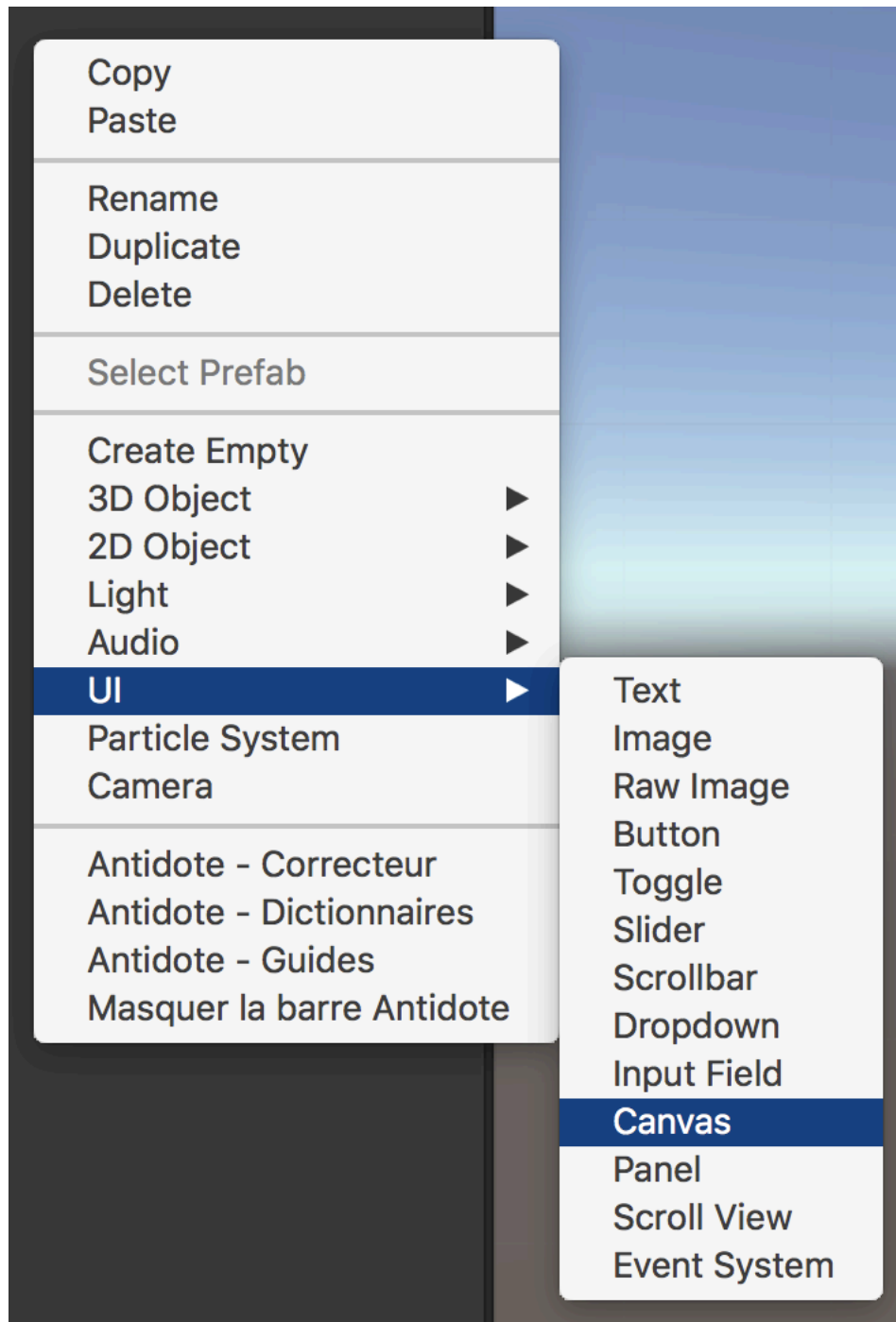


VSSHARE

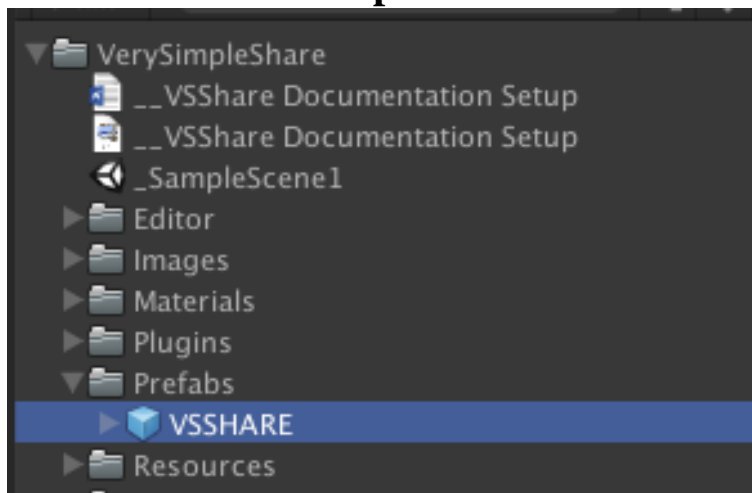
QUICK START :

1) Create a new Canvas if there is no Canvas in your scene.

