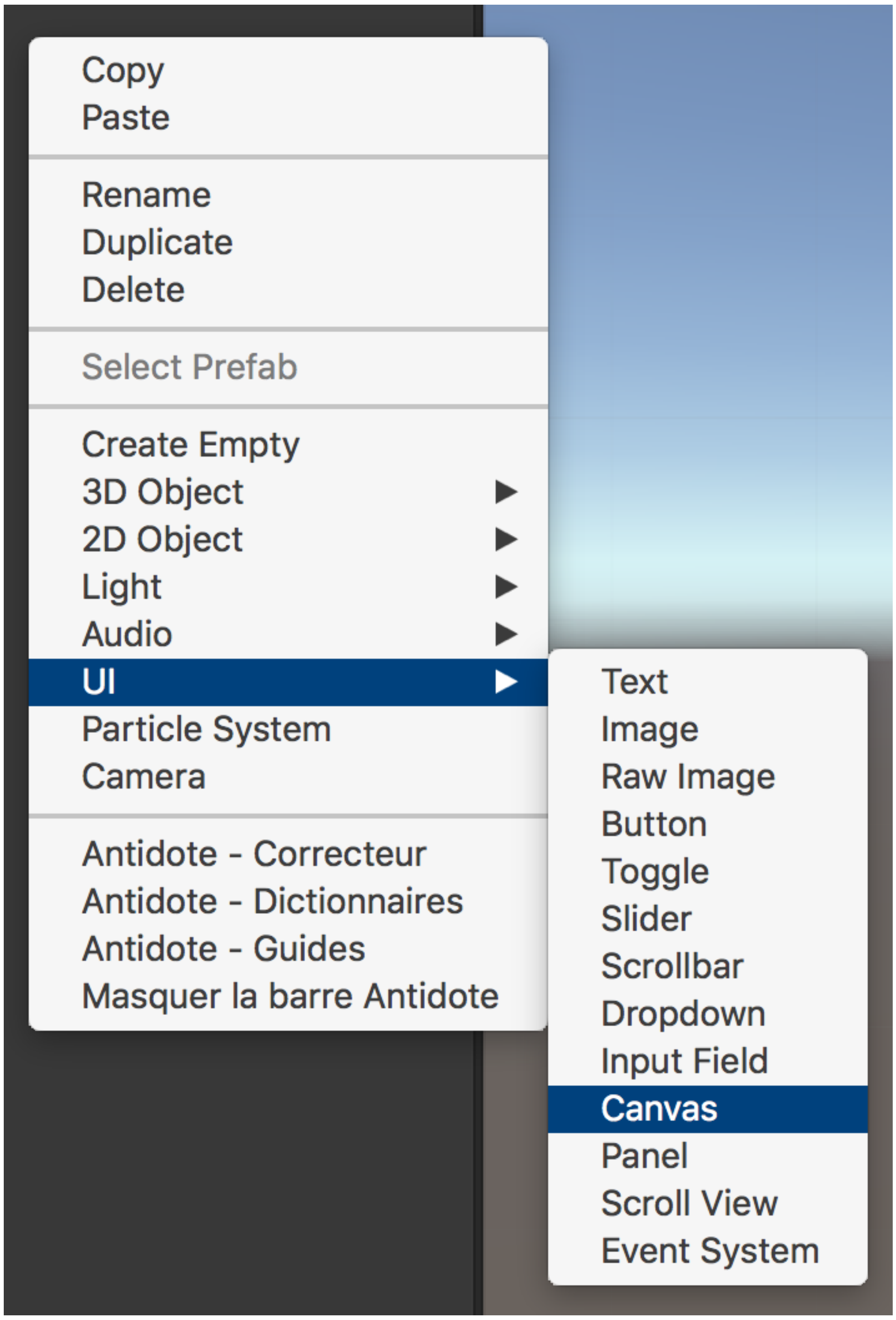
APP ADVISORY

Very Simple Share

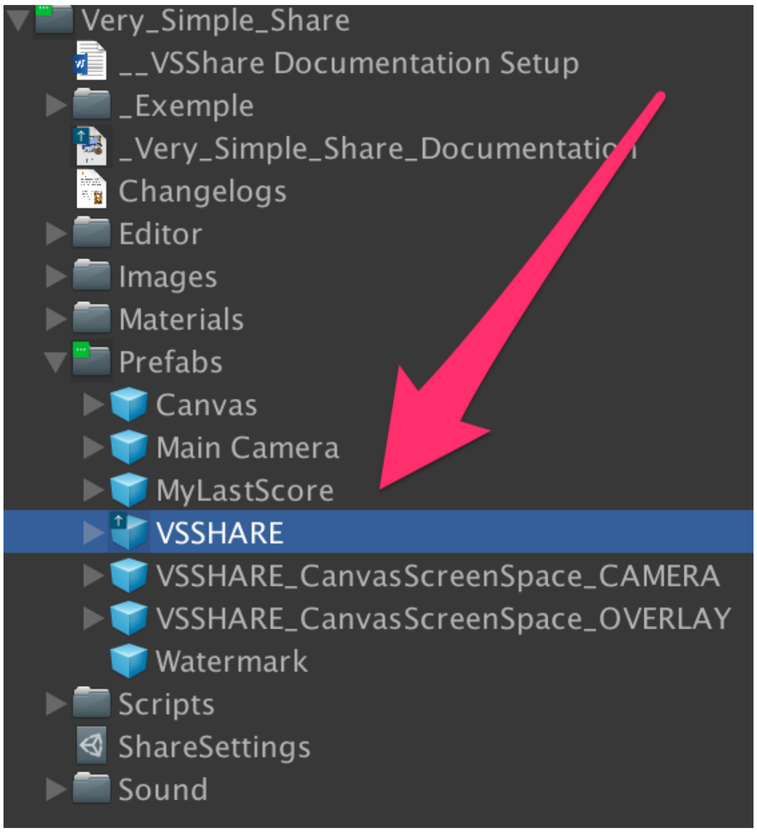
QUICK START :

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

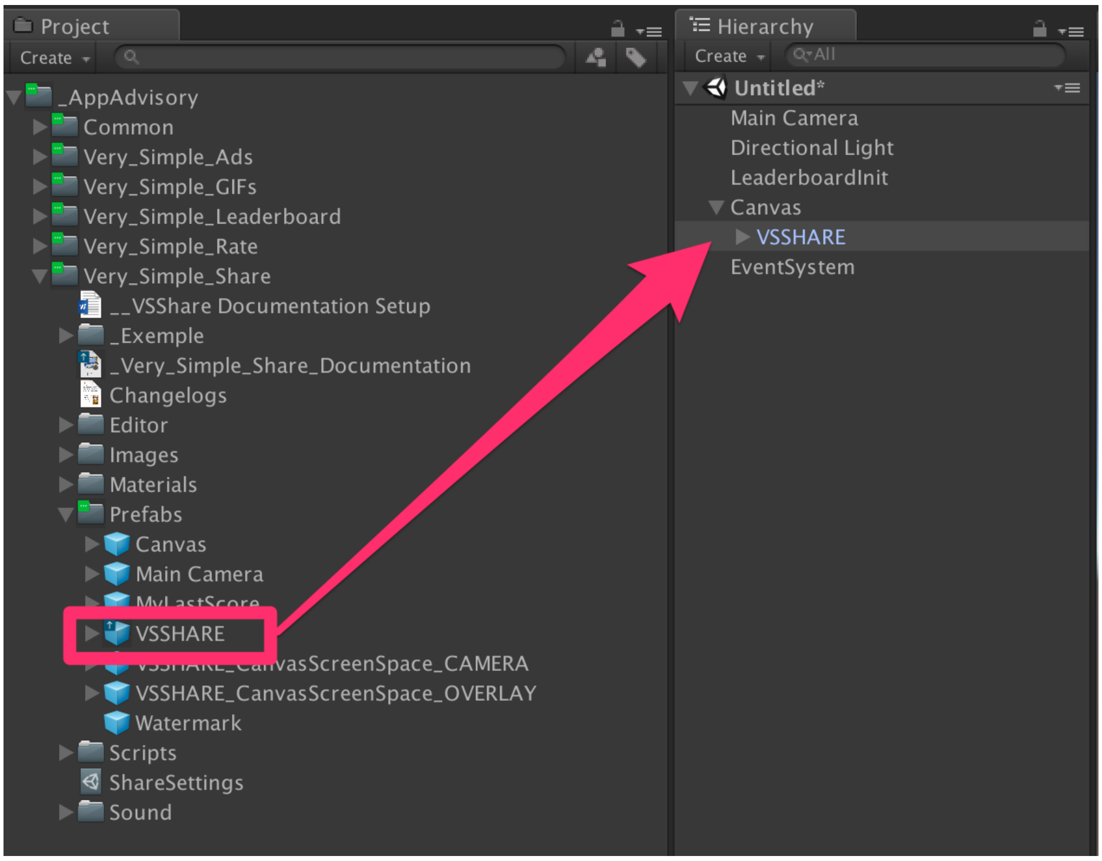
To create a new one :



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();
```

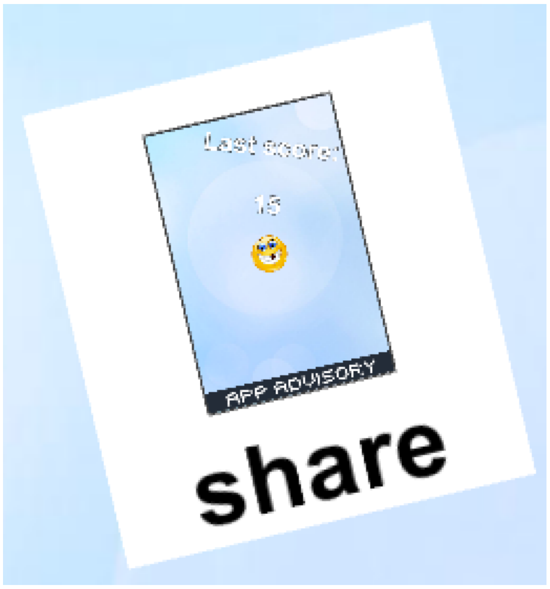
5. Show the screenshot :

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.



