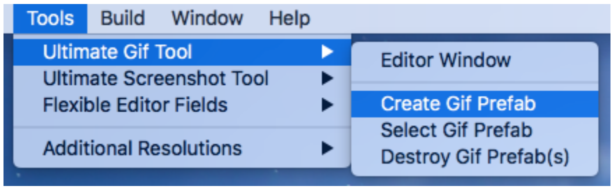
Ultimate Gif Tool

Tangled Reality Studios LLC

Table of Contents

Getting Started

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



Buttons

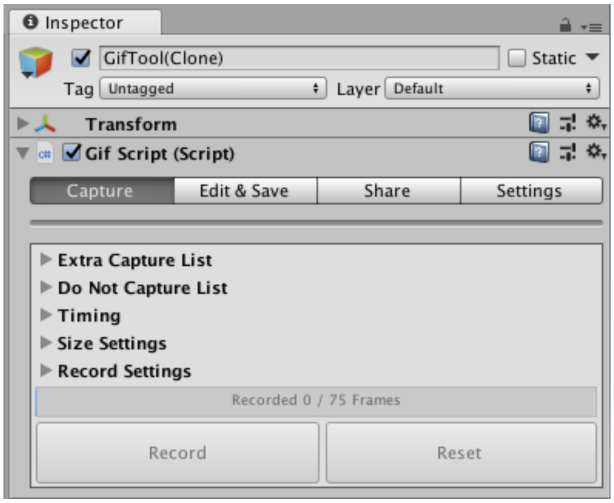
- Record/pause - Start/stop recording a gif

- Reset - Clear all frames and prepare to record a new gif

- Edit & Save - Open tab to choose frames and save

- Share - Open tab to share to social media or upload to your favorite site

- Settings - More custom settings to record the perfect gif as well as support contact information.



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://developers.giphy.com/docs/` for details on getting a Giphy API key. The rate-limited development key should be used without a username. See `https://developers.giphy.com/faq/` for details on getting a production key and conditions 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.

Tool Setup

Functionality may be added or removed from the GifTool by adding and removing components. Such components include: ShareScript and UpdateShareWithGifScript

Features

- Easily take gifs during gameplay!
- Make marketing easy with instant share buttons!
- Play captured gifs on textures in game for cool gameplay moments!
- Quality support, demo, and documentation!
- Full source code included!

- Loads of Recording Options:
- Record gifs with **transparency**, **reverse**, and/or **ping pong** mode for epic highlights!
- Playback gifs in different fps for **ultimate slow-mos** or **super speed craziness**!
- Record the **last few seconds** to always save that perfect moment or for **a set amount of time** when you know the action is about to start!

- Boost your productivity:
- Instant share buttons! #giffriday and #indiedevhour buttons that appear at exactly the right time and enable you to share to Twitter in 2-clicks!
- **Perfect for updating social media** with new builds.
- **Hotkeys** for quick capture during gameplay.

- Show gifs in game to highlight cool gameplay moments!
- Texture tools to create gradients, layer textures, cutout, resize, and solidify!
- **Loads of capture options:** layer textures into the background or foreground of your gifs, 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 gifs! Totally customizable!

- Edit your gifs after capture!
- View the last gif or the gif folder with the touch of a button!
- Option to customize gif background to be transparent or a solid color.
- Threaded when possible during gif saving and importing from save files.
- DontDestroyOnLoad option with features to automatically update the cameras when you switch scenes (during gameplay)
- Selectable quality settings

- 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 gifs)
- Events to allow programmers to create custom code to handle their own special cases
- Additional scripts such hotkey scripts, GameView controls, & various extensions!

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.

iOS Crashes and Memory Errors

- Keep in mind that iPhone 4s has a RAM of 512MB and some newer devices have a RAM of 1GB+. Crashes due to running out of memory are very possible.
- Resizing After Recording with Auto Prepare for Preview set will lower the resolution after each frame is captured to lessen the overall memory demands. Keeping the fps lower as well will also help. Half resolution with 15 fps worked on iPad 2 & iPhone 4s.
- Be sure copy before reset is disabled on your DisplayGifScript to avoid having 2 gifs in memory at once.

