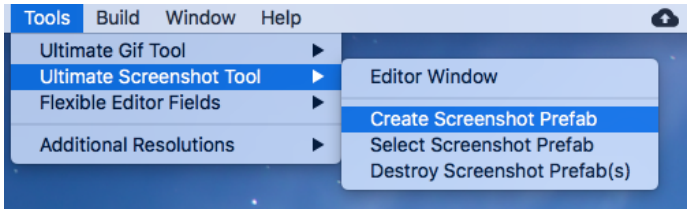
Ultimate Screenshot Tool

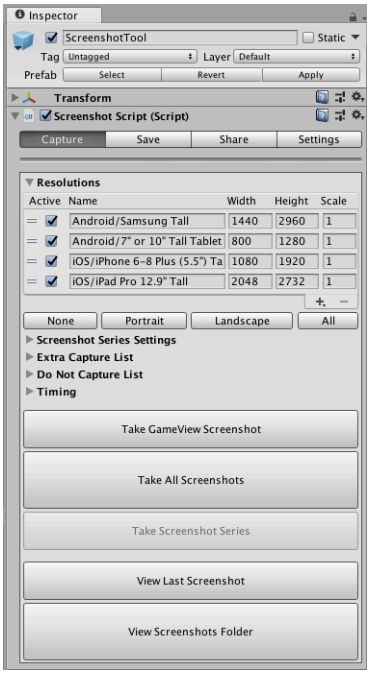
Tangled Reality Studios LLC

Getting Started



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

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 All Screenshots - Takes a screenshot for each resolution in the resolutions list



- View Last Screenshot - Open the last screenshot
- View Screenshot Folder - Open the screenshot folder

More Buttons:
Add ScreenshotSeriesScript to your ScreenshotTool:
- Take Screenshot Series - Capture multiple screens with a single press. Select the buttons to have the tool navigate to the proper screens

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

Add GameObjectScreenshotScript:
- Take GameObject Screenshots - Enable/disable game objects & take screenshots at different resolutions. Perfect for creating promotional shots/capturing item previews

Add MultiLangScreenshotScript:
- Take All Multi-Language Screenshots - Take a screenshot for each resolution in the resolution list for each language in the languages list
- Take Multi-Language Screenshot Series - Take a screenshot series as mentioned above for each language in the language list
Note: Set your preferred language last to return to it when the screenshots are finished.

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

Project Setup
Add
<provider
android:name="com.yasirkula.unity.UnitySSContentProvider"
android:authorities="MY_UNIQUE_AUTHORITY"
android:exported="false"
android:grantUriPermissions="true" />

to your AndroidManifest.xml between the <application> and <application/> tags. Replace MY_UNIQUE_AUTHORITY with a unique string (like your bundle identifier). This is very important as two apps with the same android:authorities string in their <provider> tag can't be installed on the same device.

If you do not have an AndroidManifest.xml:
1. Build for Android from File -> Build Settings -> Android -> Build
2. Open directory '[path/to/your-project]/Temp/gradleOut/src/main/'
3. Copy './AndroidManifest.xml' to '[path/to/your-project]/Assets/Plugins/Android/AndroidManifest.xml'
4. Make your customizations to that file

Alternatively, you may find your default AndroidManifest.xml in Temp/StagingArea/AndroidManifest.xml. Otherwise, you can grab the default Unity AndroidManifest.xml from C:\Program Files\Unity\Editor\Data\PlaybackEngines\androidplayer or Applications/Unity/PlaybackEngines/AndroidPlayer/Apk/ on mac

Share Keys
You'll need to set your own keys in Shared/Social Share/Config/CaptureToolKeys.cs

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, GameObjectScreenshotScript, MultiLangScreenshotScript, ShareScript, and UpdateShareWithScreenshotScript

Demos, Support, & Credits

- 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 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
- Screenshot 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

Features

- Capture all your app store resolutions in a single click!
- Take high resolution screenshots live during gameplay and in Editor!
- Make marketing easy with editor buttons to share your game everywhere!
- Share screenshots with Share Sheets, Imgur upload, or automatically generated urls!
- Quality support and documentation with quick contact buttons directly in the tool!

- Instantly share! Has #screenshotsaturday and #indiedevhour buttons that appear at exactly the right time and enable you to share to Twitter in 2-clicks!

- Take SceneView shots for dynamic alternate angles without breaking your cameras!
- Add your custom logos, hide UI, or show a build number by selecting GameObjects to automatically enable/disable during screenshots! Totally customizable!
- Show captured screenshots in game for cool gameplay moments!

- Save in JPG or PNG with selectable JPG quality setting
- 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)

- 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

- Hotkeys for instant capture during gameplay
- Cutouts for taking screenshots of a select portion of the screen
- Option to customize background during screenshot to be transparent or a solid color
- Gridlines to help you line up shots, but that disappear while you take them
- Automatically generated debug text to show your build information during screenshots
- Outline canvas to highlight part of the screen to share with other team members

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

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).

- 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
-- f takes a single screenshot
-- r takes screenshots at all resolutions
-- b takes a burst of screenshots (if BurstScreenshotScript is added to tool)
-- n takes a set of cutouts from the selected resolution (if CutoutScreenshotSetScript)
-- c toggles the cutout preview visibility (if cutout is active)
-- t toggles the gridline visibility (set in HotKeyActivationScript on "Gridlines Canvas")

- Hotkey for takes 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 lists work with cameras and entire canvases. This may be an easier/faster way to exclude them from your 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

Quick Fixes

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 will use the proper resolution.
- Resizing the resolution doesn't work on mobile. This is part of the system. The texture could be resized, but the typical screen size updates wouldn't go through.

Cameras/Canvases layering
- Using the same canvas sort orders/camera depths 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. Could use it to for a higher resolution & scale it down.
- 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 gallery to share file as gallery saves are not directly accessible by the app.
- "Can't find ContentProvider, share not possible!" - Add the provider code as described in the Getting Started section.
- If the "Share by Email" or "Email" support button 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 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 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

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.

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 gameobject 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.

- 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.

- 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
- 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
— Only shown if Screenshot Series Script is attached to a separate GameObject, otherwise, it is automatically set.
— The ScreenshotScript that should be used to take the screenshots.
- 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
— Only shown if Screenshot Series Script is attached to a separate GameObject, otherwise, it is automatically set.
— The ScreenshotScript that should be used to take the screenshots.
- 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.)

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.

Save Tab

Save Path
- The directory to save the screenshots in.
- 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.

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.

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.

Background Camera
- 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.

UI Camera
- Autoswitch Render Mode
- Only in RenderTexture mode.
- 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 mode.
— 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 mode.
— Toggles if the plane distance should be overwritten at all. Likely unnecessary if using the default UI camera.
- Plane Distance Override
— Only in RenderTexture mode.
— 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 mode.
— Refreshes the list of canvases that should by 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.

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