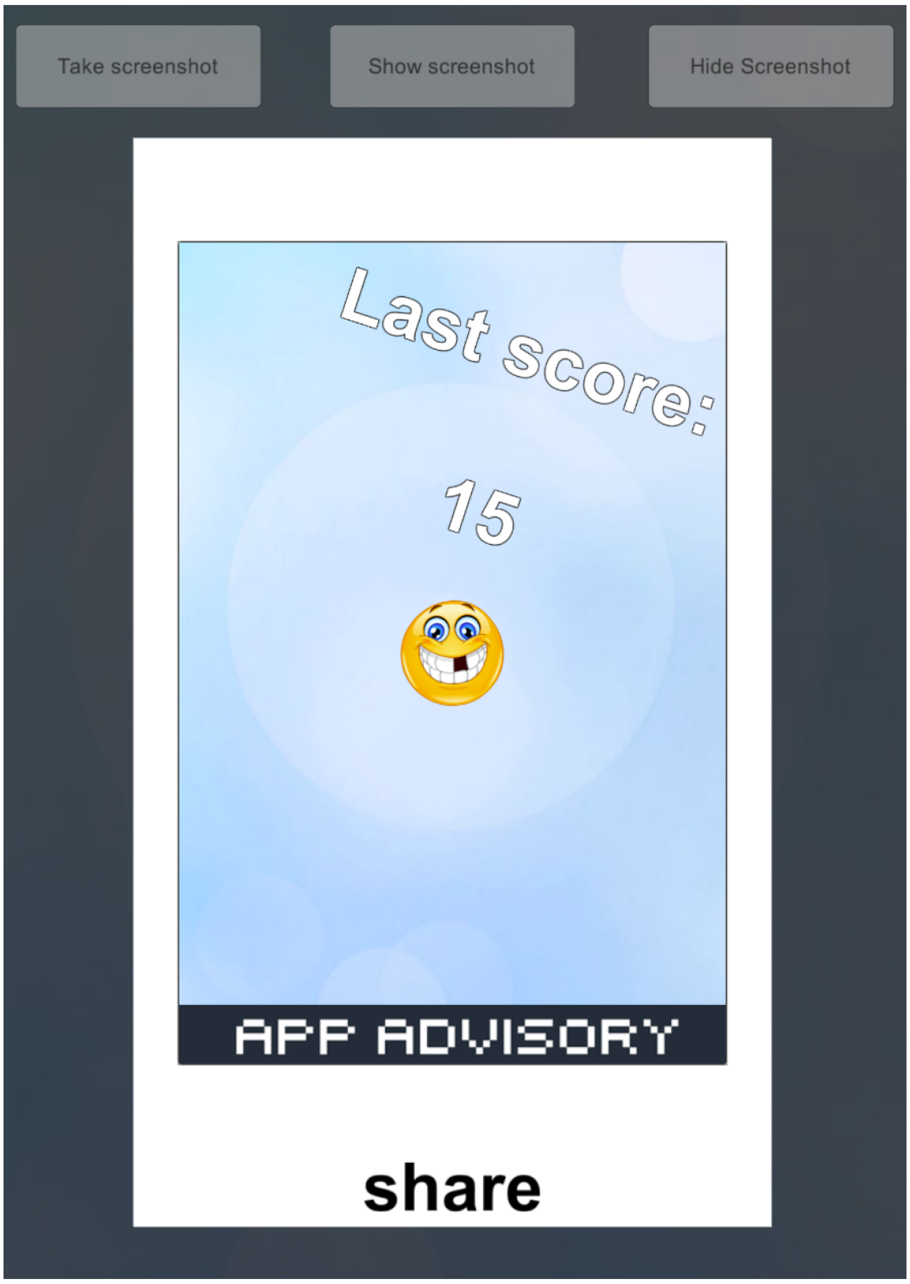
(image from the demo scene)

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).

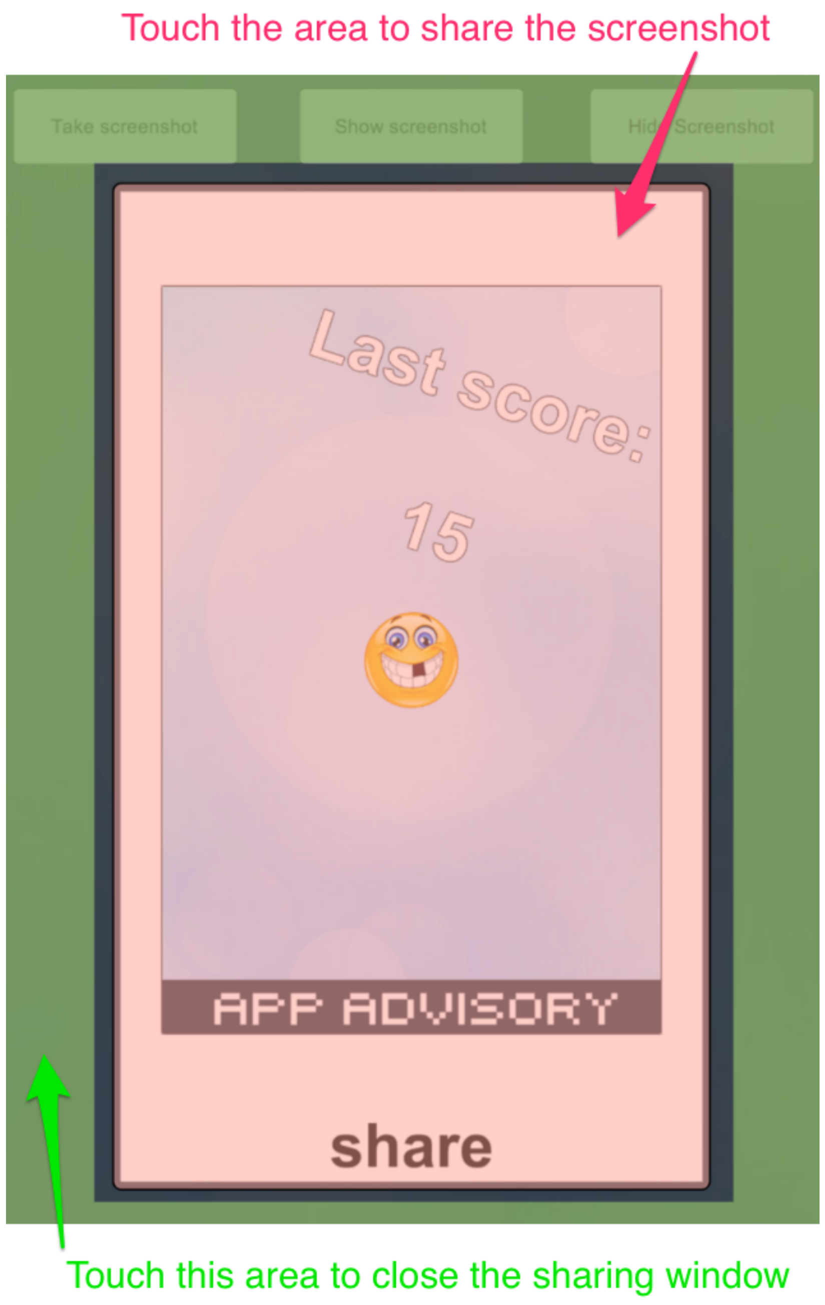


(image from the demo scene)

You can do it from code with this method :

```
VSSHARE.DOOnclickedOnIconScreenshot();
```

7. Share or close the screenshot sharing window:


Touch the area to share the screenshot
Take screenshot
Show screenshot
Hide Screenshot
Last score:
15
APP ADVISORY
share
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.DOOnclickedOnIconScreenshot();
```

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 which is called when a screenshot is taken :