Transparency

- Gif files have a single level of transparency. They do not have an alpha channel and do not handle partial transparency. The only way to create transparency is to replace a color with 100% transparency. For that reason, ensure your transparency color is not a color used in your game.
- For best ease of use and efficiency, UGT replaces pixels with an alpha value of 0 with the transparency color when saving. This means that to create a transparent gif you may use transparent pixels or the transparency color. Using transparent pixels will create transparent previews and save files and look better during capture.

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 gif, check that your overlay camera's culling mask includes the UI flag.
- If your UI is distorted in Single Camera or MultiCamera 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 gif prefab.

Sharing
- Gif 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
- ShowDuringGifScript/HideDuringGifScript only work in play mode.
- Async capture mode seems to only work with a single camera in the scene. Hopefully it will be updated in future versions of Unity.
- If you're getting errors related to Metal in your Editor on Mac, open Unity with the metal flag using the terminal command: /Applications/Unity/Unity.app/Contents/MacOS/Unity -force-metal
- If "View Last Gif" or "View Gif 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.

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!

- Only the textures of the current gif are cached in GifScript. If you need to persist a texture longer, copy the texture first. See UltimateGifTool/Scripts/DisplayGifScript.cs.

- Use Hotkeys - change in settings
-- g toggles recording the gif (start recording or pause)
-- y records a single gif frame
-- x saves the gif
-- h resets the gif
-- t toggles the gridline visibility (set in HotKeyActivationScript on "Gridlines Canvas”)

- Creating gif folder within project is tidy, but can cause longer build times. Hide the gif 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`
- Gifs may also be ignored by git by adding the gif folder to your .gitignore

- Extra capture list/do not capture list are temporarily enabled/disabled objects that automatically appear/disappear during gifs. Afterwards they return to their initial state.

For example, the "Gridlines Canvas" is set to automatically disable during gifs. If enabled, it would be hidden during gifs. It is disabled by default.
Similarly, the "Debug Text Canvas" can be set to automatically appear during gifs if 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. It may be an easier/faster way to exclude them from your gifs depending on your build.

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

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

Programming API

GifScript

Most of the relevant code for using this tool will be accessed through the GifScript. Some additional functionality can be found in the add-on scripts such as the ShareScript or other scripts attached to that GameObject.

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

```
    public static System.Action<GifScript, RecorderState> RecorderStateChanged;

    public static System.Action<GifScript, int, int> WillTakeGif;
    public static System.Action<GifScript, Texture2D[], int> GifTaken;
    public static System.Action<GifScript, Texture2D[], int> GifToSave;
    public static System.Action<GifScript, float> GifSaveProgress;
    public static System.Action<GifScript, string> GifSaved;

    public static System.Action<GifScript> WillResetGif;
    public static System.Action<GifScript> GifReset;

    public static System.Action<int, float> WorkerGifSaveProgress;
    public static System.Action<int, string> WorkerSavedGif;
```

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

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

The public methods are:

```
public void ToggleRecording()
public void Record()
public void RecordOneFrame()
public void Pause()

public void Save()
public void Save(string fullFilePath = “”, int firstFrame = 0, int lastFrame = -1, bool temp = false)
public void Reset()

public void PrepareForPreview()
public Texture2D[] GetFrames(int firstFrame = 0, int lastFrame = -1)
public void RemoveFrames(int firstFrame, int lastFrame)
public void RemoveFramesRange(int firstFrame, int count)
public void RemoveFrame(int frameIndex)
public void RequestFrames(int firstFrame = 0, int lastFrame = -1)
```

When saving, the default file path will be used if an empty string is passed.
You can find the default file path with (reference to the GifScript).fileSettings.FullFilePath().

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 GifSunday Button under GifTool -> Reminder Canvas -> GifSunday 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 gifs (ShowDuringGifScript and HideDuringGifScript). Using Extra Capture List/Do Not Capture List may be simpler though.

Other useful scripts can be found in Demo -> Scripts, Shared -> Helpful Extras, Shared -> Scripts, Shared -> Social Share -> Demo, and Shared -> Social Share -> Scripts.

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:

```
(texture reference).AlphaBlend( (template texture reference), true);
```

That could be useful for adding in logos, overlays, etc. that you do not want to temporarily appear on screen (like with an Extra Capture List item for example).