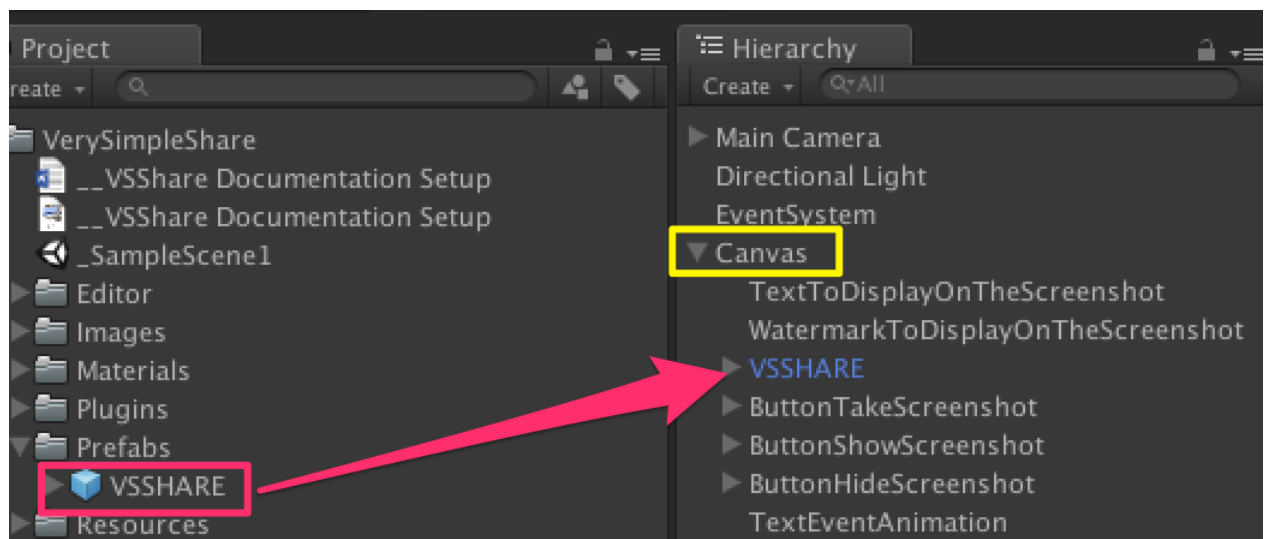
To create a new one :



2) Find the VSSHARE prefab :



3) Drag and drop the VSShare prefab in the scene as a child of the Canvas :



4) Take the screenshot :

Always import the sharing system at the top of your script to be able to call the methods :

```
using UnityEngine;  
using System.Collections;  
using AppAdvisory.SharingSystem;
```

To take a screenshot, you have to call this method :

```
VSSHARE.DOTakeScreenShot();
```

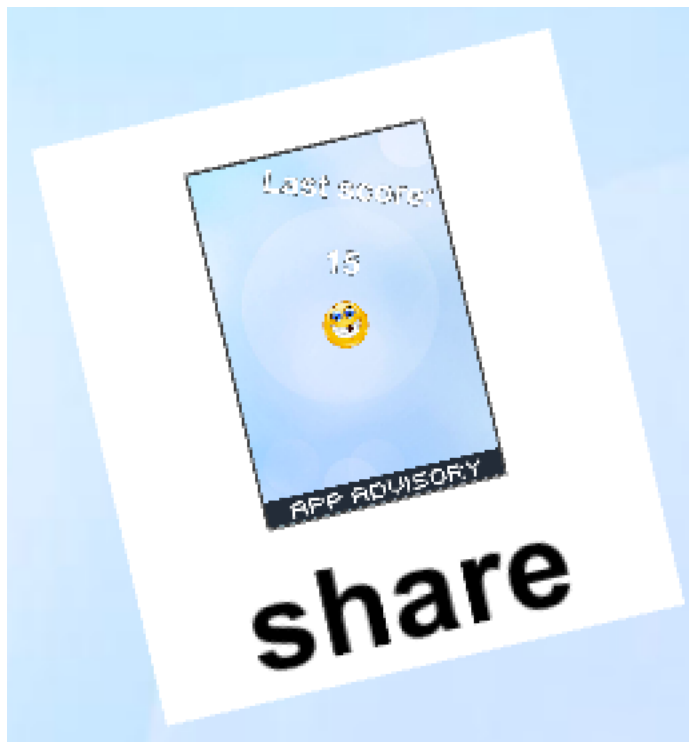
5) Show the screenshot :

You can show the screenshot only after you take one.

