

Readme - Screenshot Helper 1.3.0

Easy capture screenshot image, capture any part of the screen or capture image from selected Unity camera.

With highly customizable capture settings, provides with flexible, easy to use APIs and code-less editor screenshot tool for capturing quality images in your app or in the Unity Editor.

Features (Free)

- Capture full-screen image.
- Capture part of the screen(Custom rect region at any screen position).
- Capture image using Unity camera, can scale the image size(0.1X – 4X).
- Support Unity built-in render pipeline and all Scriptable Render Pipelines
- Support Anti-Aliasing for camera capture methods.
- Support touch, including touch-to-capture example.
- Code-less OnEditorScreenshot feature(For the Editor Play Mode).
- Capture and returns Texture2D, Sprite, or RenderTexture.
- Handy image display handler script for UGUI.
- Compatible with Pro GIF for saving texture(s) as GIF.
- All platforms.

Features + (Plus)

The below features are included in **Screenshot Helper Plus** only:

Mobile Media Plugin

- Save audio files to the Music folder on Android.
- Pick audio files from native folders on Android.
- Save image, video, GIF to native gallery(Android & iOS).
- Pick image, video, GIF from native gallery(Android & iOS).
- Check native storage permission and request permission.
- Prompt native setting menu for changing permission.
- Save files to any specific folder in the Gallery.
- Supports save files with specific filename on Android and iOS.
- Supports popup view or fullscreen view for selecting file on iOS.
- Supports getting thumbnail and full-size image for Photo, Video, and GIF, on iOS.
- Detailed example scene included, ready-to-build for testing on devices.

*Demo scene included for showing how to pick/save image or video from/to mobile device gallery. Please find the Readme document and demo(**MobileMediaTest.unity**) in the MobileMedia folder.*

Watermark Util

- Add watermark texture on screenshot
- Add watermark texture on particular texture
- Support transparency
- Set watermark position
- No need to place the watermark logo on UI
- Flexible, easy, texture combine API, can be used independently

(1) Reminders, Setup & Requirement

Setup for MobileMedia Plugin (Plus Only)

Requires Android 4.4(API Level 19) or later for Android platform,

Requires iOS 8.0 or later for iOS platform.

For Android, set **Write Permission** to “External (SDCard)” in Unity “Player Settings”.

Setup for Scriptable Render Pipeline (SRP) : URP/LWRP/HDRP.

If you are using Scriptable Render Pipeline (e.g. URP/LWRP/HDRP) in your project, please ensure to use the **OnUpdateRender** mode for the camera capture methods. Just call the below API once before capturing images with camera:

[`ScreenshotHelper.SetRenderMode\(ScreenshotHelper.RenderMode.OnUpdateRender\);`](#)

Watermark Util

Please set the Read/Write Enable flag as 'true' in the texture Import Settings for the imported textures and watermark icon.

(2) The MainOnCaptured callback

All capture methods in the ScreenshotHelper script will fire the **MainOnCaptured** callback on capture complete. It is like a central place for receiving captured images. So, add these code in your script for receiving all captured images if need:

** It is optional to set the MainOnCaptured callback, because every capture method has a callback for returning the captured image. We will talk about that later.

- **Set the callback to receive captured image in Texture2D format:**
`ScreenshotHelper.iSetMainOnCapturedCallback((Texture2D texture2d)=>{
 //Your code for handling texture2d:
});`
or
- **Set the callback to receive captured image in Sprite format:**
`ScreenshotHelper.iSetMainOnCapturedCallback((Sprite sprite)=>{
 //Your code for handling sprite:
});`
or
- **Set the callback to receive captured image in RenderTexture format:**
`ScreenshotHelper.iSetMainOnCapturedCallback((RenderTexture renderTexture)=>{
 //Your code for handling renderTexture:
});`

(3) Capture Methods

All screenshot capture methods has a callback for receiving the newly captured image. Capture a screenshot image can be done with just one line of code!

The callback parameter **onCapturedCallback** that used in the below examples can be a method name or Action, for receiving & handling the captured image:

- **Capture fullscreen image:**
`ScreenshotHelper.iCaptureScreen(onCapturedCallback);`

- **Capture a region of the screen:**

Specify a screen position and size(Vector2) to capture an particular area of the screen.