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

Example of use :
BtnTakeScreenshot.cs in the example 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 which is called when the VSSHARE if in the icon state :

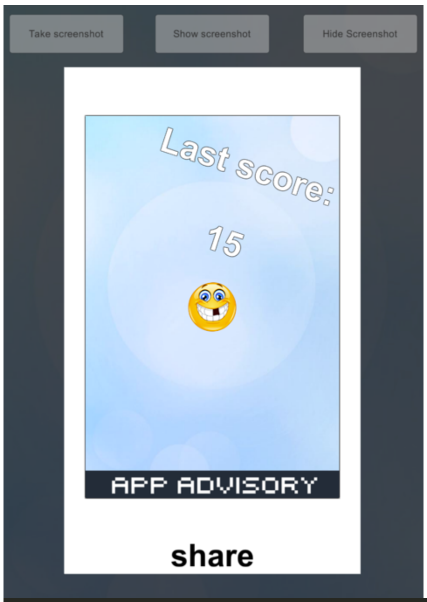
Make the subscription :

```
VSSHARE.OnButtonShareisIcon += OnButtonShareisIcon;
```

Receive the event :

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

Delegate which is called when the VSSHARE if 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;
```

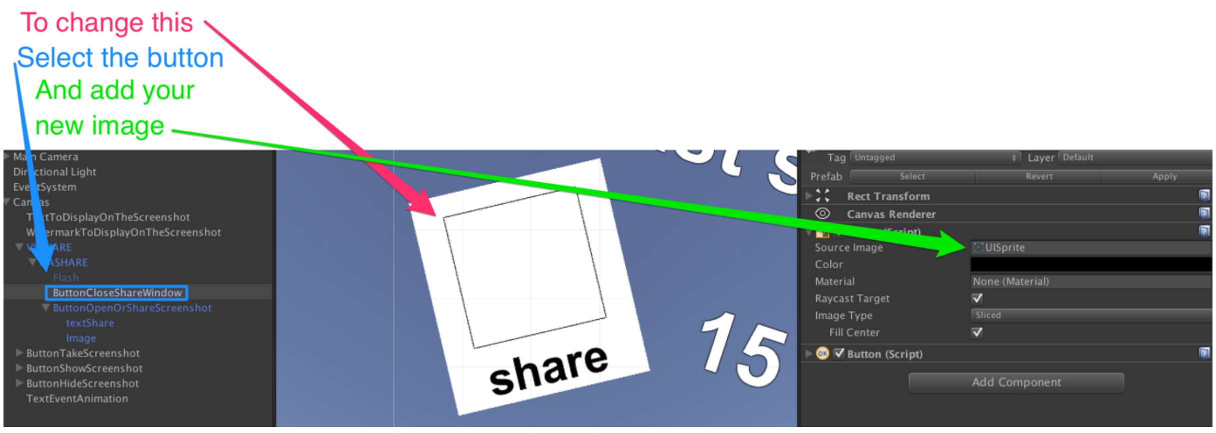
Make the subscription :

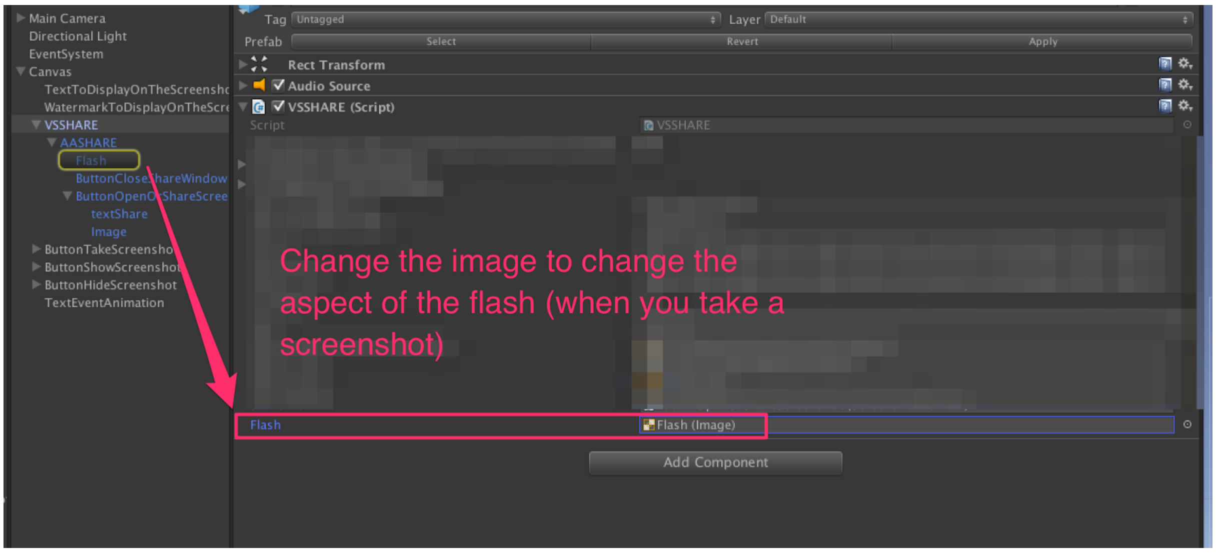
```
VSSHARE.OnButtonShareIsShareWindow += OnButtonShareIsClosed;
