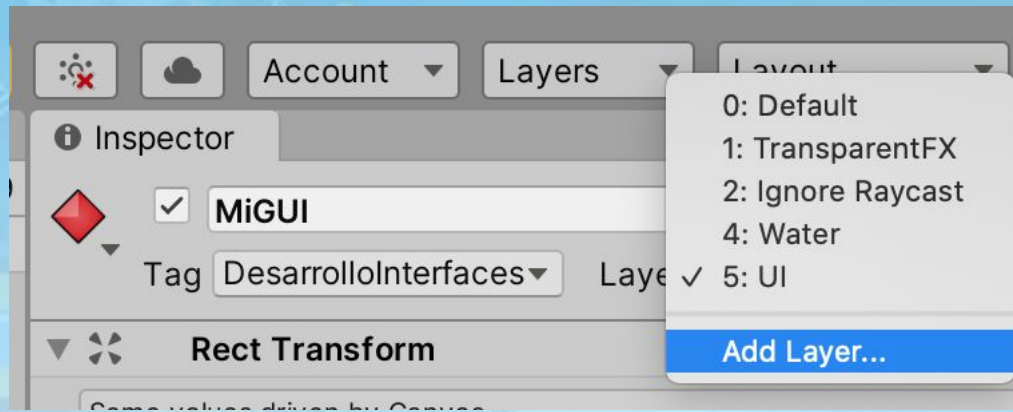
- Canvas
- Tag.



• Canvas

- **Layer.** Similar a los Tag, en este caso, en vez de etiquetas hablamos de Capas. Esto nos permite:
 - Visualizar/Ocultar los elementos que pertenecen a dicha capa.
 - Hacer modificaciones sobre todos los elementos que pertenecen a una determinada capa.
 - Que los GameObject's que pertenecen a una capa, se comporten de una determinada manera frente a un evento (esto lo veremos más adelante).
- Siendo similares el parámetro Tag y Layer, hay algunas diferencias a tener en cuenta:
 - En una Layer, puede haber varios GameObject's con diferentes Tag's.
 - Podemos crear todas las Tag's que queramos, pero solo podemos tener 32 Layers y las 8 primeras, están reservadas para Unity.

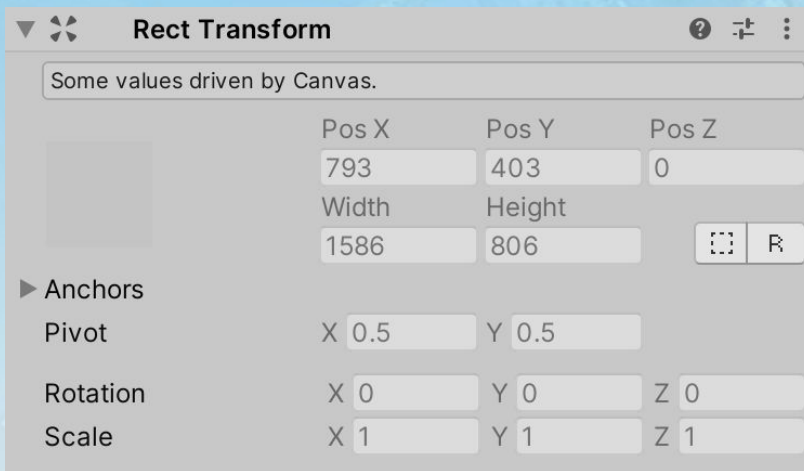
- Canvas
- Layer.



• Canvas

• Rect Transform.

El conjunto de parámetros de **Transform** es común a todos los **GameObject's** de Unity. En él se puede modificar la escala, la posición y la rotación. El **Rect Transform**, hereda del **Transform** y nos permite definir la **ubicación** de un elemento dentro del espacio del Canvas.



Como veremos más adelante, en esta misma presentación, el componente **Rect Transform** aparece desactivado por el tipo de Canvas que está seleccionado por defecto.

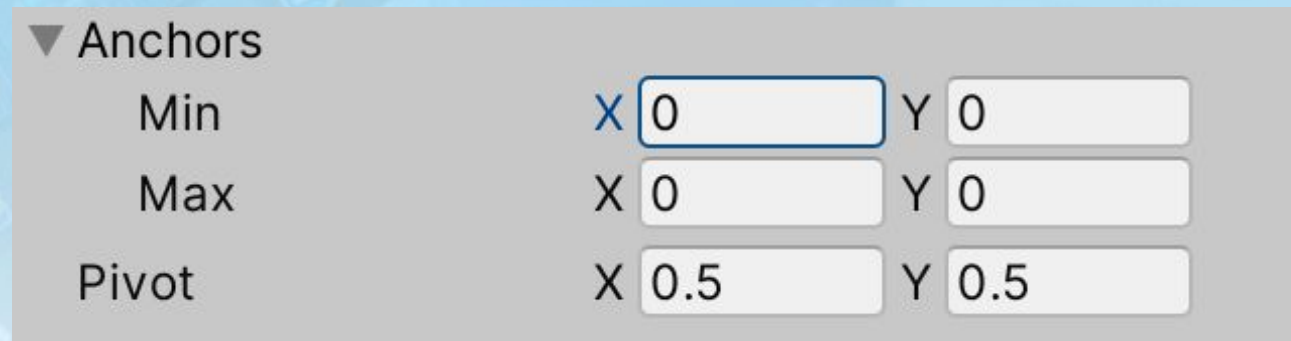


<https://docs.unity3d.com/2020.2/Documentation/Manual/class-RectTransform.html>

- **Canvas**

- **Rect Transform.**

Tenemos las opciones de los **Anchor**s, estos son los puntos de anclaje y el de rotación dentro del Canvas. Lo normal es que los puntos de anclaje estén en X e Y y en la posición (0,0) y el punto de rotación esté en el centro del Canvas. Son más utilizados para los GameObject's que se encuentran dentro del Canvas más que para el Canvas en sí mismo.



▼ Anchors			
Min	X	<input type="text" value="0"/>	Y <input type="text" value="0"/>
Max	X	<input type="text" value="0"/>	Y <input type="text" value="0"/>
Pivot	X	<input type="text" value="0.5"/>	Y <input type="text" value="0.5"/>

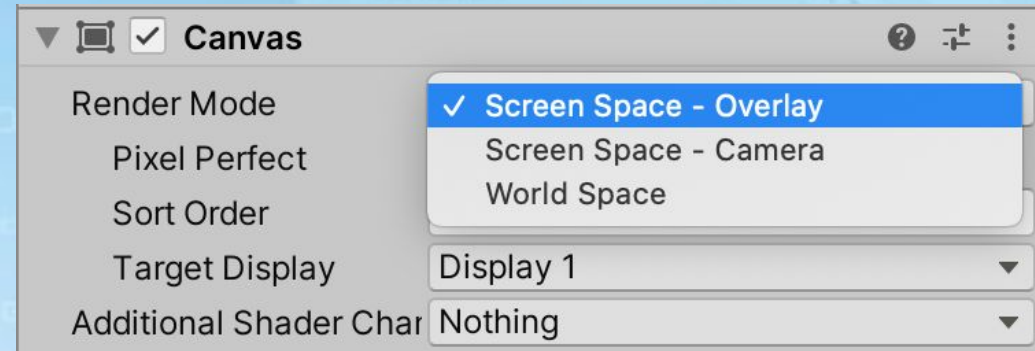
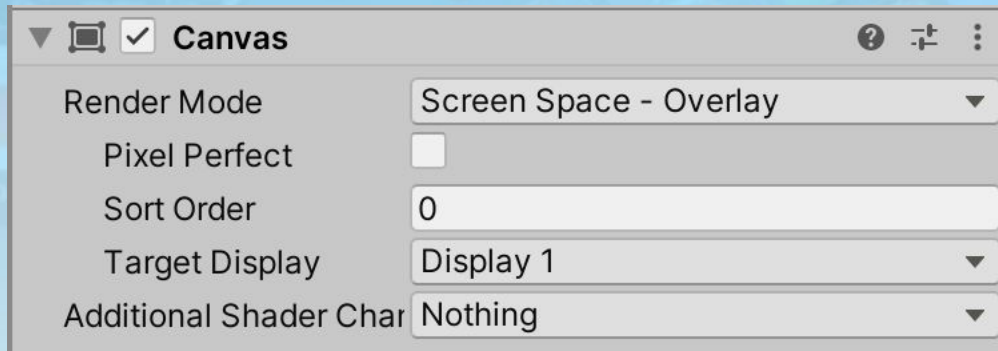


• Canvas

• Canvas.

Este apartado es importante, en él vamos a especificar el tipo de Render que vamos a hacer de nuestro GUI.

En las pantallas siguientes podemos ver las diferentes propiedades a modificar.



• Canvas

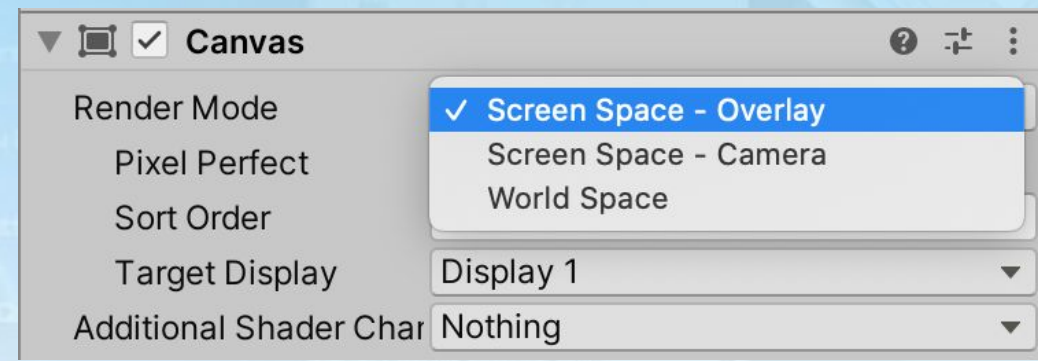
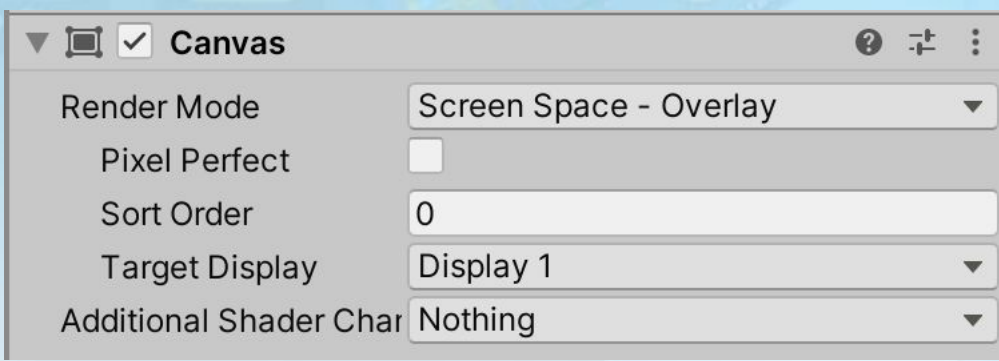
• Canvas.

