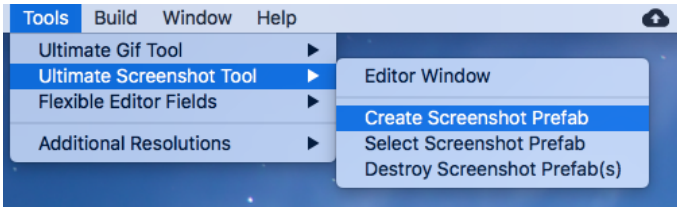
Ultimate Screenshot Tool

Tangled Reality Studios LLC

Table of Contents

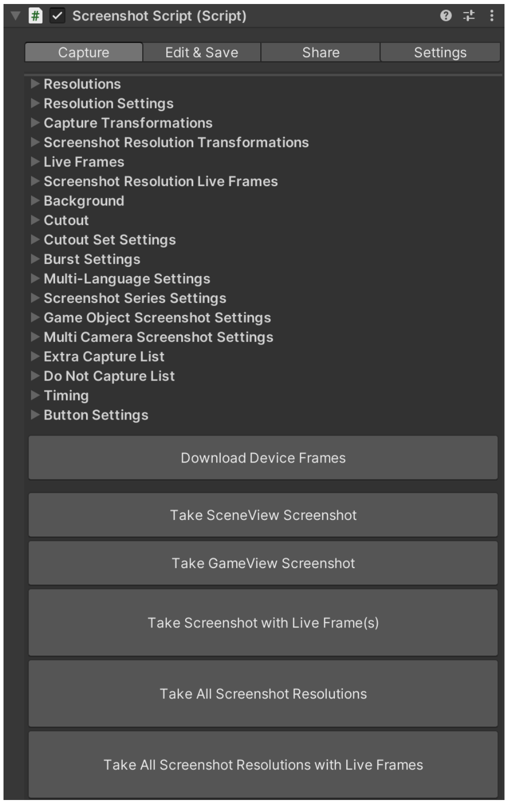
Getting Started

Create the Screenshot prefab from Tools -> Ultimate Screenshot Tool -> Create Screenshot Prefab



Default Buttons

- Take SceneView Screenshot - Takes a single screenshot of the current SceneView (may need to open scene view for button to appear)
- Take GameView Screenshot - Takes a single screenshot of the current GameView
- Take Screenshot with Live Frame(s) - Takes a GameView screenshot and within each of the frames listed in live frames
- Take All Screenshot Resolutions - Takes a screenshot for each resolution in the resolutions list
- Take All Screenshot Resolutions with Live Frames - Takes a screenshot for each resolution in the resolutions list and within each of the frames listed in live frames and screenshot resolution live frames


Screenshot Script (Script)
Capture
Edit & Save
Share
Settings
Resolutions
Resolution Settings
Capture Transformations
Screenshot Resolution Transformations
Live Frames
Screenshot Resolution Live Frames
Background
Cutout
Cutout Set Settings
Burst Settings
Multi-Language Settings
Screenshot Series Settings
Game Object Screenshot Settings
Multi Camera Screenshot Settings
Extra Capture List
Do Not Capture List
Timing
Button Settings
Download Device Frames
Take SceneView Screenshot
Take GameView Screenshot
Take Screenshot with Live Frame(s)
Take All Screenshot Resolutions
Take All Screenshot Resolutions with Live Frames

- Download Device Frames - Downloads a large collection of device bezels/outlines from Facebook and Apple. See their terms of services for usage:
http://facebookmicrosites.github.io/design/disclaimer/
https://developer.apple.com/app-store/marketing/guidelines/#section-products

(Not displayed)
- View Last Screenshot - Open the last screenshot
- View Screenshot Folder - Open the screenshot folder

Additional Buttons
Additional buttons can be added by adding certain scripts to the GameObject with the ScreenshotScript. These scripts are added by default to the prefab and may be removed if you don't need the functionality.

Buttons may also be hidden by unselecting them in Button Settings.

Add ScreenshotBurstScript:
- Take Screenshot Burst - Capture multiple frames
- Take Burst All Screenshot Resolutions - Capture multiple frames in all resolutions

Add ScreenshotSeriesScript:
- Take Screenshot Series - Capture multiple screens with a single press.
- Take Screenshot Series with Live Frames - Capture multiple screens with a single press within each of the frames listed in live frames and screenshot resolution live frames

Add MultiLangScreenshotScript:
- Take Multi-Language Screenshots- Take a screenshot for each language in the languages list
- Take Multi-Language Screenshots with Live Frames- Take a screenshot for each language in the languages list within each of the frames listed in live frames

- Take Multi-Language All Screenshot Resolutions - Take a screenshot for each resolution in the resolution list for each language in the languages list
- Take Multi-Language All Screenshot Resolutions with Live Frames - Take a screenshot for each resolution in the resolution list for each language in the languages list within each of the frames listed in live frames and screenshot resolution live frames

- Take Multi-Language Screenshot Series - Take a screenshot series as mentioned above for each language in the language list
- Take Multi-Language Screenshot Series with Live Frames - Take a screenshot series as mentioned above for each language in the language list within each of the frames listed in live frames and screenshot resolution live frames

Add CutoutScreenshotSetScript:
- Take Cutout Set - Takes a set of cutouts at the selected resolution.

Add GameObjectScreenshotScript:
- Take GameObject Screenshots - Take screenshots of game objects at different resolutions. Perfect for creating promotional shots, capturing item previews, and capturing screens without the need for navigation.

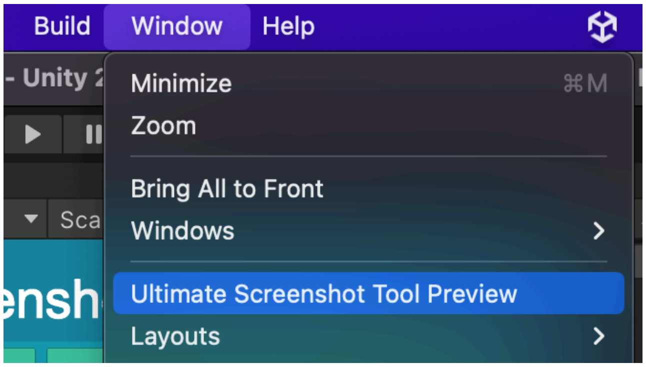
Add MultiCameraScreenshotScript:
- Take Multi-Camera Screenshots - Take screenshots from each of the listed cameras using render textures.

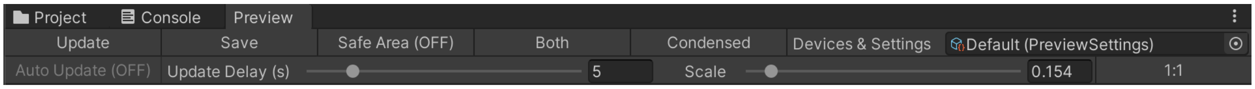
Add MultiCameraGameObjectScreenshotScript:
- Take Multi-Camera GameObject Screenshots - Take screenshots of game objects at different resolutions from each of the listed cameras using render textures.

Preview

Open the Ultimate Screenshot Tool Preview from the Window Menu

A default Preview Settings config will automatically be created. A new Preview Settings config can be created from the Create menu.