```

Receive the event :

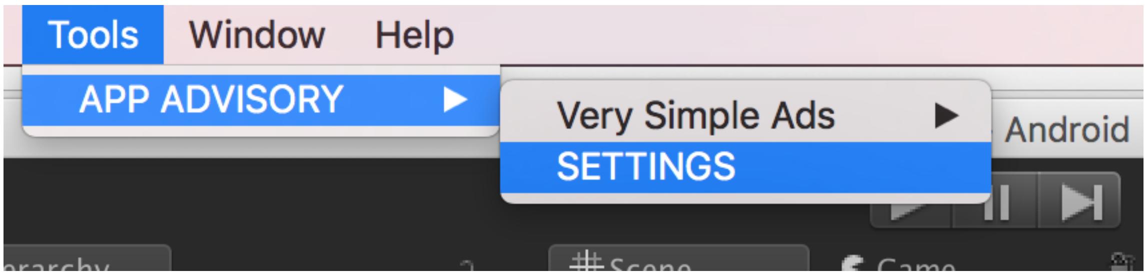
```
void OnButtonShareIsClosed()
{
    Debug.Log("UnityEventListener - the screenshot button is closed!!");
}
```

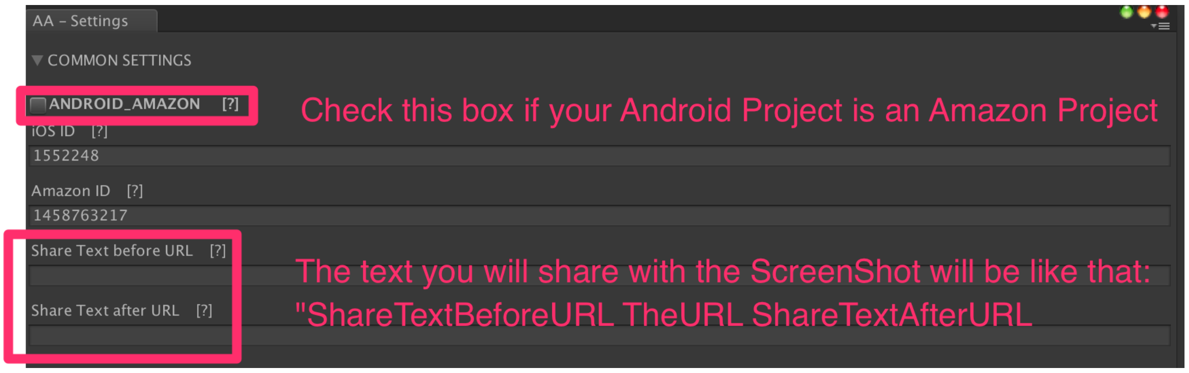
Customization


To change this
Select the button
And add your
new image


Change the image to change the aspect of the flash (when you take a screenshot)

Open the Settings Window




AA - Settings

COMMON SETTINGS

☐ ANDROID_AMAZON [?]

Check this box if your Android Project is an Amazon Project

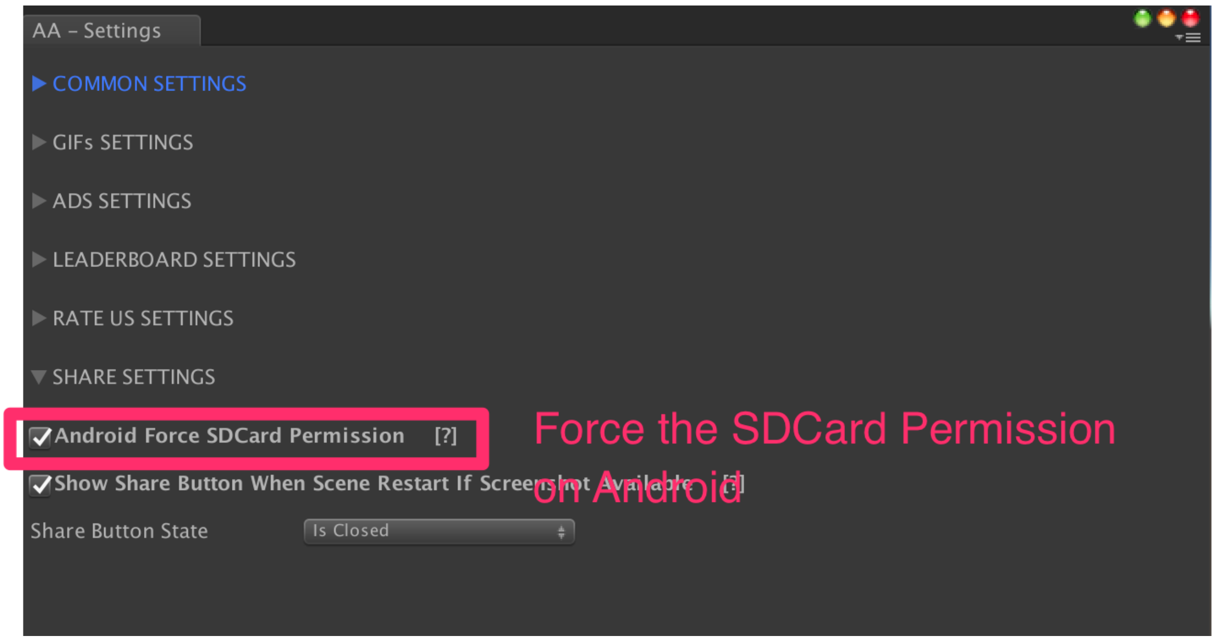
iOS ID [?]
1552248

Amazon ID [?]
1458763217

Share Text before URL [?]

Share Text after URL [?]