El primer parámetro que podemos observar es el de “Render Mode”. Veamos las opciones:

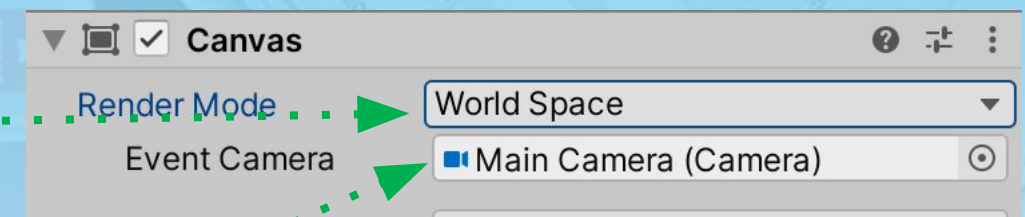
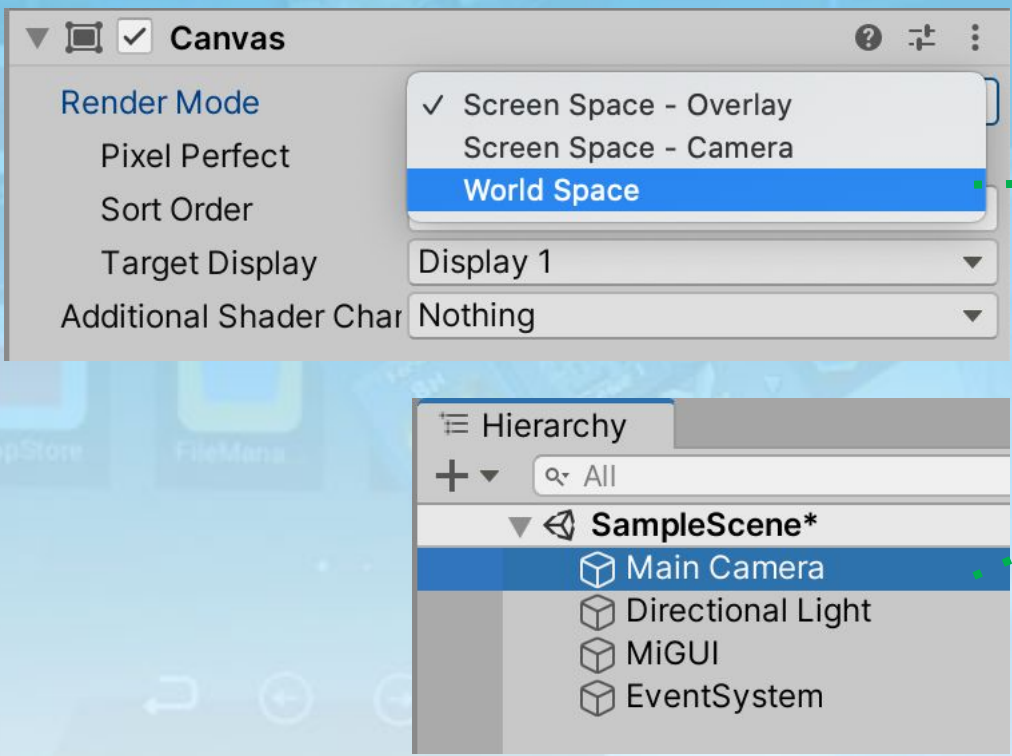
Screen Space – Overlay: Aparece por defecto, y ajusta el Canvas a toda la pantalla, es **resize**.

Screen Space – Camera: Similar a la anterior, pero tiene **profundidad** de cámara. Está asociada a una cámara, en caso de tener varias en la escena.

World Space – Se ajusta a un punto en el espacio 3D. Si activamos esta opción, nos aparecerá un parámetro para cargar la cámara.



- Canvas
- Canvas.



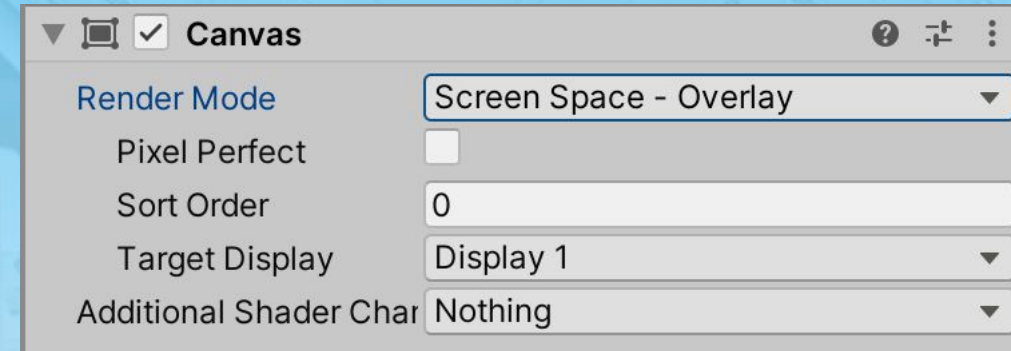
<https://docs.unity3d.com/2020.2/Documentation/Manual/class-Canvas.html>

• Canvas

• Canvas.

Cada “Render Mode” tiene sus parámetros. En el Screen Space – Overlay nos encontramos:

- **Pixel Perfect:** Para activar el antialiasing.
- **Sort Order:** Si tenemos varios Canvas, nos indica en que orden se va a Renderizar.
- **Target Display:** Trabajamos con varias cámaras.
- **Additional Shader Channels:** Podemos trabajar con Shaders (texturas) y asignarles un canal.



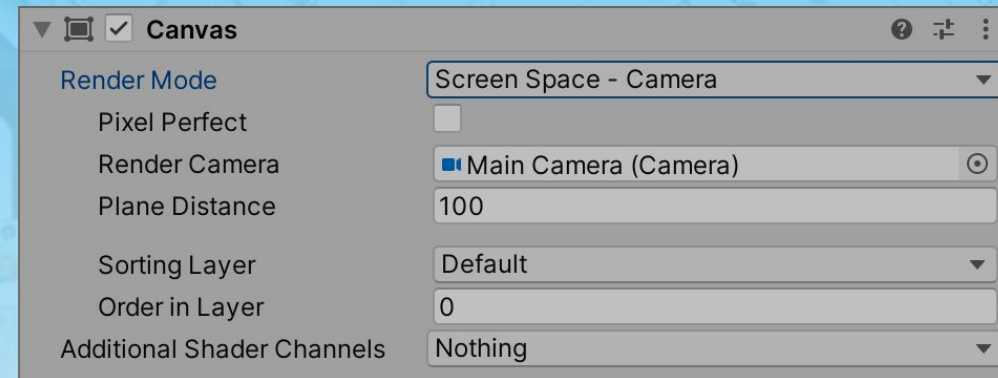
<https://docs.unity3d.com/2020.2/Documentation/Manual/class-Canvas.html>

• Canvas

• Canvas.

En Screen Space – Camera, se nos añaden algunos parámetros como:

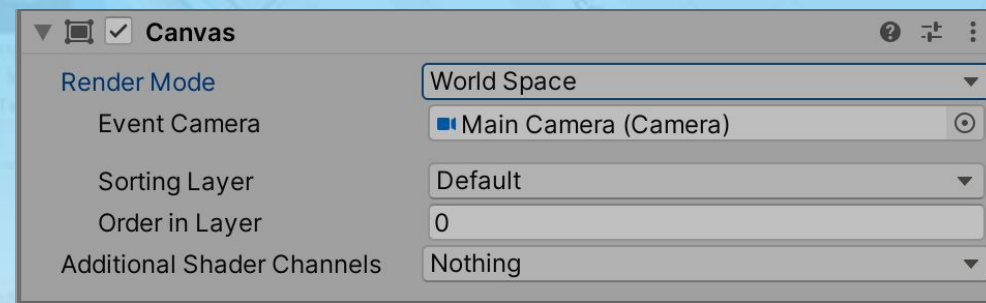
- Render Camera: Debe aparecer la cámara visible para este Canvas.
- Plane Distance: Podemos ajustar la distancia a la que se nos va a mostrar el Canvas. Esto puede hacer que haya GameObject's delante del Canvas.



• Canvas

• Canvas.

En World Space, las opciones son las vistas con anterioridad en los dos Render Mode anteriores.

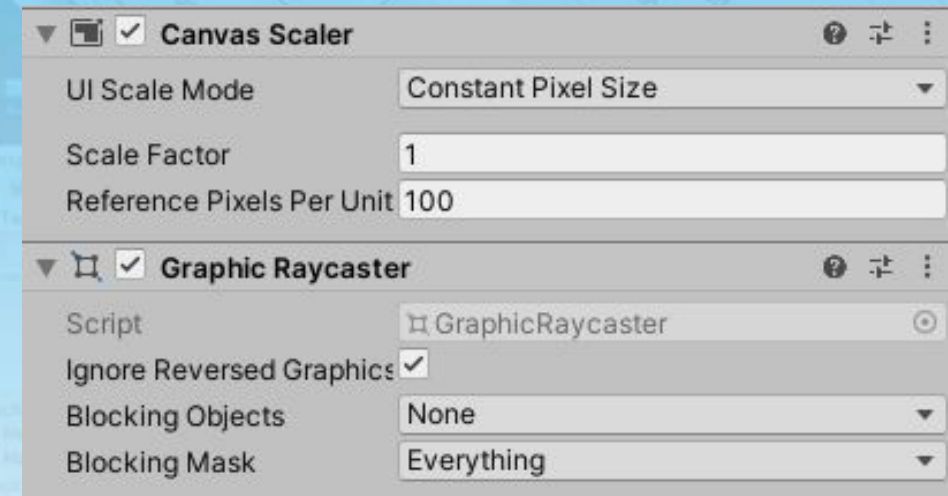


<https://docs.unity3d.com/2020.2/Documentation/Manual/class-Canvas.html>

• Canvas

• Canvas Scaler y Graphics Raycaster.

Estos dos componentes los vamos a utilizar para definir el Factor de Scala y por lo tanto la definición de nuestros componentes GUI (Canvas Scaler) y para decir si este Canvas va a admitir el RayCast (Graphics Raycaster). Esto lo veremos más adelante, en la U.T. dedicada a la VR.