Devices & Settings Default (PreviewSettings)

Update Delay (s) 5

Scale 0.154

UI Buttons:

Update - Update the previews (after changing screens, a device, etc.)

Save - Save the currently displayed previews

Safe Area - Toggle showing the safe area on the previews

Portrait/Landscape/Both - Toggle showing only a subset of the devices during the next update

Condensed/Grid - Toggle between condensed and grid views

Devices & Settings - Scriptable object with the list of devices to capture and other settings

Auto Update - Only enabled when the game is playing. Will automatically update the preview at fixed intervals

Update Delay (s) - The time between Auto-updates

Scale - The scale of the previews

1:1 - Resets the scale to 1

Preview Settings

Devices

Type - the type of preview capture

Width/Height - the width/height to capture

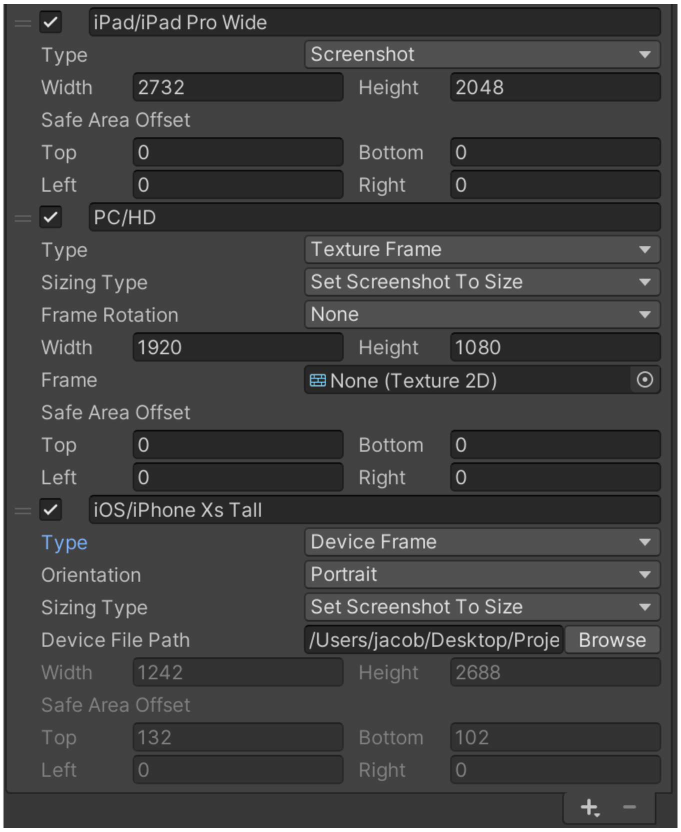
Safe Area Offset Top/Bottom/Left/Right - The safe area for the device. Only used to preview the safe area on top of the device

Screenshot Type - captures a screenshot at the specified size only

Texture Frame Type - captures a screenshot with the specified texture layered on top

Sizing Type - allows toggling to have the input width and height also include the size of the frame

Frame - the texture to use as the device frame



☑ iPad/iPad Pro Wide
Type: Screenshot
Width: 2732 Height: 2048
Safe Area Offset
Top: 0 Bottom: 0
Left: 0 Right: 0

☑ PC/HD
Type: Texture Frame
Sizing Type: Set Screenshot To Size
Frame Rotation: None
Width: 1920 Height: 1080
Frame: None (Texture 2D)
Safe Area Offset
Top: 0 Bottom: 0
Left: 0 Right: 0

☑ iOS/iPhone Xs Tall
Type: Device Frame
Orientation: Portrait
Sizing Type: Set Screenshot To Size
Device File Path: /Users/jacob/Desktop/Proje Browse
Width: 1242 Height: 2688
Safe Area Offset
Top: 132 Bottom: 102
Left: 0 Right: 0

Device Frame Type - captures a screenshot using the size and frame texture from the Unity Device Simulator Plugin: https://docs.unity3d.com/Packages/com.unity.device-simulator@3.0/manual/index.html

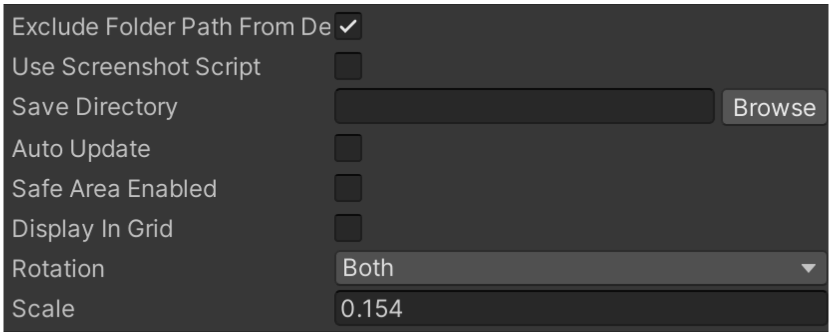
Orientation - allows specifying the orientation to capture

Device file path - the file path to the Device Asset in the Device Simulator plugin. Should automatically find the proper folder. If not, should be at: Assets/../Library/PackageCache/com.unity.device-simulator@{version number}/Editor/SimulatorResources/DeviceAssets

Width, Height, and Safe Area Offsets will be updated automatically. The device texture is also automatically set.

Other Device Settings

Exclude Folder Path From Device Display Name - Exclude the folder path from the device display name under the preview screenshot



Exclude Folder Path From De	☑
Use Screenshot Script	□
Save Directory	Browse
Auto Update	□
Safe Area Enabled	□
Display In Grid	□
Rotation	Both ▾
Scale	0.154

Save Directory - the location where previews will be saved when the save button is tapped

The rest of the settings mirror what may be set by buttons on the preview window.

All Tools

Cameras and Audio:

This contains a UI camera used to capture the UI during camera capture modes. No need to adjust. Screenshot audio source contains the audio source to play when a screenshot is taken if that feature is enabled.

Canvases:

Reminder Canvas - example of how to quick share to Twitter. It contains buttons that by default will only activate in the editor and only during indie dev hour and during Screenshot Saturday. The Instant Share Panel is activated to do this.

Gridlines Canvas - can be activated to show a set of rule of thirds gridlines that will automatically be hidden before a screenshot is taken.

Debug Text Canvas - can be activated to show customizable debug text to display when taking screenshots. Useful to display debug information when the screenshot is taken. This is great for recording errors during QA or generally showing the progress of the game.

Outline Canvas - can be activated to show an adjustable red highlight square. Also useful for pointing out issues when debugging.

Cutout - used to create cutouts (crop). The adjusted cutout canvas is used as a parent for components that should be positioned relative to the cutout. Useful for pinning logos or text to the corner for example. Cutout canvas contains a list of canvases that can be used either as the main cutout or used in a cutout set to get multiple different cutouts of the same image. Adding game objects as children to a specific cutout will enable only activating that game object for that specific cutout.

Mouse - used to allow an in game mouse. That’s useful if you want to capture the mouse pointer within the screenshot. The scene mouse is a 3D game object and the mouse canvas contains a UI-based mouse.

Texture Tools:

Gradient texture generator - used to generate a gradient texture.

Group texture combinator - combines pairs of textures by matching file names in a pair of directories across folders.