To show the screenshot in the VSSHARE UI Game Object, call this method :

```
VSSHARE.DOOpenScreenshotButton();
```

The VSSHARE button will appear on the screen.



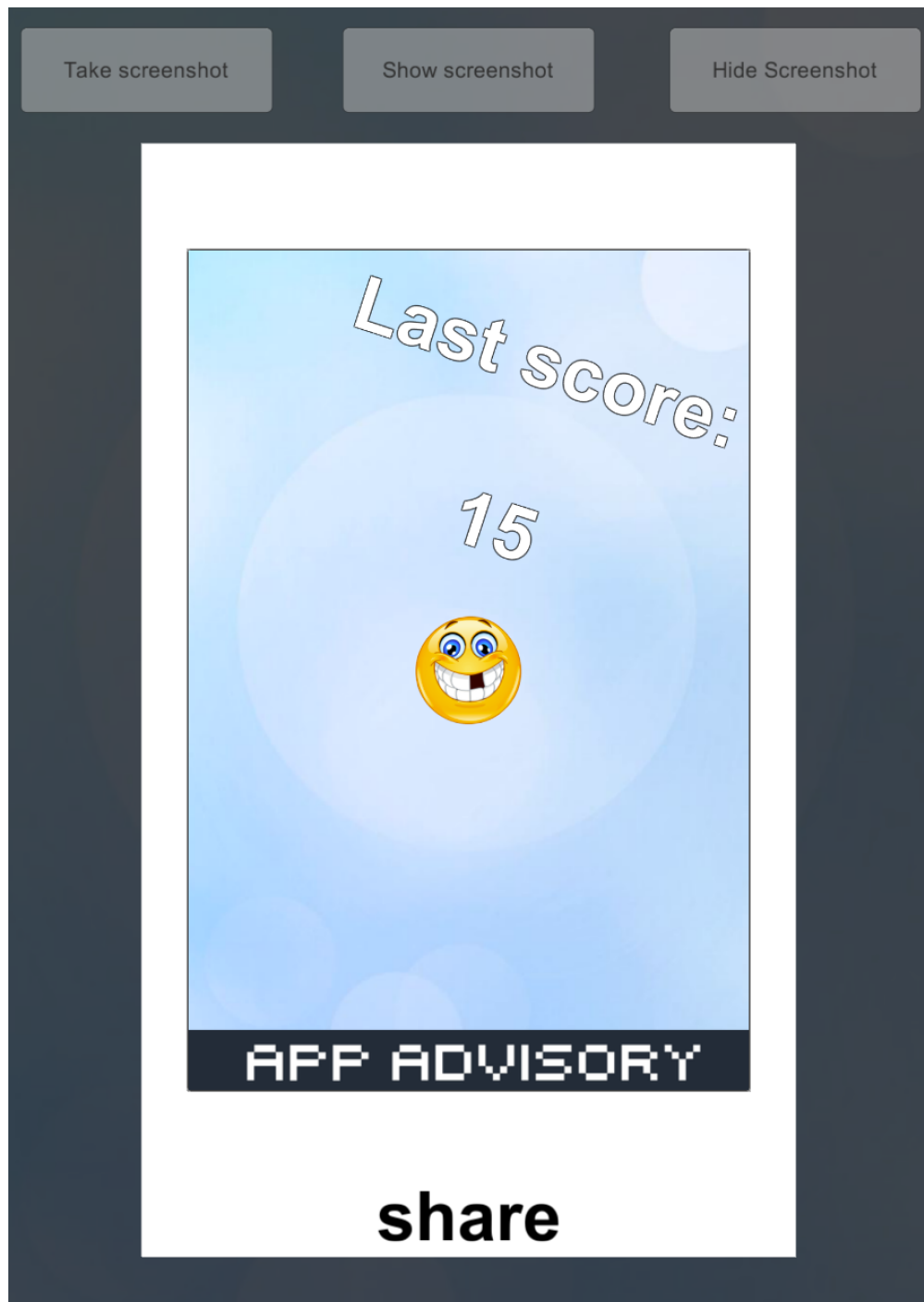
(image from the demo scene)

You can close (= hide) the screenshot button (at this state only) by calling :

```
VSSHARE.DOHideScreenshotIcon();
```

6) Open the screenshot sharing window:

Just click/touch the VSSHARE button (view in 5) section).