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

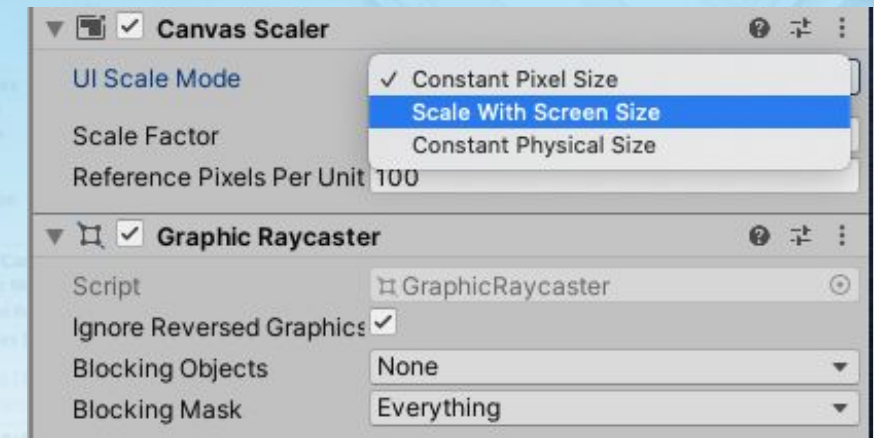
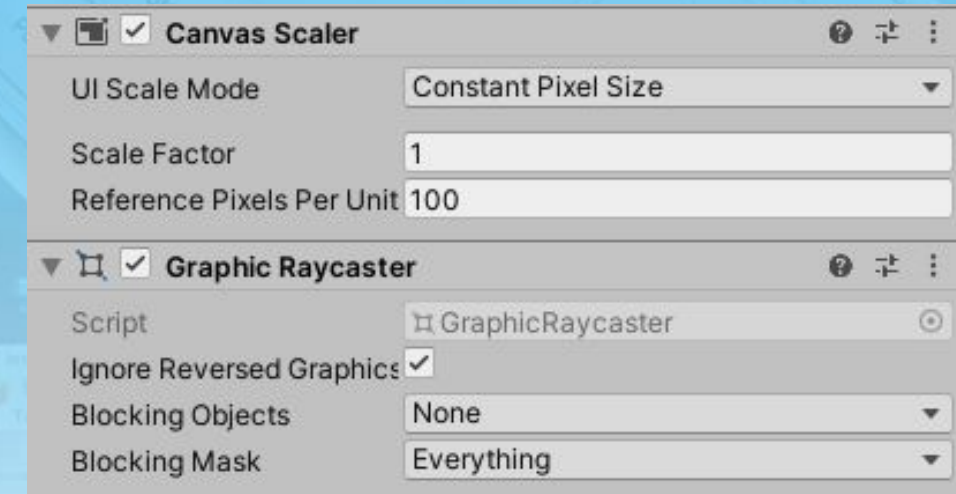
• Canvas

• Canvas Scaler y Graphics Raycaster.

Para el Desarrollo de Interfaces, es de importancia el parámetro UI Scale Mode, que nos va a mostrar tres opciones.

- **Constant Pixel Size:** Define un tamaño de manera estática. Muy útil cuando se desarrolla únicamente para un dispositivo (por ejemplo una consola o unas gafas de un modelo concreto)
- **Scale With Screen Size:** Se va a redimensionar al dispositivo en el que se ejecute nuestro Canvas.

Recomendamos utilizar Scale With Screen Size

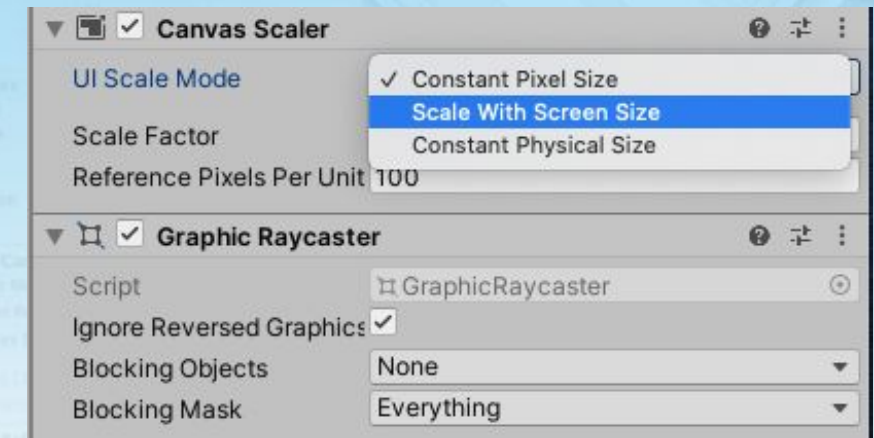
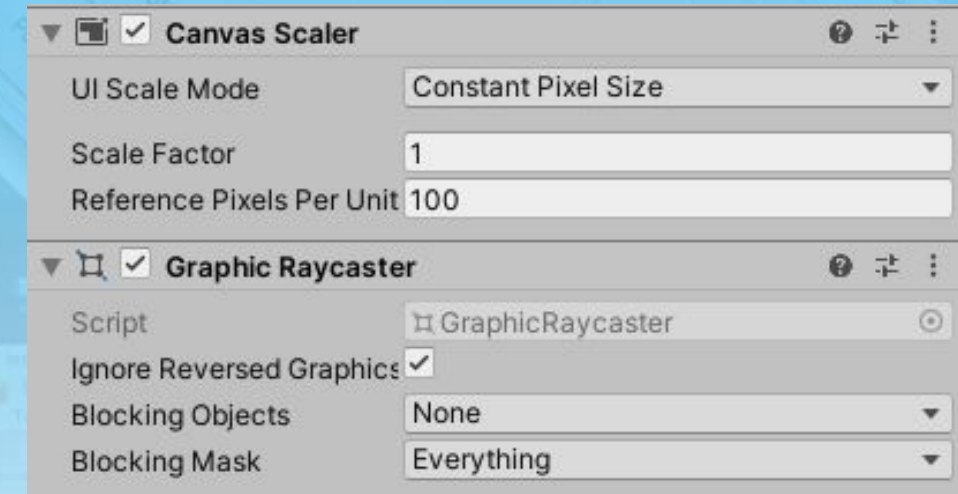


• Canvas

• Canvas Scaler y Graphics Raycaster.

Para el Desarrollo de Interfaces, es de importancia el parámetro UI Scale Mode, que nos va a mostrar tres opciones.

- **Constant Physical Size:** Hace que los elementos de la interfaz de usuario conserven el mismo tamaño físico independientemente del tamaño y la resolución de la pantalla. Útil para que algunos elementos, como botones no aparezcan demasiado pequeños/grandes, por ejemplo, en pantallas de móviles.



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

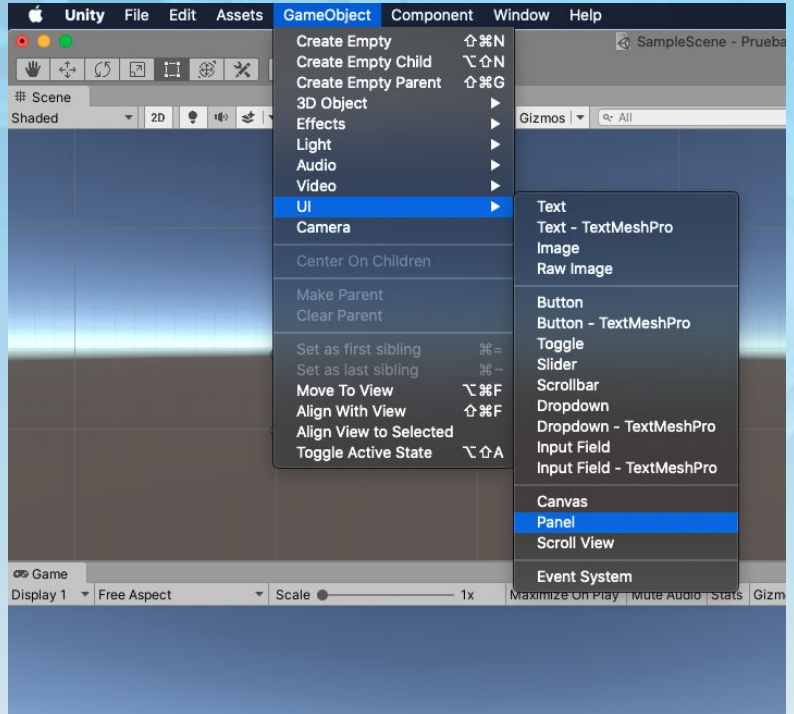
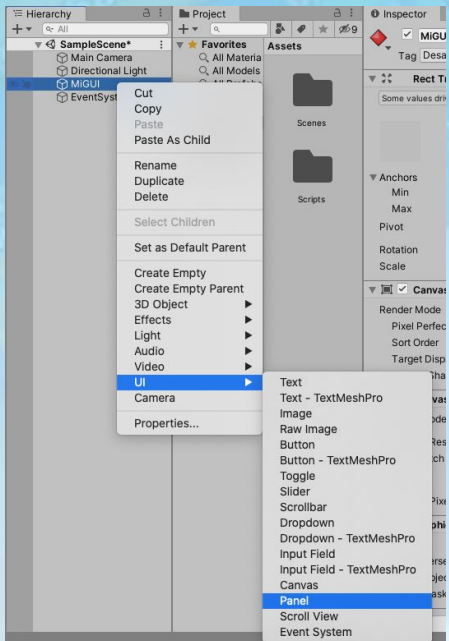
• Paneles

Al igual que los Canvas, los Paneles son contenedores. La diferencia está en:

- Un Canvas, contiene paneles, pero un panel, nunca contiene un Canvas.
- Utilizamos los Paneles para agrupar GameObject's de la GUI (botones, texto, imágenes, etc) que tienen un significado o una función concreta.
- En los paneles podemos utilizar colores e imágenes (Sprites).
- Los paneles, se pondrán siempre dentro del espacio delimitado por los Canvas.
- Cuando cargamos un GameObject de tipo "Panel" en nuestra ventana, siempre aparecerá como hijo de un Canvas. Si no hemos creado y asociado a un Canvas, Unity introduce uno por defecto y lo asocia.