The text you will share with the ScreenShot will be like that: "ShareTextBeforeURL TheURL ShareTextAfterURL


AA - Settings

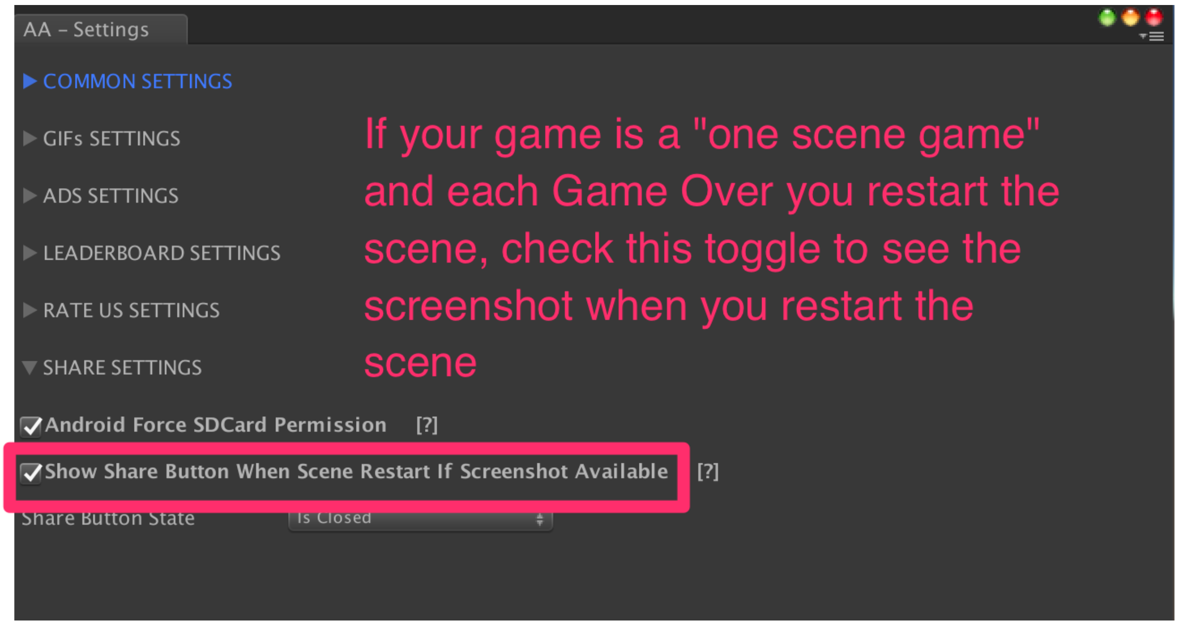
COMMON SETTINGS
GIFs SETTINGS
ADS SETTINGS
LEADERBOARD SETTINGS
RATE US SETTINGS
SHARE SETTINGS

✓ Android Force SDCard Permission [?]

Force the SDCard Permission on Android

✓ Show Share Button When Scene Restart If Screenshot Available [?]

Share Button State: Is Closed


AA - Settings

COMMON SETTINGS
GIFs SETTINGS
ADS SETTINGS
LEADERBOARD SETTINGS
RATE US SETTINGS
SHARE SETTINGS

If your game is a "one scene game" and each Game Over you restart the scene, check this toggle to see the screenshot when you restart the scene

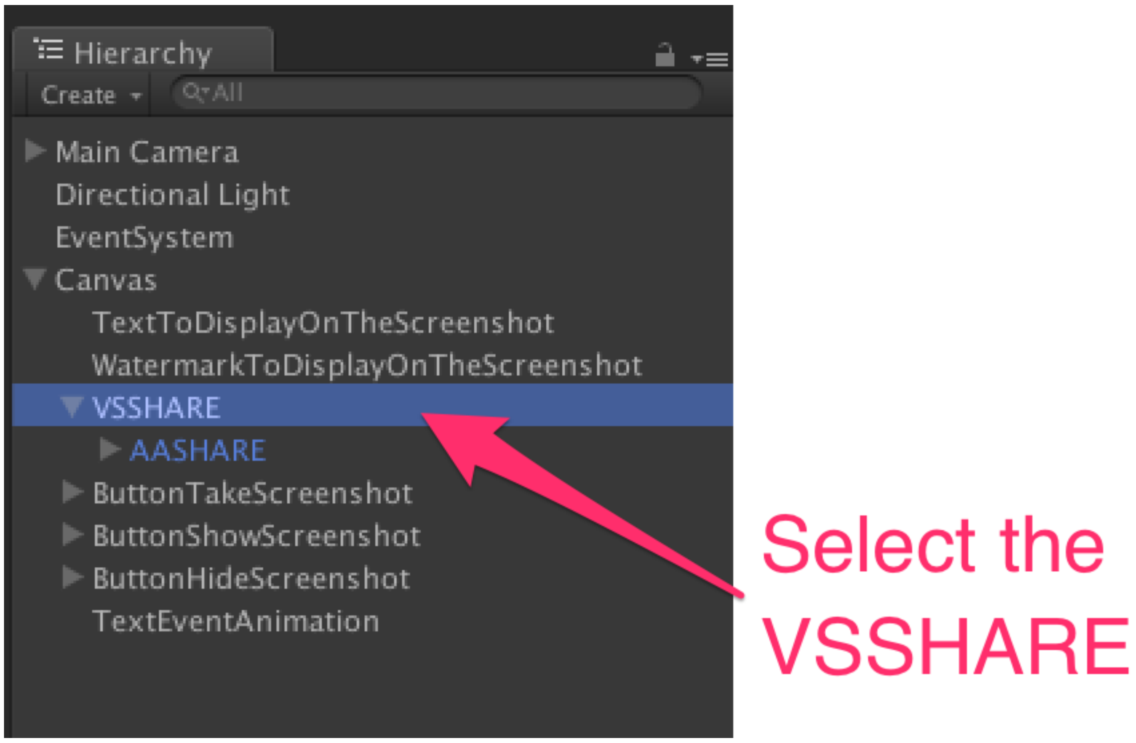
✓ Android Force SDCard Permission [?]