(image from the demo scene)

You can do it from code with this method :

```
VSSHARE.D00nclickedOnIconScreenshot();
```

7) Share or close the screenshot sharing window:

Touch the area to share the screenshot



Touch this area to close the sharing window

(image from the demo scene)

To close this windows from code :

```
VSSHARE.DOCloseShareWindow();
```

To share the screenshot from code :

```
VSSHARE.D00nclickedOnIconScreenshot();
```

DELEGATES

You can subscribe to some delegates to have more controls.

```
#region delegate
[System.Serializable] public delegate void OnButtonShareIsClosedHandler();
[SerializeField] public static event OnButtonShareIsClosedHandler OnButtonShareIsClosed;

[System.Serializable] public delegate void OnButtonShareIsIconHandler();
[SerializeField] public static event OnButtonShareIsIconHandler OnButtonShareIsIcon;

[System.Serializable] public delegate void OnButtonShareIsShareWindowHandler();
[SerializeField] public static event OnButtonShareIsShareWindowHandler OnButtonShareIsShareWindow;

[System.Serializable] public delegate void OnScreenshotTakenHandler(Texture2D tex);
[SerializeField] public static event OnScreenshotTakenHandler OnScreenshotTaken;
#endregion
```

Delegate who is called when a screenshot is taken :

```
[System.Serializable] public delegate void OnScreenshotTakenHandler(Texture2D tex);
[SerializeField] public static event OnScreenshotTakenHandler OnScreenshotTaken;
```

Exemple of use :

[BtnTakeScreenshot.cs](#) in the exemple scene.

Make the subscription :

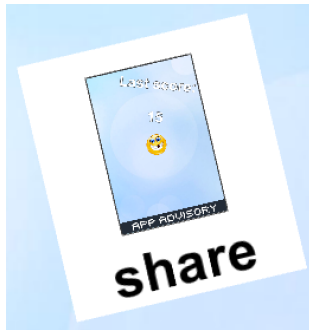
```
VSSHARE.OnScreenshotTaken += OnScreenshotTakenDelegate;
VSSHARE.DOTakeScreenShot();
```

Receive the event :

```
void OnScreenshotTakenDelegate(Texture2D tex)
{
    VSSHARE.OnScreenshotTaken -= OnScreenshotTakenDelegate;
    Debug.Log("UnityEventListener - Screenshot taken!!");
}
```

The delegate return the Texture2D (= the screenshot).

Delegate who is called when the VSSHARE is in the icon state :



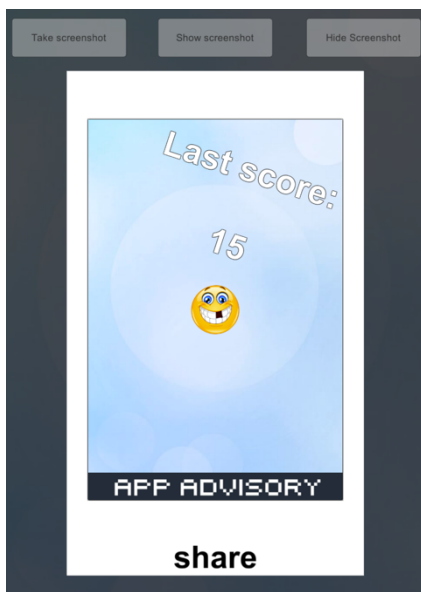
Make the subscription :

```
VSSHARE.OnButtonShareIsIcon += OnButtonShareIsIcon;
```

Receive the event :

```
void OnButtonShareIsIcon()  
{  
    Debug.Log("UnityEventListener - the screenshot button is in the icon state!!");  
}
```

Delegate who is called when the VSSHARE is in the full screen sharing window state :



```
[System.Serializable] public delegate void OnButtonShareIsShareWindowHandler();  
[SerializeField] public static event OnButtonShareIsShareWindowHandler OnButtonShareIsShareWindow;
```


Make the subscription :

```
VSSHARE.OnButtonShareIsShareWindow += OnButtonShareIsShareWindow;
```

Receive the event :

```
void OnButtonShareIsShareWindow()  
{  
    Debug.Log("UnityEventListener - the screenshot button is in the full screen window state!!");  
}
```

Delegate who is called when the VSSHARE is closed:

```
[System.Serializable] public delegate void OnButtonShareIsClosedHandler();  
[SerializeField] public static event OnButtonShareIsClosedHandler OnButtonShareIsClosed;
```

Make the subscription :