• Paneles

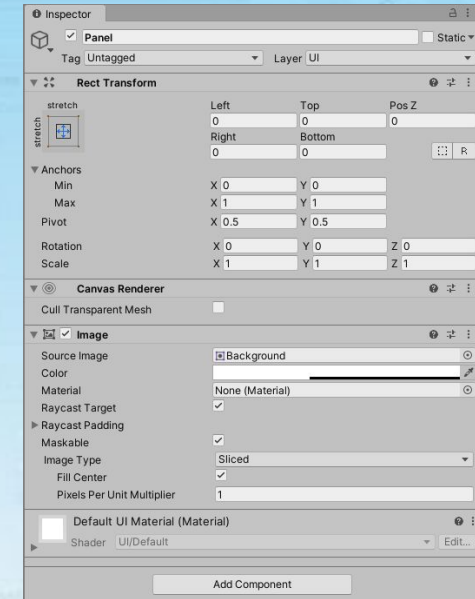
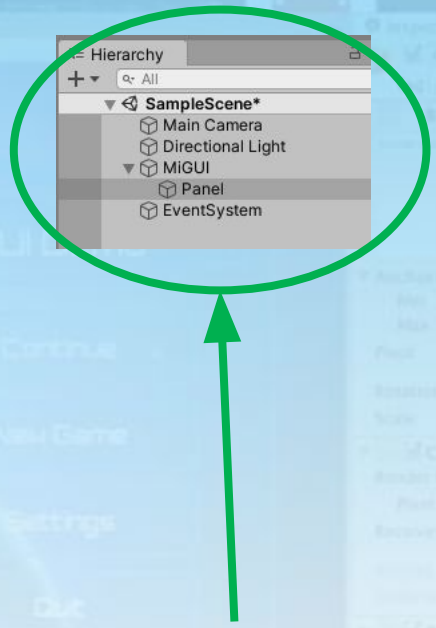
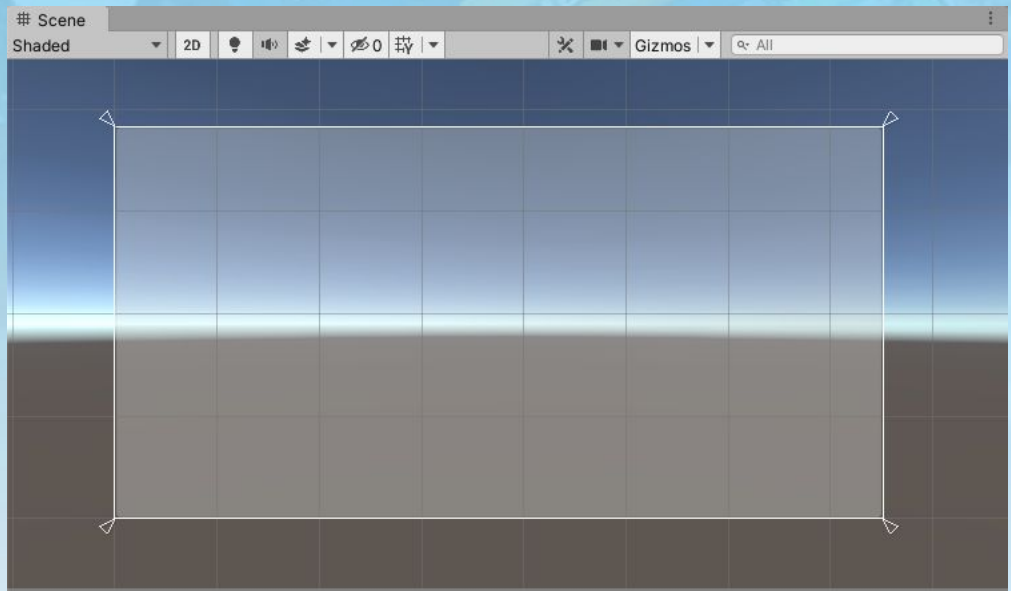
Para insertar un Panel, seguiremos los mismos pasos que para insertar un Canvas, es decir, desde el menú contextual de la ventana Hierarchy o desde la GameObject, situado en la barra de menús.



Es importante que si introducimos un Panel desde el menú contextual, esté seleccionado el Canvas que lo va a contener, de esta manera, Unity lo emparenta directamente.

• Paneles

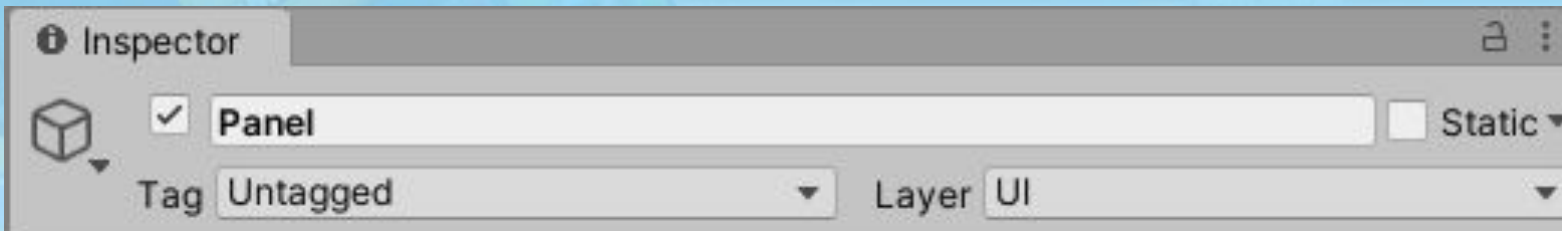
Una vez insertado el Panel, nos encontramos con, el panel en la venta Scene, en la ventana Hierarchy y sus propiedades en el Inspector.



Fijarse como ha emparentado el Panel dentro de nuestro Canvas que se llama MIGUI.

• Paneles

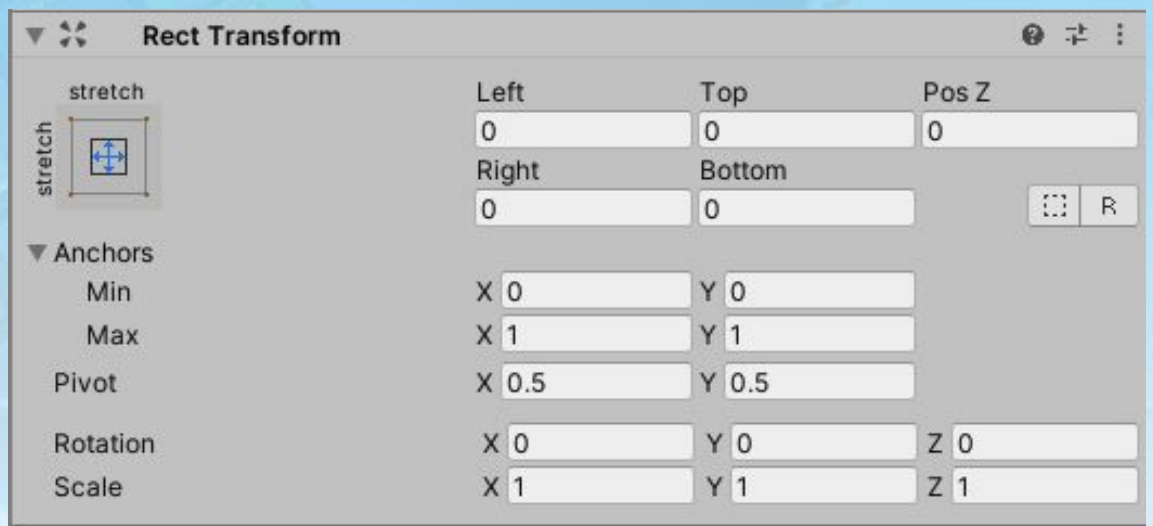
Los parámetros de la ventana inspector, son similares a los que hemos visto con el Canvas.



La parte de los Gizmos, habilitado/deshabilitado, Nombre, Static, Tag y Layer, funciona igual que hemos visto con los Canvas.

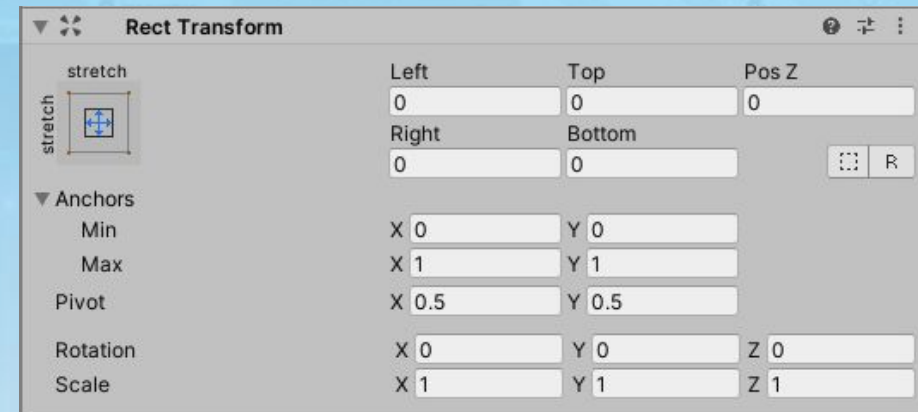
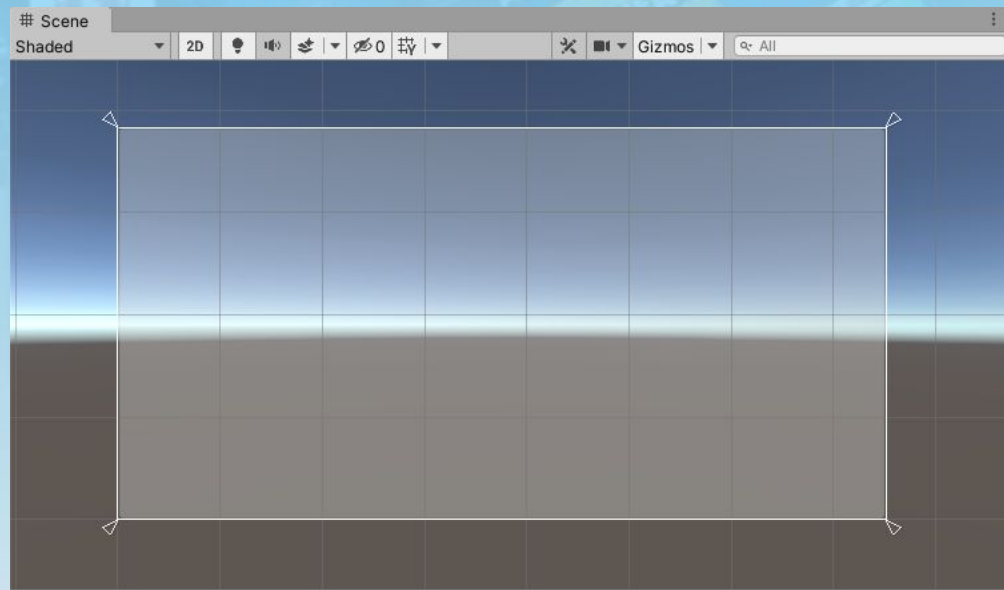
• Paneles

La parte del Rect Transform, también es igual que en el Canvas, pero en este caso, nos aparece habilitado por si queremos modificar la posición de los **Anchors**, la **Scala**, la **Position** de panel, la **Rotate**.