Full Overview

GIF Tool

Capture Tab

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

Extra Capture List/Do Not Capture List
- Enable/Disable certain GameObjects during your gifs. Super useful for hiding debug info or automatically adding your logo to all your gifs.
- Do Not Capture list contains “Gridline Canvas” by default. Enable it for a demo.

Timing
- Delay Before Capture
— Delay in seconds between when button is pressed and when gif 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.

Size Settings
- Resize Before Recording
— Option to resize screen before recording.
— Resizing before recording results in higher quality output and speed of capture than resizing after recording. Can also be used to change the aspect ratio.
— Resizing screen does not work on mobile.
- Resize To Resolution/Scale button
— Toggle between resizing to a specific resolution or resizing to a scale of the original resolution. Resizing screen allows the screen to adjust, so aspect ratio is not fixed.
- Cutout
— Cutout a portion of the screen for your gif. Useful if you want to crop something out or just capture a part of the screen.
— Must be active to be in use.
— Preview overlay will automatically be hidden during gifs.
- Resize After Recording
— Option to resize the frames after recording.
— Smaller gifs save memory.
— Resizing screen does work on mobile.
- Resize To Resolution/Scale button

— Toggle between resizing to a specific resolution or resizing to a scale of the original resolution. The aspect ratio is fixed to avoid distortion.

Record Settings

- Auto Prepare for Preview

— Fully prepares each frame when it is captured, so that it can be immediately previewed. Resizing after capture can be useful to save memory.

- Prepare in Editor

— Only appears if Auto Prepare for Preview is not set.

— Prepares frames only in the editor. Useful for quick recording in editor.

- Adjust Time for Capture

— Adjust frame rate for optimal capture. This will slow the frame rate to allow better processing and capture. Depending on the frame rate it may make gameplay difficult.

— Recommended for capturing gifs in editor, but not during gameplay.

- Override 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.

- Use Transparency

— Option to use transparency within the gif. See Transparency section in Quick Fixes for more details. In short, since gifs do not have an alpha channel select a unique color to be used as transparent. Avoid using a color that already appears in game.

— Defaults to a pink color that seems to be uncommon among games.

- Stop Automatically

— Whether the gif recording should stop automatically once the desired duration is reached or keep recording always keeping the last {duration} seconds

- Duration

— Max length of the gif in seconds. Can still be paused/saved early or trimmed down.

- Frames Per Second

— The number of frames to record per second. The higher the number the smoother the gif, but the larger the resulting file size.

- Different Playback Speed

— Set to select a playback speed that is different from the capture rate.

— Useful to capture ultra-slow mo or super speed gifs. For example, with Adjust Time for Capture set, you could record 600 fps and playback at a slower speed like 15 fps.

Edit & Save Tab

Preview

— Only appears after frames have been recorded

— The play button will play the preview at the playback speed. The buttons on either side will trim the start/end of the gif to he current frame. The frames may also be selected in the two-sided range or int fields. They are zero-indexed.

Edit

- Edit Saved Gif - load a saved gif into memory to edit it.
- Edit Transformations - transformations to apply to the preview gif.

- Apply Transformations - transformations are applied on a copy; the original is not lost.
- Restore - undo changes to the gif.
- Save - save the gif.

Save Settings
- Save Path
— The directory to save the gifs 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 the gifs in within Standalone builds. The entered values are relative to the data path.
— The default values should out the folder at the same hierarchy level as the game.

- Mobile Save Settings
— The settings on if/where to save the gifs within Mobile builds.
— Save to Gallery will save the gifs 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 gifs 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 yourself when no longer useful.

- WebGL Save Settings
— The settings on if/where to save the gifs within WebGL builds.
— Open In New Tab will open the gif(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 gif 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 gifs 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 yourself when no longer useful.

- 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 the tabs to enable/disable a component of the file name (as demonstrated in the 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 gif. (Multiple gifs taken at the same time will have the same count.)

— Reset Counter will return the counter to 0.

