Readme - Screenshot Helper 1.2.1

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

Features (Free)

- Captures full-screen image.
- Captures part of the screen. (Rect region at any screen position)
- Captures image using Unity camera(s), can scale the image size(0.1X – 4X).
- Support touch, including touch-to-capture example.
- Codeless OnEditorScreenshot feature (For the Editor Play Mode)
- Save screenshot/cutout texture as JPG, PNG.
- Compatible with Pro GIF for saving texture(s) as GIF.
- Mobile friendly.
- Runtime

Features + (Plus)

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

Mobile Media Plugin

- Save image, video, gif to Native Gallery(Android Gallery & iOS Album)
- Pick image, video, gif from Native Gallery(Android Gallery & iOS Album)
- Check native storage permission and request permission
- Prompt native setting menu for changing permission
- Support create sub-folder by specifying the folder name
- Support specifying the file name on save file to Android gallery
- Support popup dialog or fullscreen album view, for selecting media file on iOS
- Support retrieve media thumbnail(image, video, gif) in the album on iOS

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

Watermark Util

- Add watermark texture on screenshot
- Add watermark texture on particular texture
- Support transparency
- Set watermark position
- Flexible, easy, texture combine API, can be used independently

(1) 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:
          //……
  });
  ```

(2) 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. (ie. You can import this asset, and then insert one of the below code snippet, done!)

The callback parameter **onCapturedCallback** that used in the below examples**,** can be a C# method name or Action, for receiving 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 with camera (at the camera's viewpoint)**

  ```
  ScreenshotHelper.iCaptureWithCamera(camera, onCapturedCallback);
  ```

  or

  ```
  ScreenshotHelper.iCaptureRenderTextureWithCamera(camera, onCapturedCallback);
  ```

(3) 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. So, it is better to check null before using them. They are also be returned with the callback in the related capture methods.

(4) 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).

(5) OnEditorScreenshot

This is a codeless screenshot feature designed for capturing images in the Editor Play Mode. So give the convenient for saving your app/game screenshot at any development moment.

1. Drag the 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 as you want in the inspector.
3. Start to capture.

(6) Clear

Call the Clear method to clean up memory if need, this methed will clear the stored texture and remove the camera script(CameraOnRender.cs) from the camera. Call this method when the texture is no longer need.

```
ScreenshotHelper.iClear();
```

(7) 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 existing textures.

THANK YOU

Thank you for using this package!
Any problem 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