Texture combinator - combines two textures.

Texture editor - can be used to apply a set of transformations to a texture. Transformations include solidify, cutout, resize, layer behind, and layer in front.

Share Keys

You’ll need to input your api keys. Press the "Create New Config" in the Share tab to create a new config. It’ll be generated in CaptureTool/Configs or double click to open it.

See https://apidocs.imgur.com/ for details on getting the Imgur client id, client secret, and refresh token. If you’d like to use Imgur in anonymous mode (unattributed to an account), then you can just use a client id. See imgur docs for terms of use.

See https://developer.twitter.com/en/apps for details on getting Twitter keys. Consumer keys are to allow users to post (requires pin auth) and access tokens allow direct posts to your account.

Giphy keys are for use with the `Ultimate Gif Tool` or `Ultimate Screenshot & Gif Tool`. Ultimate Screenshot Tool can be upgraded to Ultimate Screenshot & Gif Tool through the Unity Asset store.

Tool Setup
Functionality may be added or removed from the ScreenshotTool by adding and removing components. Such components include: ScreenshotSeriesScript, MultiLangScreenshotScript, GameObjectScreenshotScript, CutoutScreenshotSetScript, MultiCameraScreenshotScript, ShareScript, and UpdateShareWithScreenshotScript

Features

- Easily take **HD screenshots** in Editor and in Game!
- **Preview** all your devices and be **pixel perfect** on all platforms!
- Make marketing easy! Meme creating tool, instant share buttons, and more!
- Support for **ALL pipelines**: HDRP, URP/LWR, Custom, and Built-In Render Pipelines!
- Save on **ALL platforms**: Windows, Mac, Linux, WebGL, iOS, and Android!
- With **Save to Gallery** on iOS and Android!

- Capture all your app store screenshots in **one click!**
- Super fast captures with async mode and background saving!
- PNG, JPG, TGA, and EXR **formats!**

- No coding required. Full source code included anyway!
- Texture transformations: crop, layer, resize, filters, shaders, and more!
- Quality support, demo, and documentation!

- Boost your productivity:
- All your app store resolutions in one click!
- Live Frames to capture localized device-specific outlines at each resolution!
- Instant share buttons! #screenshotsaturday and #indiedevhour buttons that appear at exactly the right time and enable you to share to Twitter in 2-clicks!
- **Texture transformations** to crop, layer, resize, and apply shaders to **ANY texture**.
- **Localization support** for Unity Localization and custom frameworks.
- Screenshot filters like gray scale, inverted, and sepia!
- **Plus tools** to create gradients, layer, and edit textures.
- Use for debugging with options to add version information, date, and custom information to your screenshots. Plus tools for highlighting bugs!
- **Hotkeys** for quick capture during gameplay.

- Show screenshots in game to highlight cool gameplay moments!
- Capture multiple photos at once: burst, multiple resolutions, multi-language, multi-screen, multi-cutout and more!
- Texture tools to create gradients, layer textures, cutout, resize, and solidify!
- Take SceneView shots for dynamic angles without breaking your cameras!
- Loads of capture options: layer textures into the background or foreground of your screenshots, create custom cutouts to capture different sections of the screen, and specify objects show or hide objects during capture! Add custom logos, hide busy UI, or show a build number by selecting GameObjects to automatically enable/disable during screenshots! Totally customizable!

- Edit your screenshots after capture!

- View the last screenshot or the screenshot folder with the touch of a button.
- ScreenCapture for simplicity and RenderTexture for customization capture modes!
- DontDestroyOnLoad option with features to automatically update the cameras when you switch scenes (during gameplay).
- Save in JPG, PNG, TGA, and EXR! With JPG quality customization and optional PNG transparency!

- Includes custom code to show the Canvases in render mode "Screen Space - Overlay." A special case that many **other plugins don't handle**!
- Tested repeatedly for many use cases (GUILayout, all Canvas render modes, render textures within the scene, and changes made just before/after screenshots).
- Events to allow programmers to create custom code to handle their own special cases.
- Optional gridlines to help you line up shots that disappear before you take them.

- Additional scripts such hotkey scripts, GameView controls, & various extensions!

Quick Fixes

Duplicate Symbols
- Files occasionally must be moved for new features or to clean up the project. A migration script has been added to automatically remove the old files. If the project has duplicate symbol errors, close and re-open Unity to trigger this script (UpdateScript).

Unity Localization Import Issues
- Building the project can resolve issues where Unity Localization is not imported.

Frame Imports
- Ensure that all downloaded frame textures have Read/Write Enabled under Advanced in the texture import settings.

Screen Resolution
- Screen.SetResolution will adapt itself to the closest supported aspect ratio unless "other" is a supported aspect ratio. You can edit supported aspect ratios within Build Settings>Player Settings->Resolution->Supported Aspect Ratios
- Selecting full screen in Standalone builds does not properly update the resolution if "Resizable Window" is not set in Edit->Project Settings->Player->Resolution and Presentation. Checking that checkbox or selecting fullscreen from the start to use the proper resolution.

Cameras/Canvases layering
- Using the same canvas sort order or camera depth on multiple canvases/cameras can cause canvases/cameras to swap layers or even not appear at all.
- Set canvas sort order and camera depth to different values to avoid ambiguity and potential bugs. The higher number will be rendered on top.
- If setting sort order/depth is not possible. You could sort cameras properly in the cameras array (from back to front). Similarly, canvases could be given different planeDistances in CanvasesAdjuster. However, doing so could cause other issues and is not recommended.

UI
- Old style Unity GUI items will only be captured in ScreenCapture mode.
- If your UI is missing from your screenshot, check that your overlay camera's culling mask includes the UI flag.
- If your UI is distorted in RenderTexture mode, that may be due to the CanvasesAdjuster. You can minimize the range of z-positions used in your UI, set your own plane distance override in advance settings, or use screen capture mode.
- You may need to Refresh Canvas List if you add new canvases to the scene after the screenshot prefab.

Scaling

- Scaling usually isn't necessary. It's simpler to use a higher resolution. The scale property is now only used if using the Screen Capture capture mode in Unity 2017.3 or newer and Use Alt Screen Capture is set in Settings. The scale option will be visible in the Resolutions list only if those conditions are true.
- If you get a warning similar to 'GameView reduced to a reasonable size for this system' or are using very large resolutions, you can set adjust scale to true in advance settings. You can then click "Scale Resolutions to Screen" in the resolutions section (or lower your target resolution and set the scale multiplier).
- Sending a non-1 scale into ScreenCapture may throws errors. This is a Unity bug (verified to occur in Unity 2018.2). To get around it, use RenderTexture mode or convert your textures to use a scale of 1.