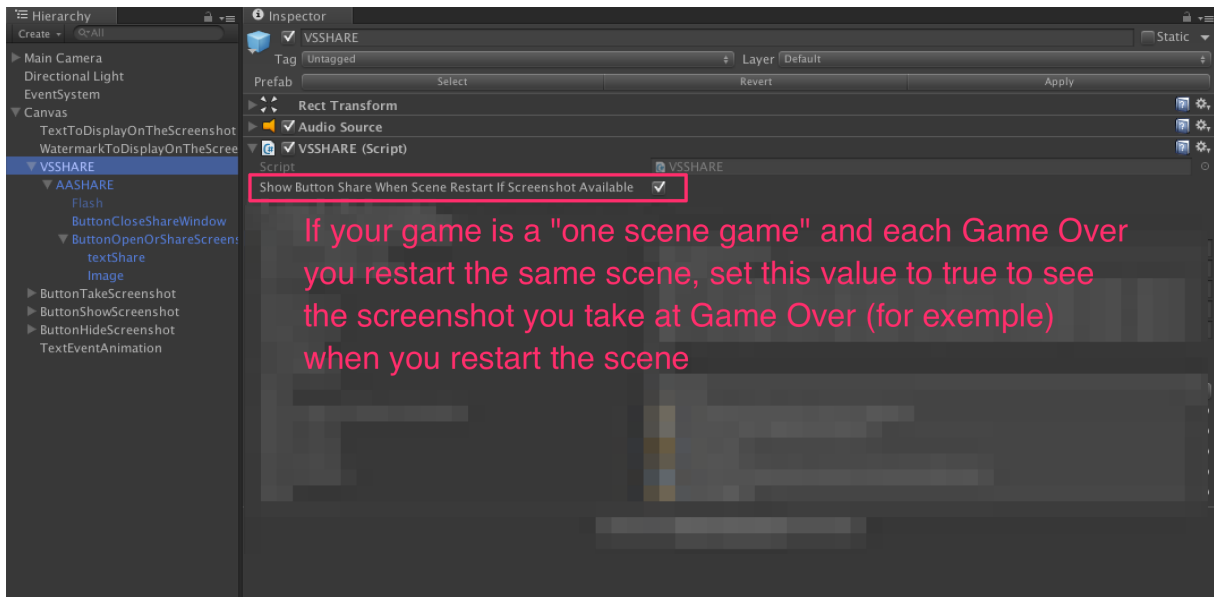
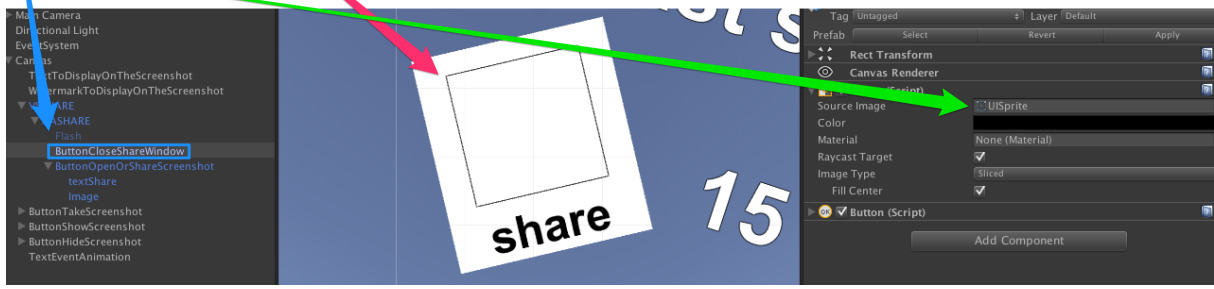
```
VSSHARE.OnButtonShareIsShareWindow += OnButtonShareIsClosed;
```