• Paneles

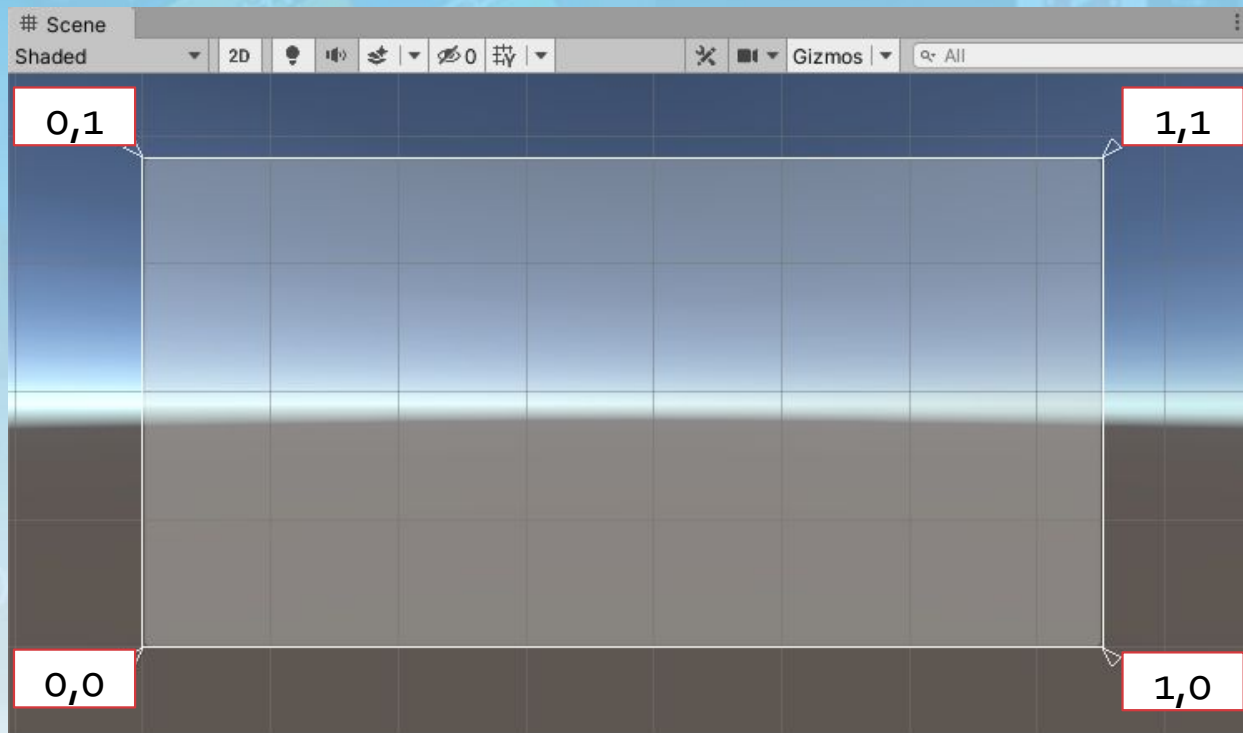
Los Anchors, vienen representados en la ventana Scene como unos triángulos pequeños. En este caso, aparecen en cada una de las esquinas del Panel.



Los Anchors son componentes de diseño que nos van a servir para posicionar los objetos que se encuentren dentro de este contenedor.

• Paneles

Es importante entender el sistema de coordenadas sobre X e Y, de los Anchors, la esquina superior derecha es el punto (1,1) y la esquina inferior izquierda es el punto (0,0).



• Paneles

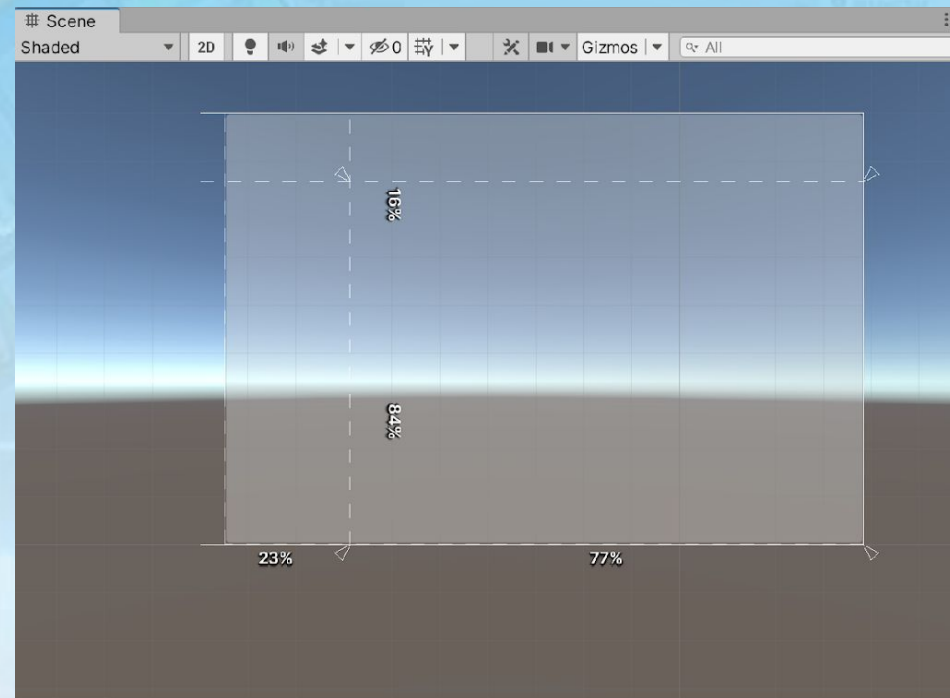
Si se pulsa sobre la imagen “stretch”, nos aparecerá una forma predeterminada de colocar los Anchors.



Recordar que los Anchors son puntos de anclaje para los GameObject que van a estar dentro de este contenedor.

• Paneles

Se puede modificar la posición de los Anchors de forma manual, posicionando el cursor del ratón encima del Anchor que queremos modificar, dejando pulsado el botón izquierdo del ratón y soltándolo en la posición deseada.

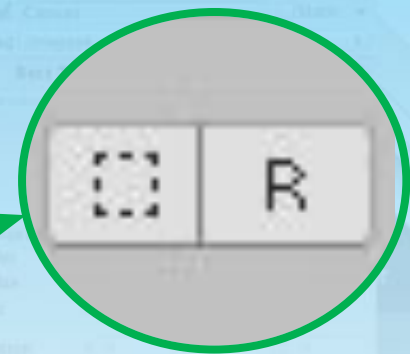
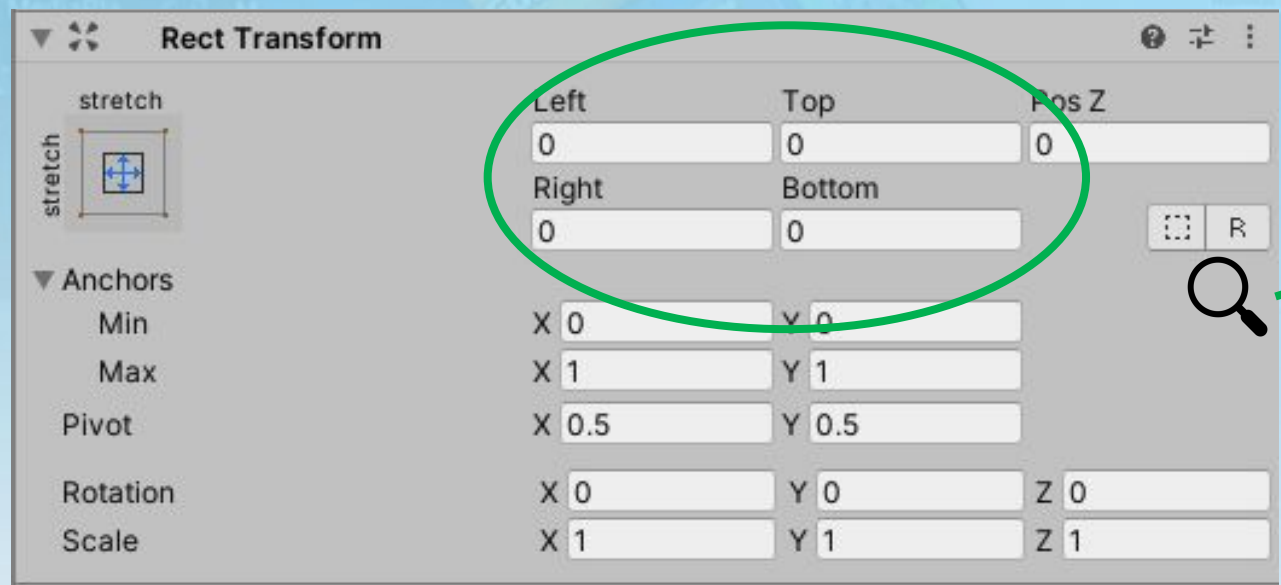


• Paneles

Con los parámetros Left, Top, Right y Bottom, podemos redimensionar y posicionar el panel dentro del Canvas.



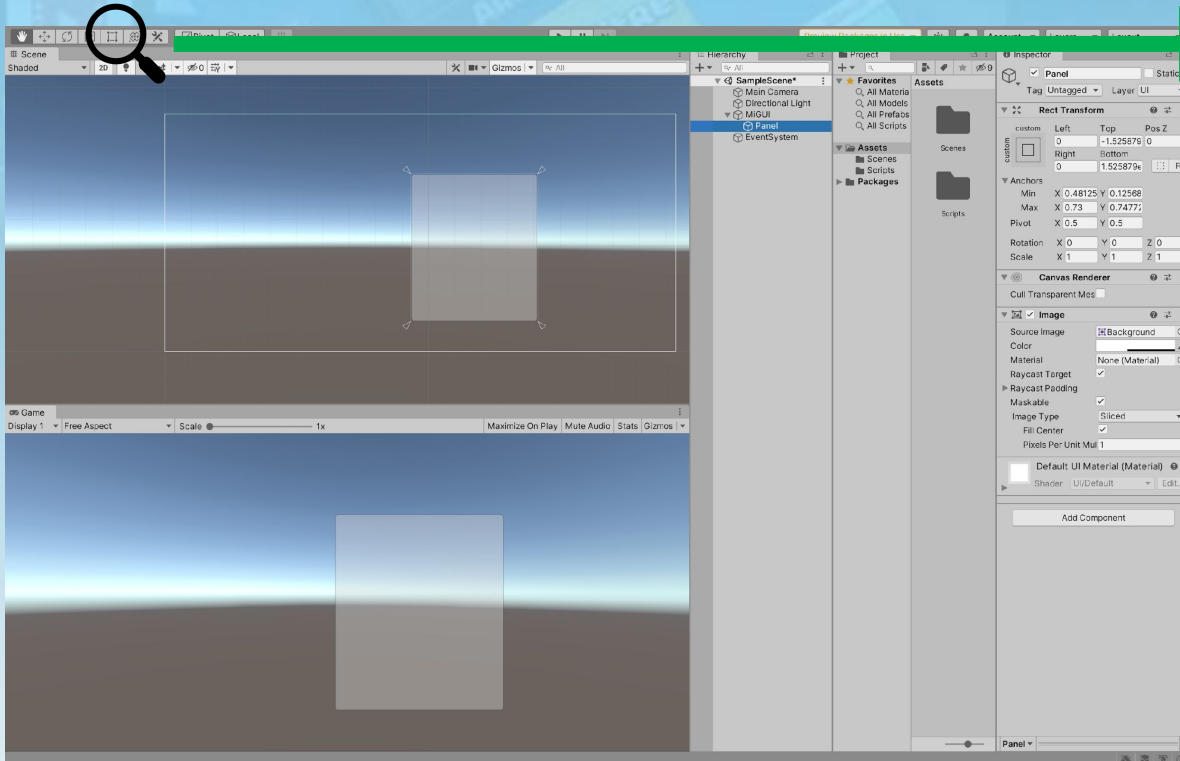
Con los parámetros X,Y,Z de la propiedad Scale, también podemos modificar la escala del Panel.