Sharing

- Screenshots must be persisted locally (outside of mobile photo galleries and web desktop saves) to share files as those saves are not directly accessible by the app.
- If the "Share by Email" or "Email" support buttons are not working, check that your browser is set to handle mailto links. (A Fix for Google Chrome)

Miscellaneous

- "Coroutine couldn't be started because the the game object 'ScreenshotTool' is inactive!" - If this is happening from a button or event, you may have connected to the project prefab instead of the prefab in the scene. Double check that the scene tab is selected when choosing your ScreenTool GameObject. If that doesn't work, Restart Unity. You can also try GameObject (Toolbar) -> Break Prefab Instance as a last resort.
- ShowDuringScreenshotScript/HideDuringScreenshotScript only work in play mode.
- If "View Last Screenshot" or "View Screenshot Folder" button fails, you may be using characters that are invalid for folder/file names. You'll notice if a character in the debug log pathname is different from the actual pathname.
- If you get an error like "Tiled GPU perf. warning: RenderTexture color surface (375x812) was not cleared/discarded. See TiledGPUPerformanceWarning.ColorSurface label in Profiler for info:" you can fix it with Edit -> Graphic Emulation -> No Emulation

Quick Tips

- Code is provided within a separate namespace to avoid naming conflicts. Most scripts are in TRS.CaptureTool, share scripts are in TRS.CaptureTool.Share, and a few extras are in TRS.CaptureTool.Extras (like hot keys and some other UI scripts).

- Additional buttons can be added to to the Screenshot Tool by adding certain scripts to the GameObject with the ScreenshotScript. These scripts are added by default to the prefab and may be removed if you don't need the functionality. (See Getting Started on page 3 for the full list of scripts.)

- Instant share buttons can be found within the Reminder Canvas. You can also add your own!

- Double click a resolution in the resolutions list to set the GameView to that resolution.

- Slashes in the resolution name can be used to further organize your file structure.
- Creating screenshot folder within project is tidy, but can cause longer build times. Hide the screenshot folder from Unity by ending the folder name with a ~ (easiest), starting the folder name with a . (hidden folder), or using another way of creating hidden assets: https://docs.unity3d.com/Manual/SpecialFolders.html
- The screenshot folder may also be ignored by git by adding it to your .gitignore

- Hotkeys (while application is playing) - may be changed in settings
-- alt+f takes a single screenshot
-- alt+r takes screenshots at all resolutions
-- alt+b takes a burst of screenshots (if BurstScreenshotScript is added to tool)
-- alt+n takes a set of cutouts at the selected resolution (if CutoutScreenshotSetScript)
-- alt+c toggles the cutout preview visibility (if cutout is active)
-- alt+t toggles the gridline visibility (in HotKeyActivationScript on "Gridlines Canvas")

- Hotkey for take screenshots at all resolutions changed from v to r. This was due to an overlap with the Vertex Snap Unity Editor hotkey.
You can view all the Unity hotkeys here: https://docs.unity3d.com/Manual/UnityHotkeys.html

- Only one screenshot texture is cached at a time in ScreenshotScript. If you need to persist a texture longer, copy the texture first. UltimateScreenshotTool/Demo/Scripts/ScreenshotDisplayScript.cs has an example of this.

- Temporary enabled/disabled objects automatically appear/disappear during screenshots. Afterwards they will return to their initial state.

For example, the "Gridlines Canvas" automatically disappears during screenshots. Since the tool returns objects to their initial state, if the gridlines were already disabled, they would not be re-enabled after the screenshot.

Similarly, the "Debug Text Canvas" can be set to automatically appear during screenshots. By default, the "Debug Text Canvas" is not listed in Temporarily Enable GameObjects. Add it to that list to displays information about the build (custom text, project, scene, version, or date).
- Extra capture list/do not capture list both work with cameras and canvases. That may be an easier way to exclude them from screenshots depending on your build.

- You can change which buttons are displayed on the share tab in display settings.

- By default, resolutions are shared over all platforms, so you can take all your screenshots at once. If you'd prefer to keep the resolutions unique per platform, enable this feature in advanced settings (uncheck Share Resolutions Between Platforms) to keep your iOS/Android/Standalone/Web/etc. resolutions separate.

- If you don't intend to use screenshot behavior in game, you can tag the screenshot tool as EditorOnly using the EditorOnly tag:
https://answers.unity.com/questions/1010525/exculde-gameobject-from-final-build.html
However, the screenshot scripts will still be added (~82.2kb unless avoided with temporary symlinks)
https://support.unity3d.com/hc/en-us/articles/208456906-Excluding-Scripts-and-Assets-from-builds

- Time scale is temporarily set to 0 during Take All Screenshots. This is to ensure all resolutions capture the same instant and can be changed in Timing in the Capture tab.
- Items that change using an unscaled time value (like Time.unscaledDeltaTime or Time.unscaledFixedDeltaTime) will not be affected by this time stoppage and must be handled with their own custom code. Pausing when the WillTakeMultipleScreenshots event is received and restoring when the MultipleScreenshotsTaken event is received would be a simple and clean method.

- You can add/remove extra resolutions to the GameView with the Additional Resolutions menu in Tools. Or add your own custom buttons in Ultimate Screenshot Tool/Helpful Extras/Scripts/Extensions/ResolutionExtensionsGameView.cs

Programming API

ScreenshotScript
Most functionality will be accessed through the ScreenshotScript. Some additional functionality can be found in the add-on scripts such as the ScreenshotBurstScript, ShareScript, or scripts attached to that GameObject.

There are static system actions that can be subscribed to for all events:

```
public static System.Action<ScreenshotScript> WillTakeMultipleScreenshots;
public static System.Action<ScreenshotScript, int (width), int (height), int (scale)> WillTakeScreenshot;
public static System.Action<ScreenshotScript, RawFrameData> RawFrameTaken;
public static System.Action<ScreenshotScript, Texture2D> ScreenshotTaken;
public static System.Action<ScreenshotScript, Texture2D> WillSaveScreenshot;
public static System.Action<ScreenshotScript, string (filePath)> ScreenshotSaved;
public static System.Action<ScreenshotScript> MultipleScreenshotsTaken;
```

General action inputs:
ScreenshotScript - ScreenshotScript performing the action
RawFrameData - Object containing unprocessed screenshot texture or render texture
Texture2D - Texture captured or being saved

The capture methods of ScreenshotScript are asynchronous, so you'll want to subscribe to ScreenshotTaken to get the texture while using the tool programmatically.

There’s also an IEnumerator returning version of each method. If you’re familiar with Coroutines, you could call these methods with ‘yield return StartCoroutine([method]);’ which would wait until that method is complete before continuing.

Other public methods are:

```
public void TakeLiveFrameScreenshots(bool save = true)
public void TakeLiveFrameScreenshotsWithTexture(Texture2D textureToFrame, bool save = true)
public IEnumerator TakeLiveFrameScreenshotsCoroutine(bool save = true)
public IEnumerator TakeLiveFrameScreenshotsCoroutine(bool saveRawScreenshot, LiveFrame[]
liveFrames, bool save = true)
        public IEnumerator TakeLiveFrameScreenshotsWithTextureCoroutine(Texture2D textureToFrame,
LiveFrame[] liveFrames, bool save = true)

public void TakeAllScreenshots(bool save = true)
public void TakeAllScreenshotsWithLiveFrames(bool save = true)
public IEnumerator TakeAllScreenshotsCoroutine(bool save = true)
public IEnumerator TakeAllScreenshotsWithLiveFramesCoroutine(bool save = true)
public IEnumerator FlexibleTakeAllScreenshotsCoroutine(bool withLiveFrames = false, bool save = true)
```

```
public void TakeSceneViewScreenshot(bool save = true) (Editor only)
IEnumerator SafeTakeSceneViewScreenshot(bool save = true) (Editor only)
```

```
public void TakeSingleScreenshot(bool save = true)
public IEnumerator TakeSingleScreenshotCoroutine(bool save = true)
```

```
public IEnumerator TakeScreenshot(Resolution resolution, Rect cutoutRect, bool save = true,
ScreenCaptureConfig screenCaptureConfigToUse = null, TextureTransformation[] transformations = null,
bool calledFromLoop = false)
```

```
public IEnumerator AsyncScreenshot(Rect cutoutRect, Resolution resizeResolution, bool solidify = true,
TextureTransformation[] transformations = null)
```

```
public IEnumerator TakeScreenshotWithCameras(Camera[] camerasToUse, Resolution resolution, Rect
cutoutRect, bool save = true, ScreenCaptureConfig screenCaptureConfigToUse = null,
TextureTransformation[] transformations = null, bool calledFromLoop = false)
```