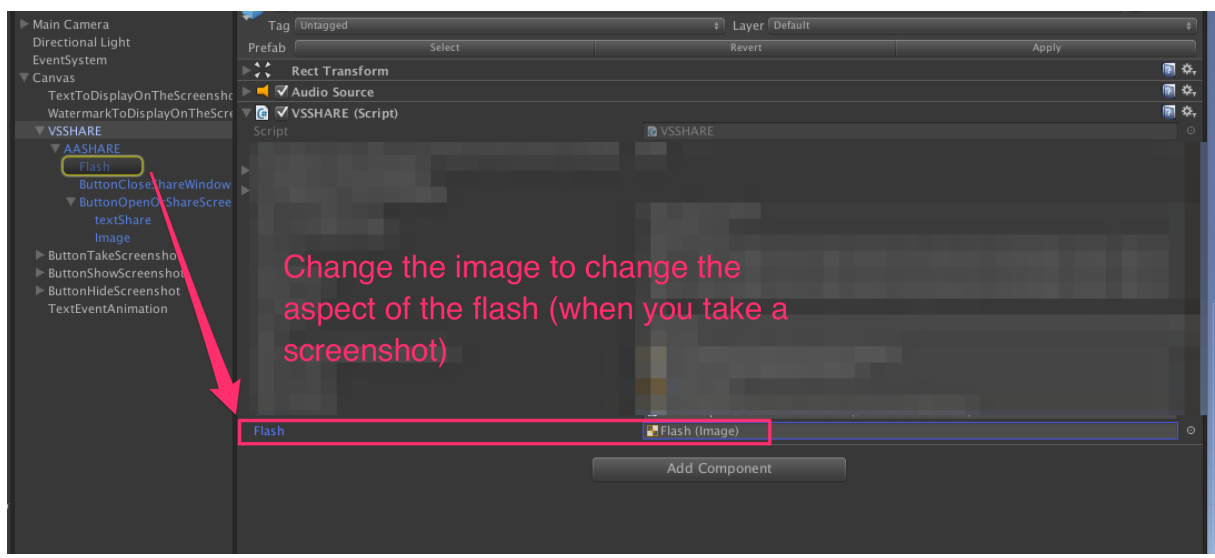
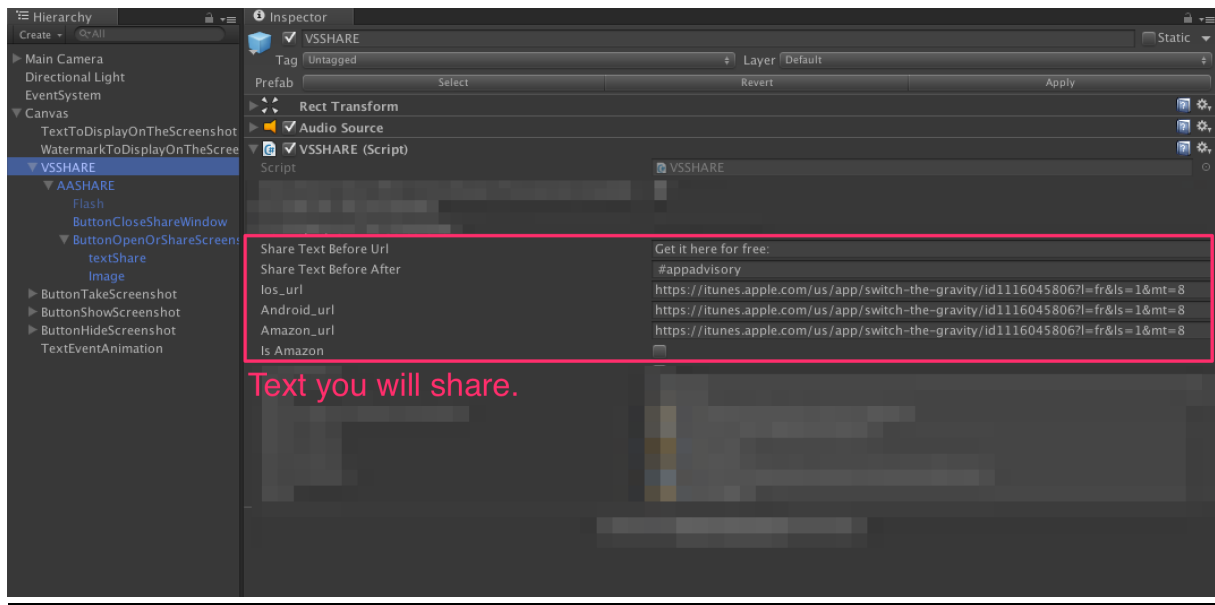
Receive the event :

```
void OnButtonShareIsClosed()  
{  
    Debug.Log("UnityEventListener - the screenshot button is closed!!");  
}
```

Customization

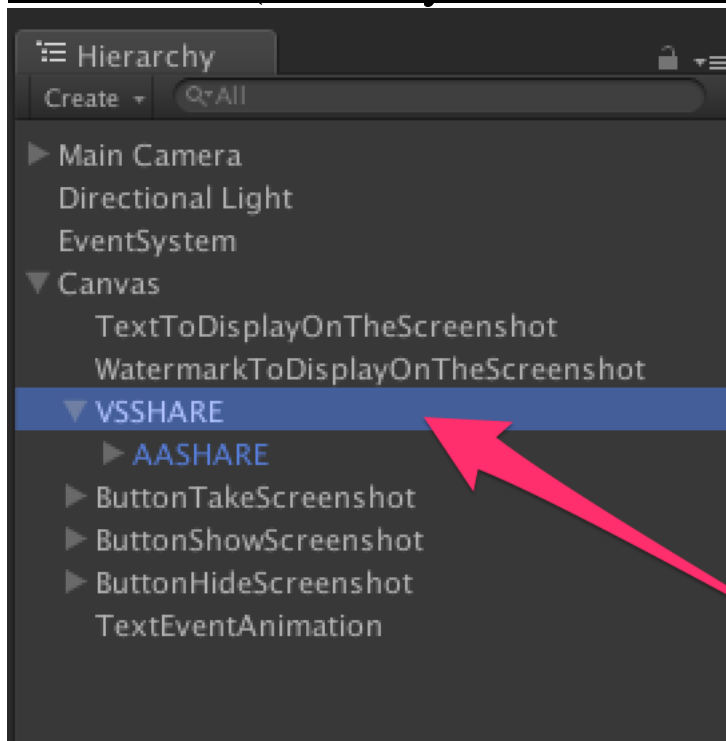
To change this
Select the button
And add your
new image