Estos dos botones, el Blueprint y R (modo de edición sin formato) los veremos en esta U.T. cuando veamos los Buttons y los TextBox.

• Paneles

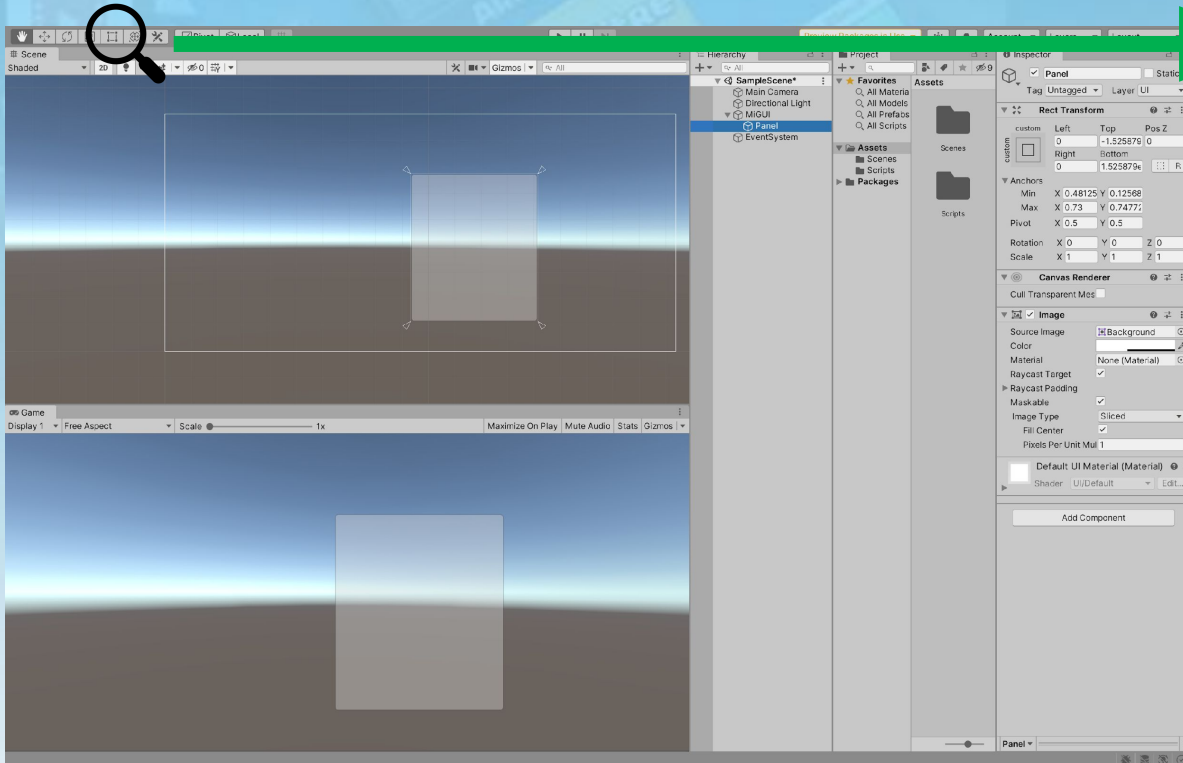
Al igual que con los Anchors, podemos modificar el tamaño y la posición del Panel utilizando las siguientes herramientas.



- El primer botón (la mano), nos permite hacer un paneo. Pulsado el botón derecho y moviendo el ratón sobre la escena.
- El segundo botón (cruz de flechas), nos permite desplazar sobre los ejes el GameObject seleccionado. Pulsando sobre las flechas.
- El tercer botón (flechas en círculo), nos permite rotar el GameObject seleccionado. Seleccionando el color asociado al eje (azul para la Z, verde para la Y y rojo para la X)
- El cuarto botón, nos permite escalar el GameObject seleccionado. Podemos seleccionar solo un eje o escalarlo en todos, si seleccionamos el cuadrado central.

• Paneles

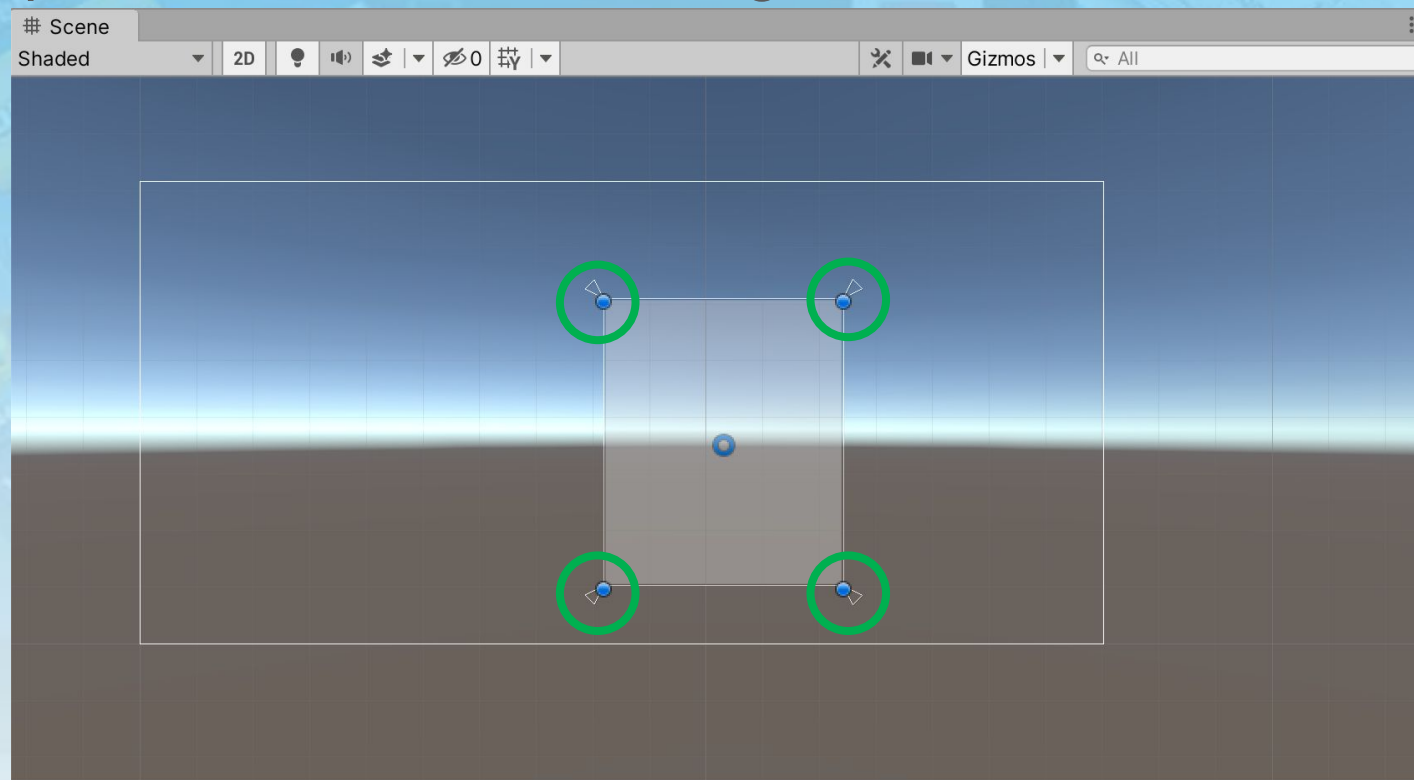
Al igual que con los Anchors, podemos modificar el tamaño y la posición del Panel utilizando las siguientes herramientas.



- El quinto botón nos deja redimensionar (Cambiar el tamaño) el GameObject seleccionado con mayor libertad. Fíjate como cambia el tamaño Width y Height.
- El sexto botón, es un conjunto en el que aparecen todas las herrameintas anteriores.
- El último botón es por si tenemos herramientas personalizadas. No accesible al inicio.

• Paneles

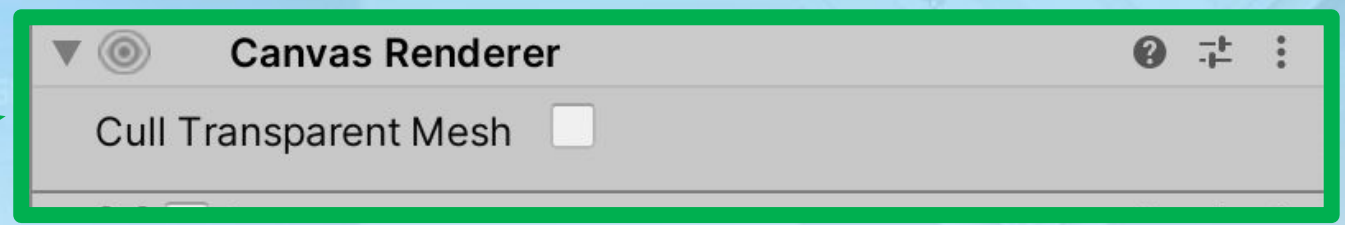
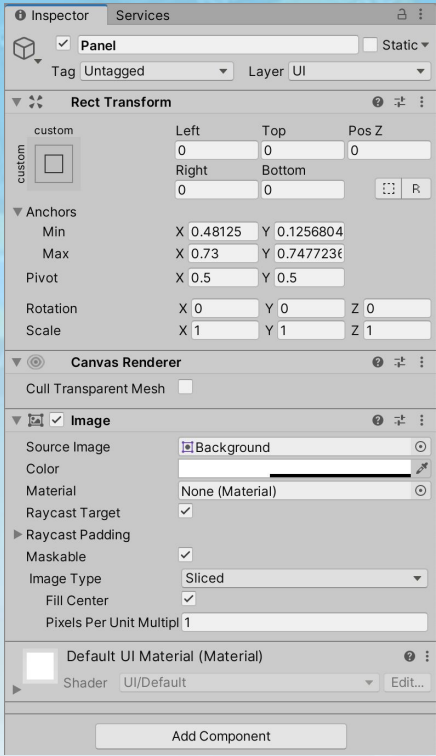
Se puede ajustar los Anchors a la superficie del contenedor y el punto de rotación, centrado, como se puede observar en la imagen.