This last method exposes an option to specify which cameras to use (only used in render texture mode), the resolution to use, a cutout rect, whether to save, a config for screen capture mode details, transformations to perform on the texture, and if this method is being called from a loop (only set this if you’re manually calling the code that would skipped).

To get a texture directly you could use the following:

```
public Texture2D Screenshot(Camera[] camerasToUse, Rect cutoutRect, Resolution resizeResolution,
bool solidify = true, int scale = 1, List<TextureTransformation> transformations = null)
```

Again, the cameras are only used in render texture mode (not the case by default) and solidify indicates whether alpha should be allowed in the final image or not. A null cutoutRect can be used for no cutout and a resolution with a width or height of 0 can be used to not resize the texture after the screenshot is taken.

However, this method skips the preparation steps (like showing/hiding various items), sending out the actions to the subscribers, any delays set with the delay before capture, and the WaitForEndOfFrame call which may be necessary depending on your use case. It's not particularly recommended unless you know what you're doing. At the very least, you’ll want to call: `yield return new WaitForEndOfFrame();` before this method.

```
public RawFrameData RawScreenshot(Rect cutoutRect, Resolution resizeResolution, bool solidify = true,
ScreenCaptureConfig screenCaptureConfigToUse = null)
public RawFrameData RawScreenshot(Camera[] camerasToUse, Rect cutoutRect, Resolution
resizeResolution, bool solidify = true, int scale = 1)
```

Raw Screenshot returns a RawFrameData. It uses slightly less compute than the Screenshot method, but requires processing later. Very likely unnecessary.

Save with:

```
public void Save(Texture2D texture, string overrideFilePath = "", bool destroyTexture = false)
```

You can use your own filepath or the one from the tool by passing in an empty or null overrideFilePath.

Sharing

For the sharing, that will really depend on what you want to do. There's a whole variety of sharing options with APIShare and WebShare. Optionally, using ShareScript will allow you to have the options pre-filled to the variables that are set in the ShareScript.

The ScreenshotSaturday Button under ScreenshotTool -> Reminder Canvas -> ScreenshotSaturday Time has a great example of doing a full capture and share flow in game. To override with your own UI, I suggest copying and modifying the Reminder Canvas externally to the tool (to avoid any overwrite issues when the tool is upgraded though it won't be an issue if you've diverged from the prefab instance). You can diverge from the prefab with GameObject -> Break Prefab Instance.

Other Helpful Scripts

There's scripts that can be attached to GameObjects to show them or hide them during screenshots (ShowDuringScreenshotScript and HideDuringScreenshotScript). Using Extra Capture List/Do Not Capture List may be simpler though.

Other useful scripts can be found in the Demo and Shared folders.

For the most part, using any of those scripts will be unnecessary or can be found in the Demo. Anything in an Editor folder can safely be ignored.

There's a Texture2D extension to blend textures together:

```
public static Texture2D Blend(this Texture2D background, Texture2D foreground, Vector2Int position,
Color emptySpaceFillColor, bool alphaBlend = true, bool apply = true, bool destroyOriginals = true)
```

Called like so:

```
(background texture reference).AlphaBlend((foregroundTexture)…);
```

Background - texture layered behind
Foreground - texture layered in front
Position - place of the foreground's top left relative to the background's top left. The position can have negative values.
EmptySpaceFillColor - if there's a space where neither the background or foreground texture has a pixel in the resultant texture use this color to fill. Color.clear is the default option. (say a super wide and a super tall texture is combined, the corners would be empty space)
AlphaBlend - see the Transparency section for the difference in AlphaBlend and the default Unity blend (SLBlend). SLBlend is used if AlphaBlend is false.
Apply - apply the changes or not. Set to true unless making further modifications.
DestroyOriginals - destroy the input background and foreground textures (if possible).

That could be useful for adding in device-shaped outlines, logos, overlays, etc. that you do not want to temporarily appear on screen (like with an Extra Capture List item for example). It can also be used through transformations on the texture.

Additional Plugins

Several additional features were explored. However, they were found to add excess code and UI complexity for relatively rare use cases and without providing significant benefit. Instead, here is a guide for integrating other plugins.

360° Screenshot Capture
https://assetstore.unity.com/packages/tools/camera/360-screenshot-capture-112864
Free
This plugin creates 360 degree screenshots. It returns an array of bytes which may be saved using the Ultimate Screenshot Tool. The Save exposed by ScreenshotScript would be able to save those bytes using the same functionality as the Ultimate Screenshot Tool.

Facebook 3D Photos
https://assetstore.unity.com/packages/tools/integration/fb-3d-photos-instant-3d-photos-from-editor-and-runtime-144646
Free
This plugin creates Facebook 3D photos. It saves the screenshot and depth to the folder specified by the input path. The same file naming settings may be used from {ScreenshotScript instance}.fileSettings.FullFilePath({folderName}, {fileName}, {createDirectory}). An "" (empty string) file name may be used to use the default filename including the counter. Ensure that the type of image (png) matches the type of image set in fileSettings.

If using the counter, don't forget to call the following to update the count:
{ScreenshotScript instance}.fileSettings.IncrementCount();
{ScreenshotScript instance}.fileSettings.SaveCount();

Full Overview

Screenshot Tool

Capture Tab

Resolutions
- List of resolutions that screenshots should be taken in.
- Double click a resolution to switch the GameView to that resolution.
- Resolutions only used when take all screenshots button is pressed/called from code.
- None, Portrait, Landscape, and All change which resolutions are active (will be used when taking screenshots).
- Scales and Delays may also optionally be set in this menu once those options are enabled from the Settings Tab under Resolution Settings. Both are usually unnecessary.

Capture Transformations
- Ordered list of transformations to apply to captured screenshots before save.

Screenshot Resolution Transformations
- Ordered list of transformations to apply to a specific screenshot resolution before save. These are keyed to the screenshot resolution name, so both must match.

Live Frames
- Create a frame around the last capture screenshot. A similar function could be performed by layering a texture with a texture transformation. Live frames enable more dynamic behaviors like text that is updated to the current language localization.