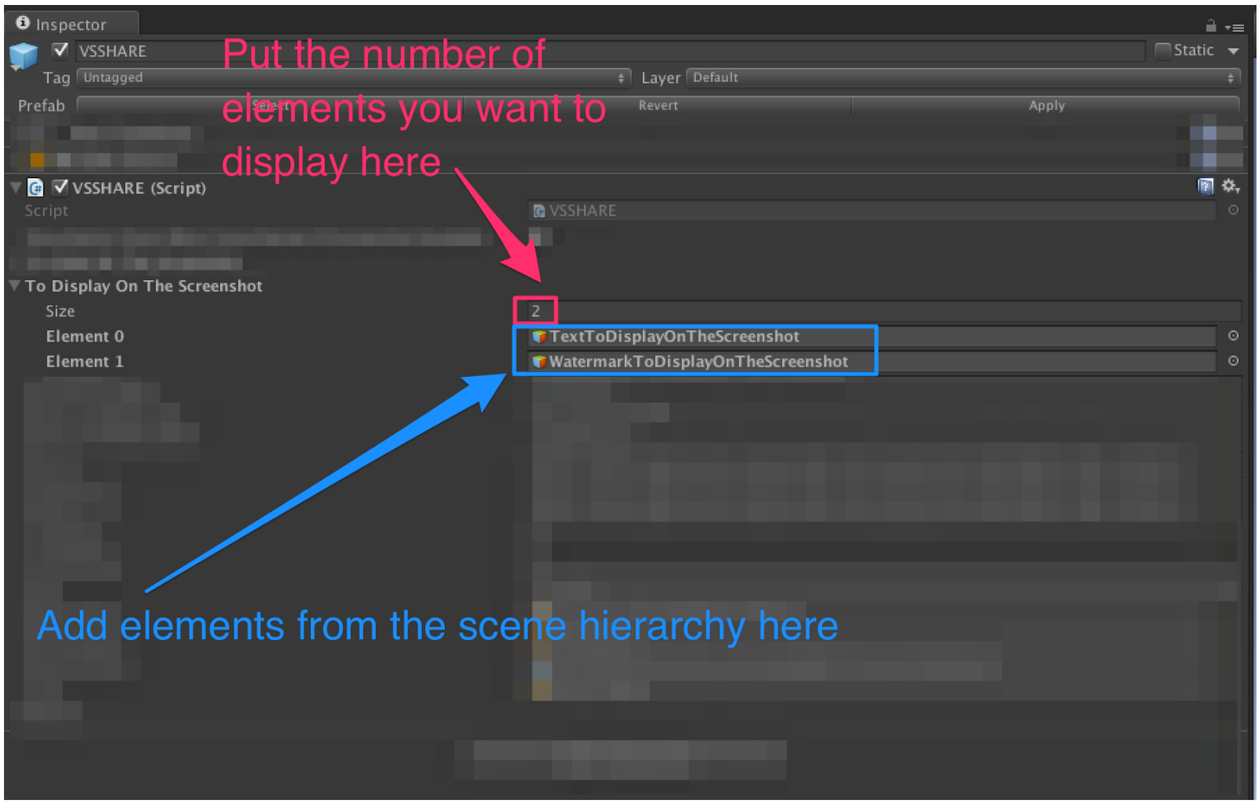
✓ Show Share Button When Scene Restart If Screenshot Available [?]

Share Button State: Is Closed

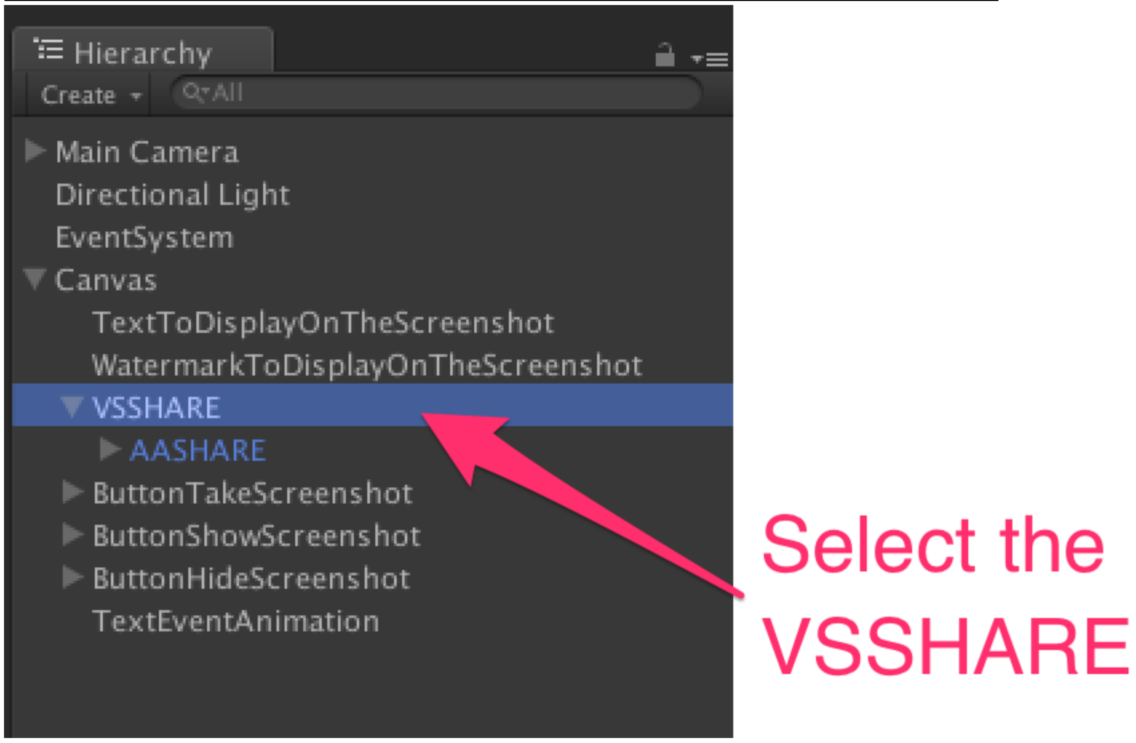
TIPS :

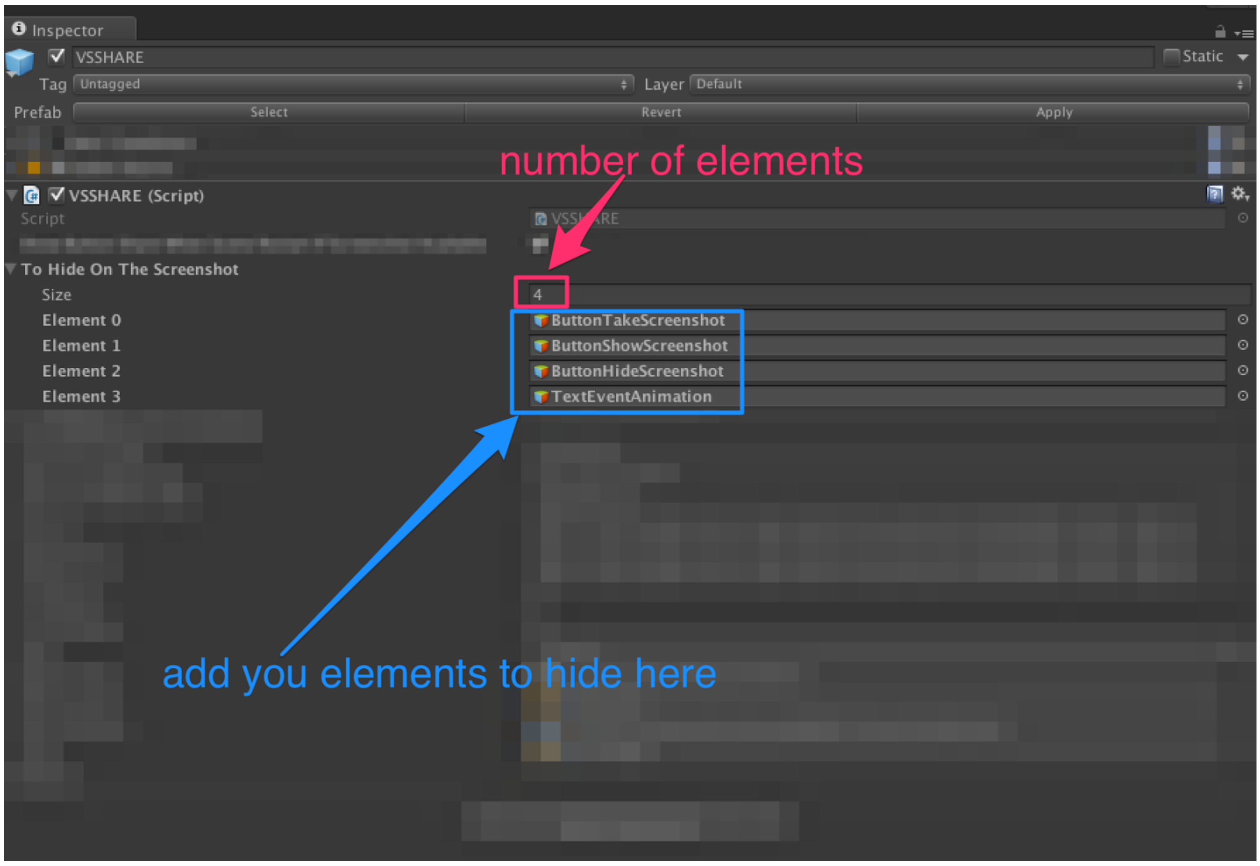
- Add text, image (like watermark) etc to the screenshot (and only visible on the screenshot !) :




Inspector
VSSHARE
Static
Tag Untagged
Layer Default
Prefab
Select
Revert
Apply
VSSHARE (Script)
Script
VSSHARE
To Display On The Screenshot
Size
2
Element 0
TextToDisplayOnTheScreenshot
Element 1
WatermarkToDisplayOnTheScreenshot
Put the number of elements you want to display here
Add elements from the scene hierarchy here

- <u>Hide some scene elements in the screenshot :</u>





VSSHARE
Static
Tag Untagged
Layer Default
Prefab Select Revert Apply
VSSHARE (Script)
Script VSSHARE
To Hide On The Screenshot
Size 4
Element 0 ButtonTakeScreenshot
Element 1 ButtonShowScreenshot
Element 2 ButtonHideScreenshot
Element 3 TextEventAnimation

Thanks!

Please rate my file, I'd appreciate it! http://u3d.as/u3N

Produced by App Advisory - http://app-advisory.com
Facebook: https://facebook.com/appadvisory
Contact us: https://appadvisory.zendesk.com/hc/en-us/requests/new
App Advisory Unity Asset Store catalog: http://u3d.as/9cs
Developed by Gilbert Anthony Barouch - https://www.linkedin.com/in/ganbarouch
Inquiries: https://appadvisory.zendesk.com/

Very Simple Ads:

Everything is done for you: « Very Simple Ad » is already implemented.
Get it here: http://u3d.as/oWD

Very Simple Rate:

Everything is done for you: « Very Simple Rate » is already implemented.
Get it here: http://u3d.as/Dt2

Very Simple Leaderboard:

Everything is done for you: « Very Simple Leaderboard » is already implemented.
Get it here: http://u3d.as/qxf

Very Simple GIF:

Everything is done for you: « Very Simple GIF » is already implemented.
Get it here: http://u3d.as/ACQ