• Paneles

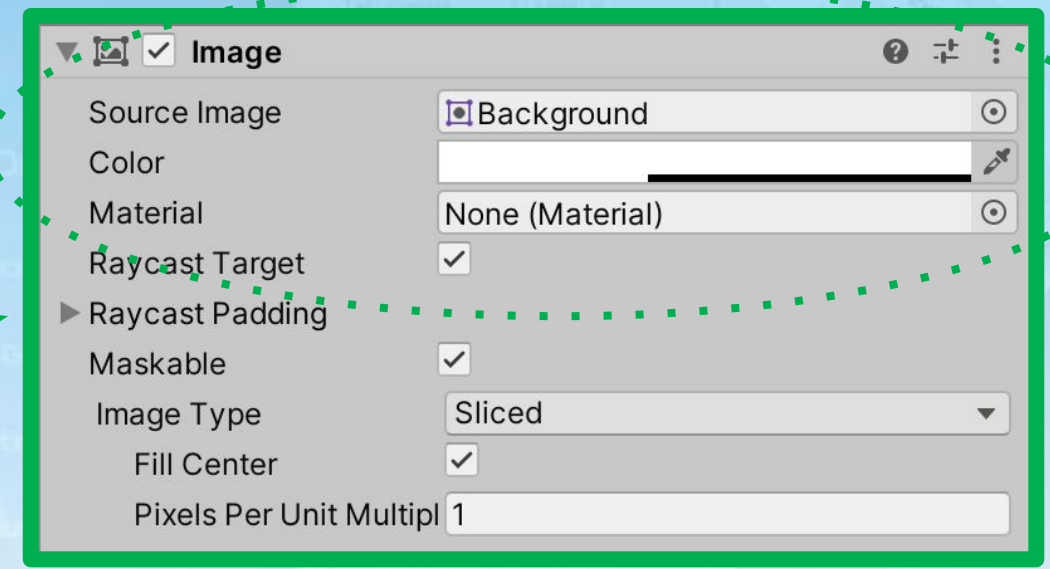
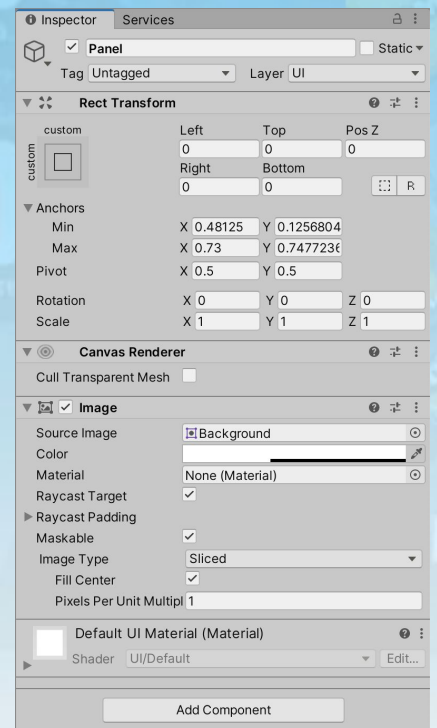
El siguiente componente que nos encontramos en la ventana Inspector es el de **Canvas Renderer**. Este es el Script que hace posible que se visualice el Panel.

propiedades accesibles desde el editor.



• Paneles

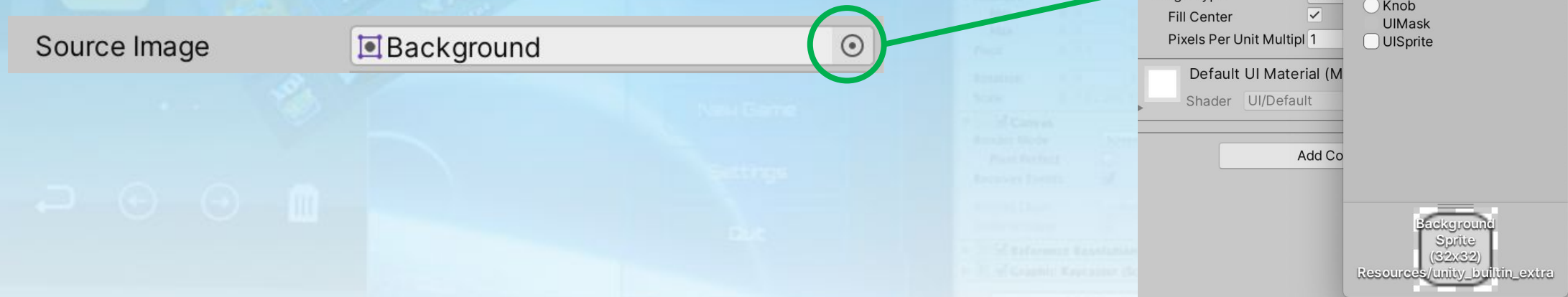
El siguiente componente que nos encontramos y que vamos a utilizar mucho, es el de **Image**. Desde este conjunto de parámetros, vamos a poder dar forma a nuestro panel, modificando el color, la imagen y la textura.



• Paneles

• Source Image.

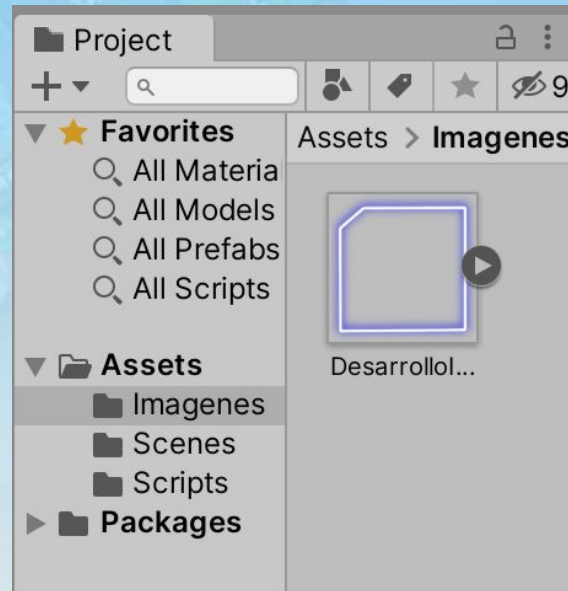
Desde este parámetro podemos introducir una imagen de tipo "Sprite", para personalizar nuestro Panel. Si pulsamos sobre el botón circular, situado a la derecha de este parámetro, se nos mostrará una lista con todos los Sprites disponibles en el proyecto y que podemos utilizar.



• Paneles

• Source Image.

Si creamos un Sprite y lo cargamos a nuestro proyecto, lo podremos utilizar. Vamos a crear un Sprite que se denomina DesarrolloInterfaces. Una vez creado, lo cargamos en una carpeta que se denomina Imágenes.



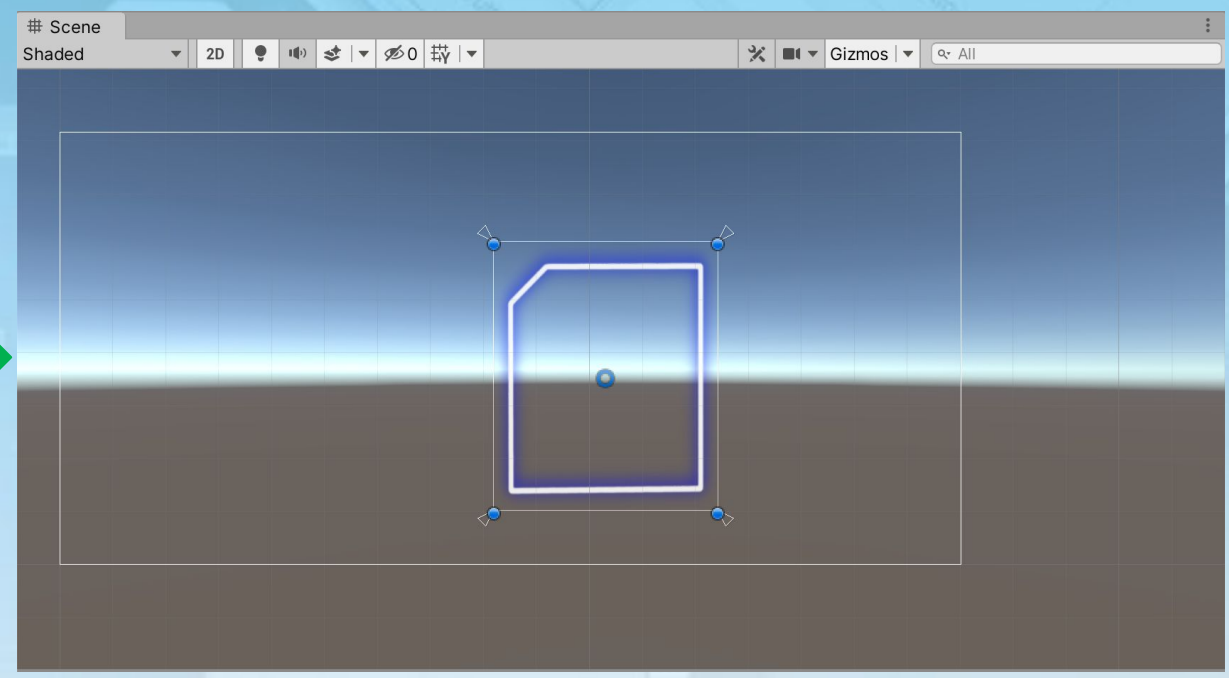
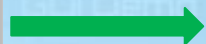
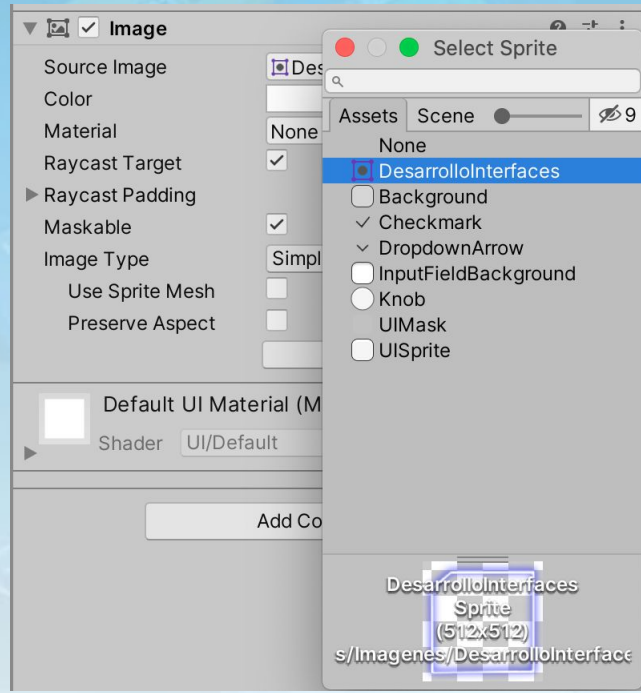
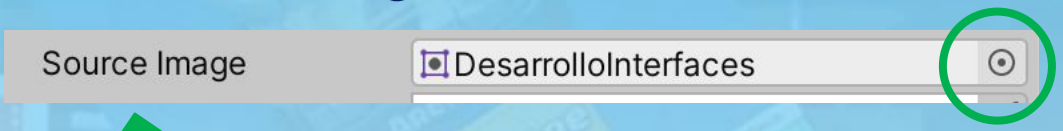
Para crear nuestros paneles personalizados, utilizando Sprites, ver el vídeo: **UT2-CreacionSpritesColorMateriales**, en el que se explica como crearlo utilizando software libre, y los pasos de configuración una vez cargado en el proyecto.



Ver el vídeo UT2-CreacionSpritesColorMateriales

Profesores: Raquel Rojo y Mario Santos.