- Save raw screenshot - whether to save the initial capture or not. Clear to only save the screenshot in the frame.
- Active - whether the frame should be used. Inactive frames are collapsed.
- Name - name used when saving the frame. Usage may be customized in file name settings.
- Texture Destination - the Raw Image that the screenshot should be rendered in. This is the destination of the texture.
- Active Only If GameObject Active - Conditionally only use this frame is a particular game object is active. For example, a frame could be specific to an individual screen, so the frame is only used if that screen is displayed (used in the demo).
- GameObject Required to Be Active - GameObject referenced in the above condition.

- Override Resolution - Set the resolution at which to capture the frame. Useful if the frame should not be captured at the same resolution as the screenshot.
- Width/Height - width and height at which to capture the frame.

- Temporarily Enabled/Disabled GameObjects - GameObjects temporarily enabled or disabled while the frame is being captured (like a HUD). Only this list is used. The temporarily enabled/disabled game objects of the overall tool are ignored during frame capture.
- Transformations - transformations to apply to the capture frame.

Screenshot Resolution Live Frames
- Same as Live Frames with the exception that these frames are specific to a resolution from the resolutions list. This can be used to give different frames to different devices.

Background
- Override the background of a camera with a custom color. Useful to green screen something or create a transparent background for promotional materials.
- Defaults to the camera to override as the one with the lowest depth.

Cutout
- Cutout a portion of the screen for your screenshot. Useful if you want to crop something out or just capture a part of the screen. Great for promotional images.
- Cutouts use the name of the cutout game object as the folder path.
- Preview overlay will automatically be hidden during screenshots.

- Use Cutout
— Choose to use cutout or not.
- Cutout Script
— Cutout is specified by the RectTransform of a GameObject with a CutoutScript and Graphic attached. (Image and RawImage are examples of Graphics.)
- Adjusted Rect Transforms
— Transforms to have their parents set to the cutout graphic to give the effect of the cutout being the full screen size. For example, if you have a logo in the bottom corner of the screen that would get cut off by your cutout, you can add it to this list to have it put in the corner of your cutout.

Cutout Set Settings
- Takes multiple cutouts at the same time.

- ScreenshotScript
— The ScreenshotScript used to take the screenshots. Only shown if ScreenshotScript is attached to a separate GameObject, otherwise, it is automatically set.
- Take Cutout Set Key Set
— Specify a hot key (or key combination) to take a cutout set.
- Override Resolution
— Check to change the resolution before capturing.

- Width/Height
— The resolution width and height to change to before capturing the cutouts.
- Override Adjusted Rect Transforms
— Override the adjusted rect transforms set in the cutout settings.
- Adjusted Rect Transforms
— Same as in cutout settings. Just overrides those settings.

Burst Settings
- Burst takes several screenshots one after another. Useful to capture the perfect frame of epic action moments.

- ScreenshotScript
— The ScreenshotScript used to take the screenshots. Only shown if ScreenshotScript is attached to a separate GameObject, otherwise, it is automatically set.
- Show Take All Button
— Show a button that takes a burst of screenshots in all resolutions. With additional resolutions, the time between frames goes up. May be difficult to time properly.
- Take Screenshot Burst Key Set
— Specify a hot key (or key combination) to take a screenshot burst.
- Burst Size
— The number of screenshots to take in a burst.
- Skip Frames
— The number of frames to skip between captures. 0 captures every frame, 1 captures every other frame, 2 captures eery 3rd frame, etc.

Multi-Language Settings
- ScreenshotScript
— The ScreenshotScript used to take the screenshots. Only shown if ScreenshotScript is attached to a separate GameObject, otherwise, it is automatically set.
- ScreenshotSeriesScript
— The ScreenshotSeriesScript used to take the screenshots. Only shown if ScreenshotSeriesScript is attached to a separate GameObject, otherwise, it is automatically set.
- Languages
— Specify the languages you want to capture in your screenshots.
— Be sure to put the original language last, so that your UI returns back to normal.

Screenshot Series Settings
- ScreenshotScript
— The ScreenshotScript used to take the screenshots. Only shown if ScreenshotScript is attached to a separate GameObject, otherwise, it is automatically set.
- Capture Initial Screen
— Set to capture the screen that is visible when the button is pressed.
- Button Interaction list
— Set a button to press (to move within your game/app), a delay (may be used without a button set), and whether or not to take a photo afterwards.

— Examples: The button likely moves you to a different place in your app, but you could also tie it to a button that creates an action and time the screenshot after that event. The delay could be used to wait for animations to take place or for enemies to spawn. The take a photo check may be left unchecked to avoid taking repeated photos when going back up a hierarchy (entering a screen and then returning home).

Game Object Screenshot Settings
- ScreenshotScript
— The ScreenshotScript used to take the screenshots. Only shown if ScreenshotScript is attached to a separate GameObject, otherwise, it is automatically set.
- GameObject list
— The list of GameObjects that should be enabled one at a time before a screenshot is taken.

- Resolution list
— The resolution to be used when screenshooting the corresponding GameObject.
— If the same resolution should be used for all screenshots, entering only a single resolution will work. If the GameObject list is larger than the resolution list, the last resolution will be used repeatedly for the remaining GameObjects.
— Useful for a variety of scenarios. I use it to take all my Asset Store images at the same time. I have separate GameObjects for what I want to display in each image and then I specify the resolution to match it. For example, I have a GameObject called "Icon Canvas (128x128)" that will be enabled, the resolution will change to 128x128 as specified in the resolution list, and a screenshot will be taken. This process is repeated for the Small, Large, and FB images. (The last GameObject is disabled, the new one is enabled, the resolution is changed, and the screenshot is taken.)

Multi Camera Screenshot Settings
- ScreenshotScript
— The ScreenshotScript used to take the screenshots. Only shown if ScreenshotScript is attached to a separate GameObject, otherwise, it is automatically set.

- Camera list
— The list of Cameras whose RenderTexture should be captured as a screenshot.

Extra Capture List/Do Not Capture List
- Enable/Disable certain GameObjects during your screenshots. Super useful for hiding debug info or automatically adding your logo to all your screenshots.
- The "Gridline Canvas" is in the Do Not Capture list by default. Enable it in the ScreenshotTool prefab for a demo.

Timing
- Stop Time to Capture
— Sets Time.timeScale to 0 before capture and restores it to its initial value afterwards.
— Useful to ensure that the same frame is captured in all resolutions.
— Items that change using an unscaled time value (like Time.unscaledDeltaTime or

Time.unscaledFixedDeltaTime) will be unaffected by this time stoppage and must be handled with their own custom code. Pausing when the WillTakeMultipleScreenshots event is received and restoring when the MultipleScreenshotsTaken event is received would be a simple and clean method.

- Delay Before Capture
— Delay in seconds between when button is pressed and when screenshot is taken. Could be useful for exact timing, a countdown, or another use.
- Time Scale Override
— Only appears while game is playing.
— Allows slowing down/speeding up time to precisely capture a super epic moment.

Button Settings
- A series of checkboxes to show or hide the various screenshot buttons. Use to customize the UI to your liking.

Edit & Save Tab

Edit
- Texture to edit - the texture to edit. Automatically set to the last screenshot captured by the tool. Can be overwritten by any other texture to edit.
- Edit Transformations - transformations to apply to the texture to edit.
- Apply Transformations - transformations are applied on a copy; the original is not lost.
- Restore to last screenshot - useful to undo changes.
- Save - save the edited screenshot.