`ScreenshotHelper.iCapture(Input.mousePosition, captureSize, onCapturedCallback);`

- **Capture using the view of the specific camera:**

`ScreenshotHelper.iCaptureWithCamera(camera, onCapturedCallback);`

or

`ScreenshotHelper.iCaptureRenderTextureWithCamera(camera, onCapturedCallback);`

Anti-Aliasing

The anti-aliasing level can be set for better quality for camera capture methods when **OnUpdateRender** mode is used(even the Anti-Aliasing option is disabled in the Unity QualitySettings):

`ScreenshotHelper.AntiAliasingLevel = 4;`

Valid value: 1, 2, 4, 8 (1 = disable, 8 = best quality)

*To enable the **OnUpdateRender** mode, please use the below API:*

`ScreenshotHelper.SetRenderMode(ScreenshotHelper.RenderMode.OnUpdateRender);`

(4) Get the current texture/sprite

```
Texture2D texture2D = ScreenshotHelper.CurrentTexture;  
Sprite sprite = ScreenshotHelper.CurrentSprite;  
RenderTexture renderTexture = ScreenshotHelper.CurrentRenderTexture;
```

The above parameters stored the image of the previous capture. They will return a null if you never take a screenshot before. It is better to check null before using them.

(5) Save Image

To save the captured image (within application folder):

```
FilePathName fpn = new FilePathName();  
  
fpn.SaveTextureAs(Texture2D:texture, SaveFormat:saveFormat);  
or  
fpn.SaveTextureAs(Texture2D:texture, AppPath:appPath, string:subFolderName,  
bool:isJPG);
```

Set the SaveFormat enum for encoding the captured texture to different image format:

JPG - `FilePathName.SaveFormat.JPG`

PNG - `FilePathName.SaveFormat.PNG`

GIF - `FilePathName.SaveFormat.GIF` (Require Pro GIF)

For saving image to the mobile device gallery please refer to the Readme document of MobileMedia Plugin (Plus Only).

(6) OnEditorScreenshot

This is a code-less screenshot feature designed for capturing images in the Editor Play Mode. Provides a convenient way for saving your app/game screenshot during the editor running. So you can capture images anytime you need.

1. Drag the prefab(**OnEditorScreenshot.prefab**) in the OnEditor folder to the running scene. Or, add the OnEditorScreenshot.cs script to a GameObject in the scene.
2. Set the capture settings in the inspector panel.
3. Start to capture images with the "Capture" buttons in the inspector panel.

(7) Clear

Call the Clear method to clean up memory if need, this method will:

- 1) clear the stored textures
- 2) clear the main capture callback
- 3) remove the camera render script(CameraOnRender.cs/CameraOnUpdateRender.cs) from all cameras
- 4) unload all assets that are not used

```
ScreenshotHelper.iClear(bool: clearCallback, bool: clearTextures);
```

(8) Watermark Util

Watermark Util is one of our image processing tools, can work independently for combining textures. Supports transparency and supports setting watermark position easily. Demo included, you can find the **WatermarkDemo** scene in the WatermarkUtil folder.

To add watermark on the captured screenshot or any existing texture:

Watermark method 1:

```
targetTexture = WatermarkUtil.DrawWatermark(targetTexture, watermarkTexture1,  
watermarkPosition, watermarkAlpha1, isDrawInRect);
```

Watermark method 2:

```
targetTexture = WatermarkUtil.DrawWatermark(targetTexture, watermarkTexture2,  
watermarkPositionX, watermarkPositionY, watermarkAlpha2, watermarkWidthOffset,  
watermarkHeightOffset);
```

*Also, remember to set the Read/Write Enable flag as 'true' in the texture Import Settings for imported textures.

THANK YOU

Thank you for using this package!

For any question and bug report please contact us at swan.ob2@gmail.com.

Remember to rate this asset on the Asset Store. Your review is always appreciated, and very important to the development of this asset!

[Review And Rating](#)

Visit our asset page to find out more!

<https://www.swanob2.com/assets>

SWAN DEV