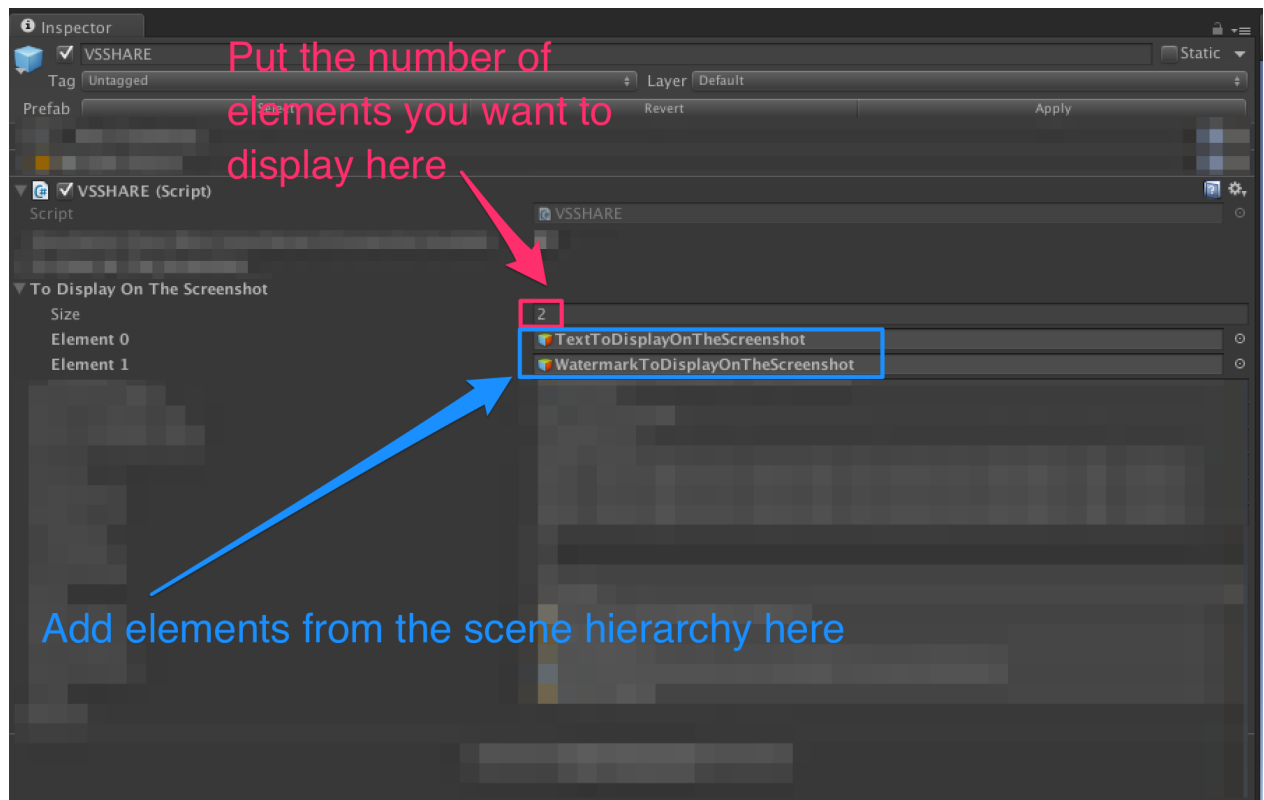


TIPS :

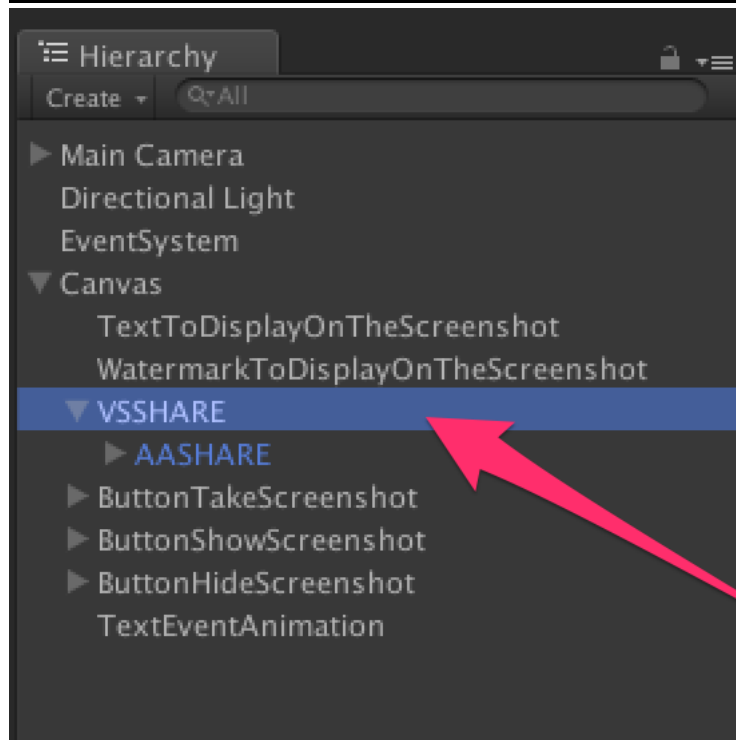
- Add text, image (like watermark) etc to the screenshot (and only visible on the screenshot !) :