Save Path
- The directory in which to save screenshots.
- Can be entered manually or selected using the Browse button.
- Recommended to be a folder ending with the ~ symbol (and is by default), so the folder is ignored by Unity. This saves time when switching builds. You could also ignore the folder in your .gitignore for cleaner/lighter commits.

Standalone Save Settings
- The directories to save screenshots in for Standalone builds (relative to the data path).
- The default values will create the folder at the same hierarchy level as the game.

Mobile Save Settings
- The settings for if/where to save the screenshots within Mobile builds.
- Save to Gallery will save the screenshots to the standard photo app location. (Photos for iOS or the Gallery for Android.) A custom album name may optionally be specified.
- Persist Locally will save the screenshots to the specified directory on the user's device. The specified directory is relative to the persistent data path. These files may be inaccessible to the user, so it is recommended to delete them when no longer in use.

WebGL Save Settings
- The settings for if/where to save the screenshots within WebGL builds.
- Open In New Tab will open the screenshot(s) in a new tab, so the user can choose what to do with them (download, share, etc.). This behavior may be automatically blocked by the browser and the user may have to manually allow it.
- Download will download the screenshot like any other file. More reliable than opening in a new tab as it typically requires less confirmation/is easier to notice when blocked.
- Persist Locally will save the screenshots to the specified directory on the user's device. The specified directory is relative to the persistent data path. These files may be inaccessible to the user, so it is recommended you delete them when no longer in use.

File Name Settings
- Specify a pattern for your file names to keep a pleasant naming convention.

— A custom prefix of whatever text you would like is optional.
— Toggle tabs to enable/disable a part of the file name (see Example File Name).

— If the date tab is enabled, the date format may also be specified.
— Counter is a number that automatically increments for each unique screenshot. (Screenshots taken at the same time will keep the same count to be easily identified.)
— Reset Counter will return the counter to 0.
— PNG or JPG file format may also be specified by selecting the appropriate tab.
— PNG files will have an additional option to Allow Transparency. This should most likely be left off unless specifically need. See the Transparency section for more details.
— JPG files have a 1-100 quality option. The higher quality values produce better image quality, but larger file sizes.
— Frame Description Before/After Prefix Toggle - Whether to include the frame description before or after the custom prefix. Only used with Live Frames.
— Frame Description Prefix - A prefix to add to the Live Frame Name. For example, if the frame is named ‘RainbowFrame’ and the prefix is ‘In', the frame description would be ‘InRainbowFrame’. Only used with Live Frames.
— Frame Description Suffix - A suffix to add to the Live Frame Name. For example, if the frame is named ‘RainbowFrame’ and the suffix is ‘Around', the frame description would be ‘RainbowFrameAround’. Only used with Live Frames.
— Include language in path - whether to include the language in the path. Multi-language captures automatically include the language.
— Save in background - whether to save in the background.
— Thread Priority - thread priority of the thread tasked with saving.

Share Tab

Display Settings
- Select which share options will be shown. Remove unused buttons to clean up the UI.

Share Settings
- Select default title, text, and game url options to be used if no other value is specified. These values are used when appropriate across all share methods.

Select File to Upload
- Select the path of the file you wish to upload by entering it or clicking the browse button. Alternatively, if the UpdateShareWithScreenshotScript is attached to the tool, the last screenshot taken file path will be used.

Select Url to Share
- Select the url to be shared using the other share methods.
- All share methods will still work without any share url selected.
- Use Custom Url allows you to enter a custom url to be shared. This could be useful if you'd like to share a link to another post or a custom link you create.
- Use Custom Url Function (only available if Use Custom Url is selected) uses the CustomUrlFunction set in the ShareScript to create a url. This allows the most customization, but requires some coding.
- Use Last Media Post Url uses the url of the post created from the last media upload.
- Use Last Media Url uses the url for just the media from the last media upload.
- If no url is set to share, the game url from Share Settings will be used.

Copy Game Url/Last Media Url/Url to Share
- Copy the chosen url to your clipboard. These options only appear if that url is available and the button is not disabled in Share Settings.

Upload to Server/Imgur/Giphy
- Uploads the file to upload to the selected service, so the url can be shared using the other sharing methods.
- Each method has its own parameters unique to the service.
- Giphy is only available for gifs, so the buttons will disappear when taking screenshots.

Share to {…}
- Share buttons open a url with the specified parameters set.
- Share methods will not appear if disabled in Share Settings.

Settings Tab

Capture Mode
- Set the capture mode.
- Screen Capture is simpler to use as it captures the screen as it is displayed. (After custom settings like background, temporarily enabled/disabled objects, etc.)
- Render Texture grabs the render texture of each camera and layers them together. Useful for specific custom cases like Case 3 in the transparency section.
- All Cameras is Render Texture with All Cameras automatically included.
- Async asynchronously grabs the texture from the GPU to improve performance.

Cameras
- Only in RenderTexture mode.
- Specify the cameras that should be captured in the screenshot.
- Add All Active Cameras adds the currently active and enabled cameras to the list.

UI Camera
- Autoswitch Render Mode
- Only in RenderTexture and All Cameras modes.
- Canvases in overlay render mode will not be captured by any individual camera, so would normally be left out of screenshots. This feature temporarily switches those canvases to camera render mode during screenshots, so your UI will appear.
- UI Camera
- Only in RenderTexture and All Cameras modes.

— The camera which should display your overlay canvases. This is traditionally the topmost camera so your UI is not hidden behind another camera layer.
— Defaults to a camera within the tool that only displays UI, has a high depth, and uses deferred rendering.

- Override Plane Distance
- Only in RenderTexture and All Cameras modes.

— Toggles if the plane distance should be overwritten at all. Likely unnecessary if using the default UI camera.

- Plane Distance Override
- Only in RenderTexture and All Cameras modes.

— The plane distance to display your overlay canvases. This is the distance from your camera at which the canvas should be displayed.
— A value of 0 will use the default near clipping plane calculation.
— By default, it is approximately equal to your camera's near clipping plane to mimic the behavior of an overlay. If having issues, try reducing the z-Position of your UI to 0.

- Refresh Canvas List button
- Only in RenderTexture and All Cameras modes.

— Refreshes the list of canvases that should be automatically switched from overlay mode to canvas mode during screenshots.
— Not done automatically to avoid frequent costly loops through the object hierarchy.

— It may be triggered automatically by other events.

HotKeys
- Take one or all your screenshots with the press of a single key.
- Customizable. Set each action to whatever key (or key combination) you'd like.
- Great for taking screenshots right in the middle of the action.

Audio Settings
- Screenshot Sound
— Audio source containing the camera shutter sound.
- Play in Editor
— Toggle if the camera shutter sound should play when taking screenshots in editor.
- Play in Game
— Toggle if the camera shutter sound should play when taking screenshots in game.

Mouse Settings
Allows you to capture the mouse in screenshots. The in game mouse will be visible.
- Show In Game Mouse
— Whether or not the in game mouse should be visible.
- Mouse Follow Script
— The type of mouse to use. In the prefab, there's a traditional UI-based mouse (Mouse Graphic) or a GameObject with a 3D object (Scene Mouse).
- Hide Original Mouse
— Whether or not the device mouse should be visible.