- Worker Priority
— The priority of the background thread used to save the gif. Higher priority results in faster save time, but may slow down the rest of the game.
- Ping Pong Mode
— Enable to cause your gif to loop back from the last frame to the first. The gif will ping pong back and forth in a perfect loop. 1st frame -> Last frame -> first frame.
- Reverse Mode
— Enable to reverse your gif. It will go from the last frame to the first. Last frame -> first frame. Works with Ping Pong Mode, but may not be very noticeable with a loop.
- Infinite Loop
— Enable to cause your loop to automatically loop forever. Disable to choose a specific repeat count or to not loop at all.
- Repeat Count
— Only appears if infinite loop is disabled.
— Specify a specific number of times the gif should loop. 0 results in no repeats.
- Quality
— Higher quality produces better colors, but slows down processing time.
- Reset After Save
— Set to automatically reset the GifScript after saving.
— Useful to save an extra button press if capturing many different gifs in a row.
— Leaving it disabled allows recording disjointed gifs (recording, pausing, and then recording again) or saving the gifs in multiple different formats.
- Estimate File Save
— Estimates the size of the file by generating a temporary file. Save will preserve it.

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 UpdateShareWithGifScript is attached to the tool, the last gif 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.

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.)
- Single Camera grabs the render texture of a single camera.
- MultiCamera grabs the render texture of each camera and layers them together. Useful for specific custom cases like Case 3 in the transparency section.
- Async asynchronously grabs the texture of a single camera. It feature relies on an experimental Unity feature. Currently it doesn't handle multiple cameras in the scene.

Cameras
- Only appears if not in Screen Capture mode.
- Specify the cameras that should be captured in the gif.

HotKeys
- Toggle recording your gif with the press of a single key.
- Customizable. Set each action to whatever key (or key combination) you'd like.
- Great for taking gifs right in the middle of the action.

Audio Settings
- Gif Record and Gif Pause Sound
— Audio source containing the camera shutter sound.
- Play in Editor
— Toggle if the sound should play when taking recording/pausing gifs in editor.
- Play in Game

— Toggle if the sound should play when taking recording/pausing gifs in game.

Mouse Settings
Allows you to capture the mouse in gifs. The in game mouse will be visible.
- Show Original Mouse
— Whether or not the device mouse should 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).

UI Camera
- AutoSwitch Render Mode
— Only appears if not in Screen Capture mode.
— Canvases in overlay render mode will not be captured by any individual camera, so would normally be left out of gifs. This feature temporarily switches those canvases to camera render mode during gifs, so your UI will appear.
- UI Camera
— Only appears if not in Screen Capture 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 appears if not in Screen Capture 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 appears if not in Screen Capture mode.
— Refreshes the list of canvases that should by automatically switched from overlay mode to canvas mode during gifs.
— Not done automatically to avoid frequent costly loops through the object hierarchy.
— It may be triggered automatically by other events.

- ResizeDuringFrameCapture
— Only appears in Single Camera and Async mode.
— Resizing each frame can be done when the frame is captured to save memory or when saving to decrease the amount of processing done on during capture.

— Resizing must be done before a preview is available.
- ResizeInEditor
— Only appears in Single Camera and Async mode & if not ResizeDuringFrameCapture
— Enable resizing in the editor so the preview is automatically updated.
— Same as ResizeDuringFrameCapture, but this option allows to disable ResizeDuringFrameCapture for your builds while still keeping the feature in editor
— If both are disabled, a button to manually update preview will appear under Record/Reset and must be used before the preview is displayed.

DontDestroyOnLoad Settings
- DontDestroyOnLoad
— Sets the GifTool GameObject to DontDestroyOnLoad. This allows you to set-up the GifTool in a single scene & continue to use it even when scene switches.
— DontDestroyOnLoad only preserves the GameObject between scene transitions. It will not create the GifTool that does not already exist.
— You could safely disable this setting and create separate GifTools in different scenes, so you can customize the behavior to the scene. Alternatively, you could create your own script to modify the settings on scene changes.
- 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) 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

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

- Ultimate Gif Tool Overview:
https://www.youtube.com/watch?v=cBtEO1OlMvo

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

- Credits:
- Dispatcher from UnityToolbag by Nick Gravelyn:
https://github.com/nickgravelyn/UnityToolbag
- Moments by Thomas Hourdel
https://github.com/Chman/Moments
- UniGif by WestHillApps (Hironari Nishioka)
https://github.com/WestHillApps/UniGif
- 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
- TextureScale.cs by Eric Haines(Eric5h5)
http://wiki.unity3d.com/index.php/TextureScale

- 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