- Paneles
- Source Image.



Ver el vídeo [UT2-CreacionSpritesColorMateriales](#)

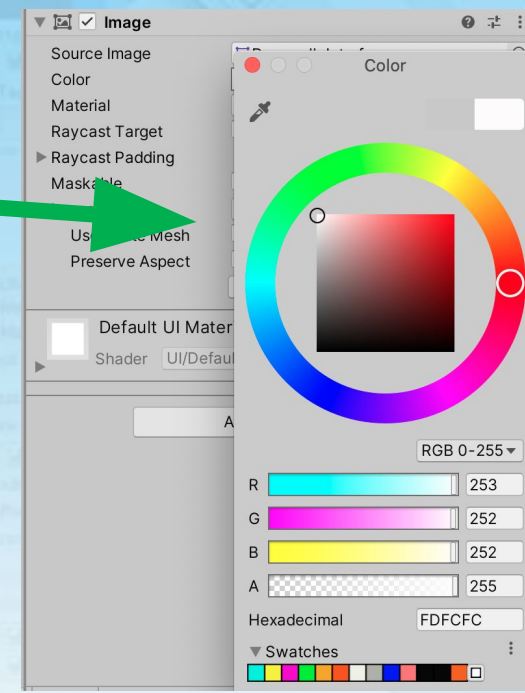
Profesores: Raquel Rojo y Mario Santos.

- **Paneles**
- **Color.**

Desde este parámetro, podremos seleccionar un color para nuestro Panel, e incluso, modificar el color de un Sprite, y modificar el canal alfa (transparencia)



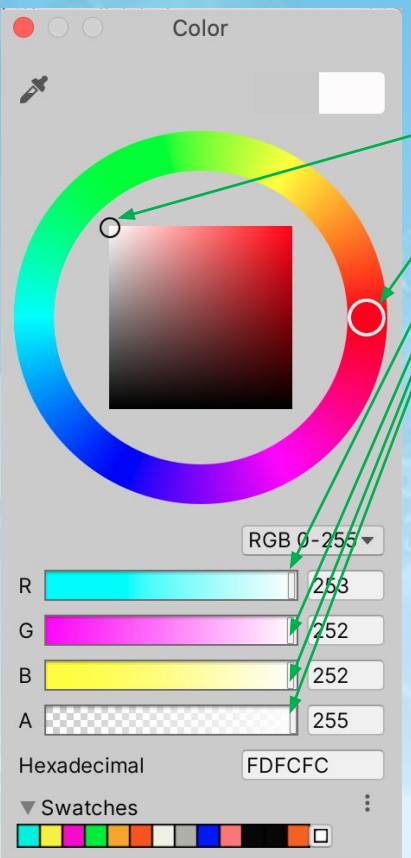
Pulsar con el botón izquierdo del ratón, encima del cuadro blanco.



Ver el vídeo UT2-CreacionSpritesColorMateriales

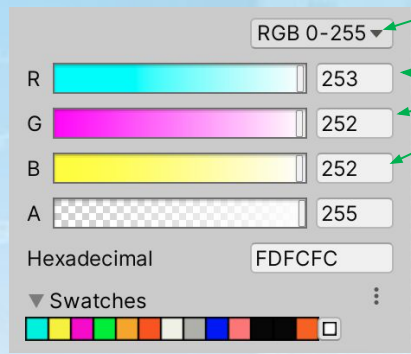
Profesores: Raquel Rojo y Mario Santos.

- Paneles
- Color.



Podemos seleccionar de una forma rápida, con el cursor del ratón, el color buscado, arrastrando cualquiera de los tiradores en forma de círculo o de barra deslizadora.

O podemos introducir nosotros directamente los valores, de 0-255, o de 0-1 o en HSV.



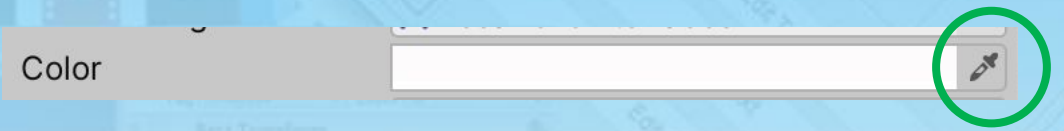
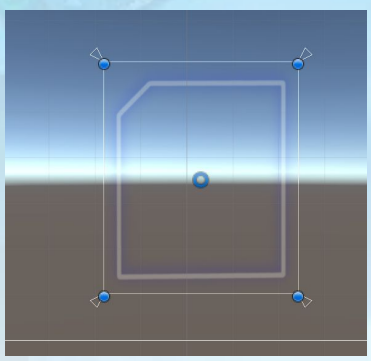
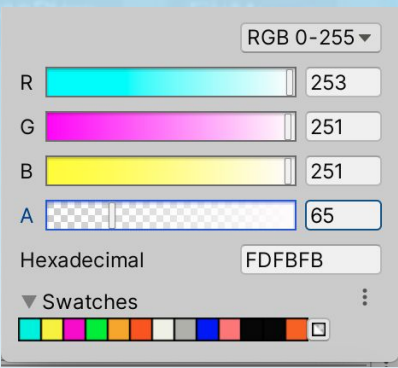
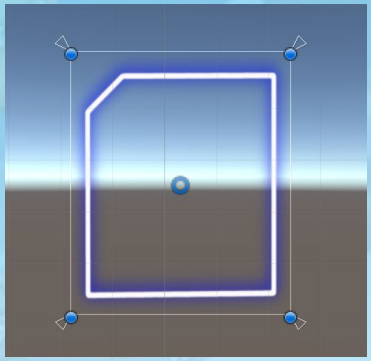
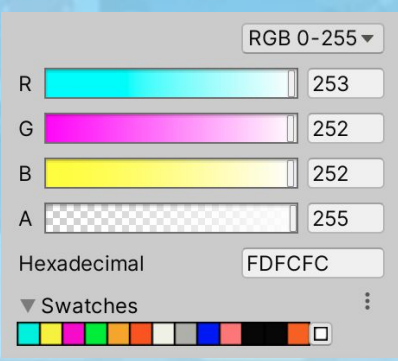
Formato del color.

Valores entre 0-255 de los parámetros RGB (red, Green, blue).

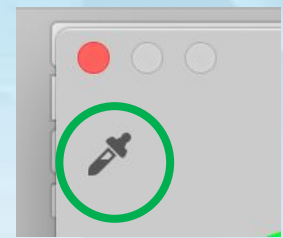


- **Paneles**
 - **Color.**

El parámetro A (alpha) nos da la transparencia del Sprite 0 transparente, 255, opaco.



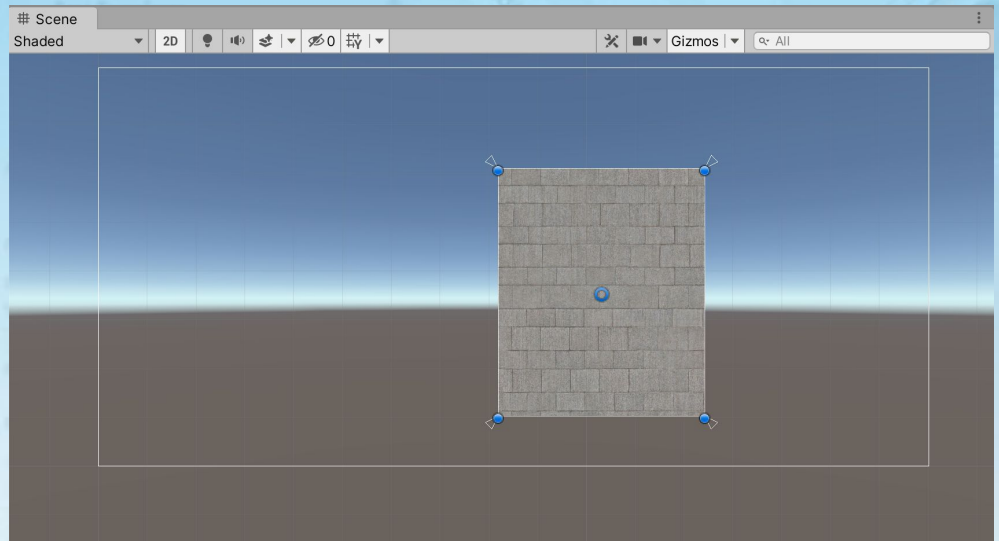
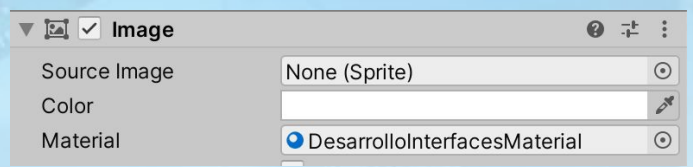
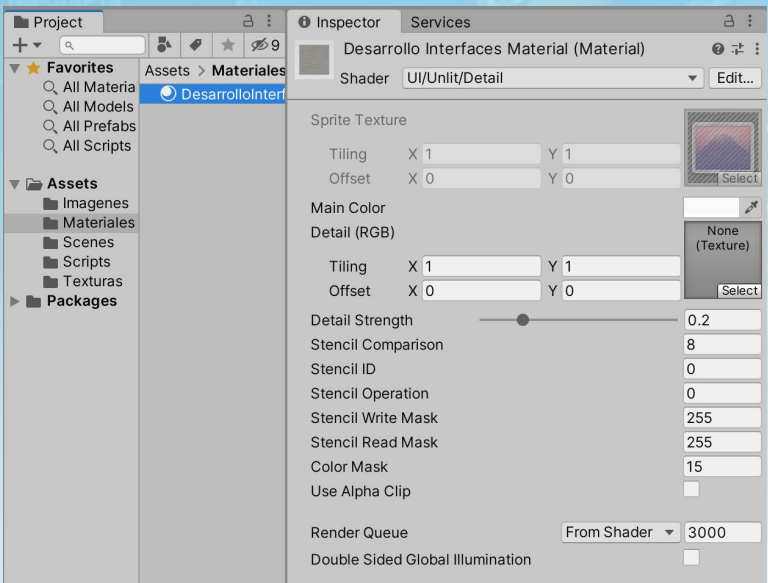
Con esta herramienta se puede seleccionar de forma automática el color de un pixel que tengamos en pantalla.



Ver el vídeo UT2-CreacionSpritesColorMateriales

- Paneles
- Material.

Por último, el parámetro Material, en el podemos **crear un material** e insertar una textura a nuestro Panel. Para que esté todo ordenado en el proyecto, la textura se guarda dentro de una carpeta denominada **Texturas** y los materiales que creamos, dentro de otra carpeta denominada **Materiales**.



Ver el vídeo UT2-CreacionSpritesColorMateriales

Profesores: Raquel Rojo y Mario Santos.

- **Paneles**
 - **Material.**

El resto de parámetros a configurar los veremos más adelante cuando trabajemos la U.T. De VR.



Ver la presentación
U.T.2_BotonTextField