Resolution Settings
- Adjust Scale
— Allows you to set a scale value on your resolutions. May be useful if taking high resolution photos and scaling down or if Unity is limiting the GameView size.
- Adjust Delay
— Allows you to set a delay before taking a screenshot. This is likely unnecessary so it is off by default. Possibly useful if Unity if delaying for an animation that occurs when rotating between landscape and portrait.

- Share Resolutions Between Platforms
— When enabled the resolution list stays the same even when you switch platforms, so you can still get all your screenshots without switching between multiple platforms.
— When disabled to the resolution list is tied to the platform it was created on, so you can take just the screenshots necessary for that platform.
— Desired setting likely depends on if there are visual differences between your platforms, if not, it's likely simpler to leave this enabled.

- SceneView Screenshot Resolution
— Changes the resolution used for SceneView screenshots.
— By default this is the camera resolution of the SceneView so the screenshot appears exactly how you see it in the Editor.

— Other options include the current GameView resolution or the Default resolution (the first resolution in the resolutions list).

DontDestroyOnLoad Settings

- DontDestroyOnLoad

— Sets the ScreenshotTool GameObject to DontDestroyOnLoad. This allows you to set-up the ScreenshotTool in a single scene & continue to use it even when scene switches.
— DontDestroyOnLoad only preserves the GameObject between scene transitions. It will not create a ScreenshotTool that does not already exist.
— To customize behavior to each scene, you could disable this behavior & use multiple tools. Alternatively, you could create a script to modify settings on scene changes.

- Max Instances

— Only shown if DontDestroyOnLoad is true.
— Max instances of the screenshot tool that should exist. Used to avoid duplication of the tool when returning to the scene in which the screenshot tool is created.
— Usually no more than 1 screenshot tool is necessary. This option is available if separate tools are used for landscape and portrait screenshots or other cases.

- AutoUpdate Cameras

— Only shown if DontDestroyOnLoad is true.
— Attempts to automatically grab the same cameras after a scene transition during which the original cameras may have been destroyed.
— This applies to the background and overlay cameras and the cameras list if RenderTexture mode is enabled.

- Find Cameras by Name/Tag button

— Only shown if DontDestroyOnLoad and AutoUpdataCameras are true.
— Toggles between finding the cameras by name or tag after scene transitions.
— In order to find the replacement cameras after a scene transition, there needs to be ways of identifying the replacement camera. You could either keep the corresponding camera names the same between scenes or have them share the same tags.
— By default the “Main Camera” can be found by either method as it has the standard “Main Camera” name and “MainCamera” tag when created.
— Multiple cameras with the same name (or tag depending on this setting) will cause unpredictable behavior.

Support

- Upload Editor Error Logs

— Option to upload the current error logs to aid in debugging.

- Email/Twitter/Discord/Reddit

— Support contact information listed in order of ease of use.
— I try to respond within a couple days.

- Rate/Donate/Other Assets

— If you're happy with this asset, please give it a heart and 5 stars! You can also check out my other assets.

Transparency

Case 1: I just want a solid image with no transparency.
Solution: Uncheck Allow Transparency in Save Settings.

Case 2: I want an image with a transparent background.
Solution: Check Allow Transparency in Save Settings (and don't put any transparencies over things that should appear transparent).

Case 3: I have a more complex case. Parts with transparent overlays need to be 100% solid and other parts should have be partially transparent.
Solution: Use Render Texture mode. Put your transparent top layers onto a separate camera. Alpha blending between camera textures will preserve the proper alpha values.

Reasons:

- Unity's standard method of alpha blending is additive. (See https://docs.unity3d.com/Manual/SL-Blend.html)

Here's an example:

```
Color backgroundColor = new Color32(128, 128, 128, 255);
Color foregroundColor = new Color32(255, 255, 255, 128);
Color resultColor = foregroundColor * foregroundColor.a + backgroundColor * (1f -
foregroundColor.a);
Debug.Log(string.Format("Result color: {0} {1} {2} {3}", ((Color32)resultColor).r,
((Color32)resultColor).g, ((Color32)resultColor).b, ((Color32)resultColor).a));
Debug.Log(string.Format("Result color: {0} {1} {2} {3}", resultColor.r, resultColor.g,
resultColor.b, resultColor.a));
```

Produces:

```
// Result color: 191 191 191 191
// Result color: 0.7519569 0.7519569 0.7519569 0.7500038
```

Changing the foreground alpha to 10:
Produces:

```
// Result color: 132 132 132 245
// Result color: 0.5214918 0.5214918 0.5214918 0.9623222
```

Changing the foreground alpha to 240:
Produces:

```
// Result color: 247 247 247 240
// Result color: 0.9707036 0.9707036 0.9707036 0.9446367
```

In other words, resultColor.a = foregroundColor.a * foregroundColor.a + backgroundColor.a * (1f - foregroundColor.a);
With backgroundColor.a = 1, resultColor.a = foregroundColor.a * foregroundColor.a + (1f - foregroundColor.a);

As you can see, despite having a solid color background the resulting color is partially transparent. The same result will happen to games with partially transparent.

To fix this, we want to stop our transparent overlays from mucking around with our solid backgrounds. To do this, we can put them in separate cameras.

This works, because Unity uses its additive alpha blend on all the elements within a camera. However, the camera layers are combined using custom code in this tool.
The custom code in this tool uses a different formula for alpha that ensures that solid backgrounds blended with transparent foregrounds stay solid.

The formula for the alpha in the tool's alpha blend is here:
resultColor.a = foregroundColor.a + backgroundColor.a * (1f - foregroundColor.a);
With backgroundColor.a = 1, resultColor.a = foregroundColor.a + (1f - foregroundColor.a) = 1

Demos, Support, & Credits

Demos
- WebGL Demo:
https://www.tangledrealitystudios.com/games/ultimate-screenshot-tool-demo/

- Ultimate Screenshot Tool Overview:
https://youtu.be/bRHlAUvy5sg
- All Mobile Screenshots in One Button Press Demo:
https://youtu.be/53GTaAQDaNM
- Make Marketing Easy Demo:
https://youtu.be/LnBKHXD1Zlo

Support
- Email: jacob@tangledrealitystudios.com - best
- Twitter: @tangled_reality
- Discord: https://discordapp.com/invite/nFuptUZ
- Reddit: https://www.reddit.com/r/tangledreality

Credits
- Twity by toofusan
https://github.com/toofusan/Twity
- Native Gallery by yasirkula
https://github.com/yasirkula/UnityNativeGallery
- Native Share by yasirkula
https://github.com/yasirkula/UnityNativeShare
- Sound effects
https://freesound.org/people/Mafon2/sounds/275307/
https://freesound.org/people/Anthousai/sounds/337226/
https://freesound.org/people/rxrxhx/sounds/401252/

https://freesound.org/people/seedlife/sounds/371090/
https://freesound.org/people/mmaruska/sounds/167556/

- Mice Icons

https://www.goodfreephotos.com/vector-images/mouse-cursor-vector-art.png.php
https://pixabay.com/en/mouse-pointer-arrow-ps-computer-1345876/
https://commons.wikimedia.org/wiki/File:Font_Awesome_5_solid_mouse-pointer.svg