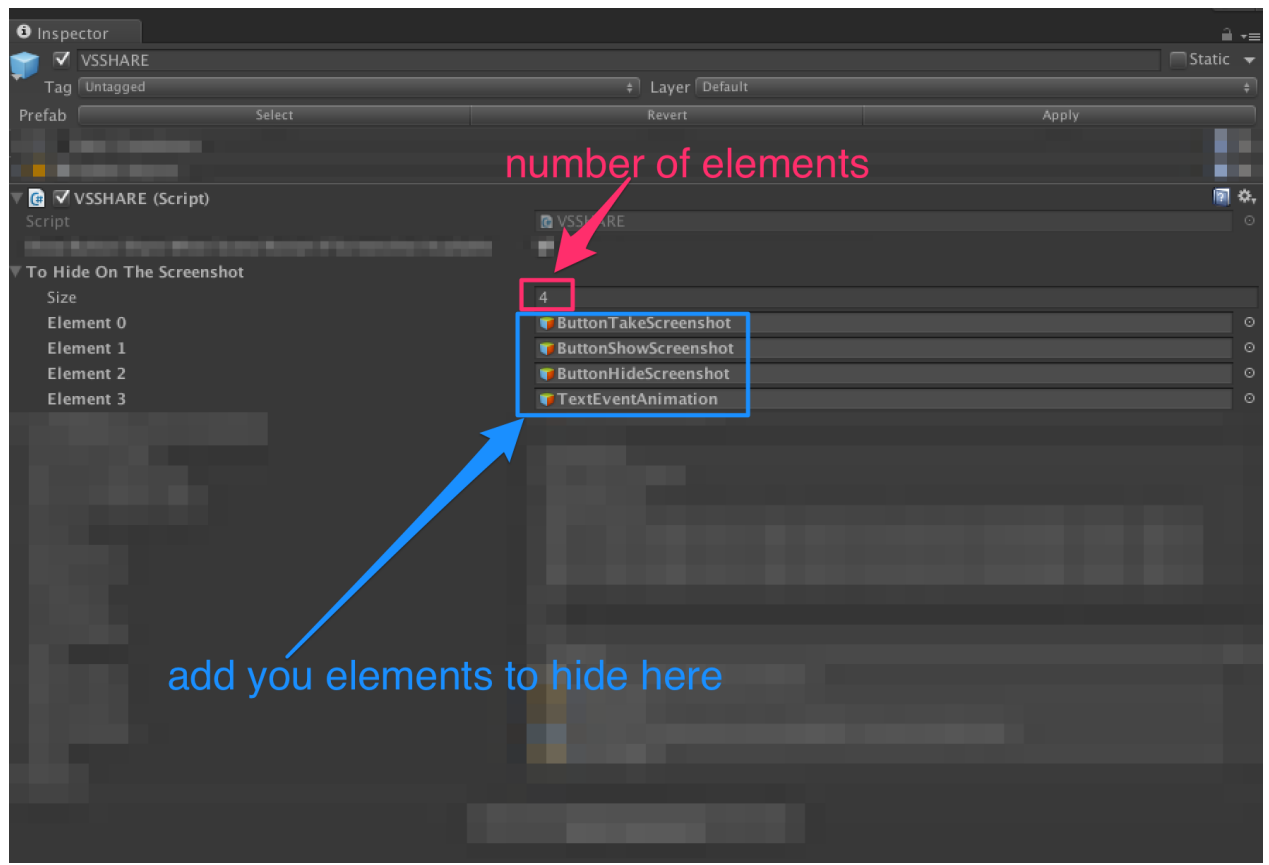
Select the
VSSHARE



- Hide some scene elements in the screenshot :



Select the
VSSHARE



Thanks !

Our other assets : <http://u3d.as/9cs>

Inquiries : <https://appadvisory.zendesk.com/>