

Readme - Pro GIF 1.51

Pro GIF is a mega-package of GIF, provides so many features about GIF. Targeted to deal with every need on GIF image processing. And, many useful stuff included. Meanwhile, it is light and easy to use, save time!

Features

<Core>

- Record GIF.
- Play GIF(support transparency).
- Load GIF file from local-path/url, decode, and playback. With option for saving file in local storage.Except for UI images, also support display gif with renderers.
- Rich settings: **FPS, Duration, Quality, RepeatCount, Aspect Ratio, Resolution**(support auto resize to fit any screen size).
- The encoding process runs in thread for better performance.

<Advanced>

- Advanced decode settings, allows setting how many frames to decode.
- Faster ProGif decoder.
- Supports multiple gif decodes and playback.
- Support record GIF with multiple cameras simultaneously.
- Convert JPG/PNG/Texture2D (List) to GIF.
- Crop GIF with a specified aspect ratio(16:9, 3:2, 4:3, 1:1, etc.) input in SetRecordSettings before StartRecord.
- Except for UI images, also support display GIF with renderers, i.e. cube, sphere, capsule, cylinder, quad, complex mesh...

<Extra>

- Complete Giphy API for accessing the world's largest GIF library.
- Use your own Giphy channels & Giphy API.
- Share on up-to 15 social platforms.
- Json tools(Newtonsoft.Json), work on mobile & desktop.
- OnEditorGifRecorder for recording gif in the Editor play mode. Record the development screens at any time.
- Some more stuff.

Support Editor, Android, iOS, Windows, Mac, Linux

Reminders

Build iOS: **.NET 2.0** is required for Newtonsoft.Json to work properly on iOS, please select **.NET 2.0** in Player Settings before building XCode project (**File > Build Settings > Player Settings > Other Settings > Optimization > Api Compatibility Level**).

———— PGif : Multiple Recorders ————

Recommended to use **PGif** if you have multiple cameras for recording GIFs.

The easiest way to record GIFs with multiple cameras. It is very simple like:

- `PGif.iStartRecord(camera, "RecorderName");`

Just specify a different name for each recorder on the target camera.

Use that name to access and control the recorder:

- `PGif.iPauseRecord("RecorderName");`
- `PGif.iResumeRecord("RecorderName");`
- `PGif.iStopRecord("RecorderName");`
- `PGif.iSaveRecord("RecorderName");`
- `PGif.iClearRecord("RecorderName");`

Also with rich settings for every recorder(encoder), just call this method before start the recorder:

- `PGif.iSetRecordSettings(.....);`

———— PGif : Multiple Players ————

Recommended to use **PGif** if you need to display multiple GIFs at a time.

You can also play multiple GIFs at a time using PGif:

- `PGif.iPlayGif(gifPathOrUrl, targetImage, "PlayerName");`

Just specify a different name for each player on the target image.

Use that name to access and control the player:

- `PGif.iPausePlayer("PlayerName");`
- `PGif.iResumePlayer("PlayerName");`
- `PGif.iStopPlayer("PlayerName");`
- `PGif.iClearPlayer("PlayerName");`

Also with rich settings for every player(decoder), just call this method before start the player:

- `PGif.iSetPlayerSettings(ProGifPlayerComponent.Decoder.ProGif,
ProGifPlayerComponent.DecodeMode.Advanced,
MaxNumOfFramesToDecode);`

Two decoder options(UniGif, ProGif), the ProGif decoder option supports decode multiple GIFs at the same time.

Two decode modes(Normal, Advanced), the Normal decode mode decode the entire GIF normally, the Advanced decode mode allows setting the number of frames to decode.

The ProGif decoder is faster.

More parameters and functions available in PGif class. Well commented.

———— OnEditorGifRecorder ————

The OnEditorGifRecorder is an editor script with prefab, for recording gif in the Unity editor play mode. Useful to record the development screens at any time.

How to use? Just drop the prefab(**OnEditorGifRecorder.prefab**) in the scene. Click the buttons in the inspector to record gif.

———— ProGifManager : Recorder ————

To setup and start

Get/Create an instance for ProGifManager:

- `ProGifManager gifMgr = ProGifManager.Instance;`

Call the methods like this:

- `gifMgr.Method(...);`

or

- `ProGifManager.Instance.Method(...);`

To make changes to the recorder settings like:

- `ProGifManager.Instance.SetRecordSettings(bool: autoAspect, int: width, int: height, float: duration, int: fps, int: repeatCount, int: quality);`

Or

- `ProGifManager.Instance.SetRecordSettings(Vector2: aspectRatio, int: width, int: height, float: duration, int: fps, int: repeatCount, int: quality);`

Start gif recording:

- `ProGifManager.Instance.StartRecord();`

Or

- `ProGifManager.Instance.StartRecord(Action<float>: onRecordProgressCallback, Action: onRecordDurationMaxCallback);`

Start gif recording with camera:

- `ProGifManager.Instance.StartRecord(Camera: camera, Action<float>: onRecordProgressCallback, Action: onRecordDurationMaxCallback);`

To pause

Pause gif recording:

- `ProGifManager.Instance.PauseRecord();`

To resume

Resume gif recording:

- [ProGifManager.Instance.ResumeRecord\(\);](#)

To stop

Stop gif recording, cannot be resumed, waiting to be saved/cleared:

- [ProGifManager.Instance.StopRecord\(\);](#)

To save stored frames to a gif file

- [ProGifManager.Instance.SaveRecord\(\);](#)

Or

- [ProGifManager.Instance.SaveRecord\(Action:
onRecorderPreProcessingDoneCallback, Action<int, float>:
onFileSaveProgressCallback, Action<int, string>: onFileSavedCallback\);](#)

To stop and save stored frames to a gif file

Stop and save the recording:

- [ProGifManager.Instance.StopAndSaveRecord\(\);](#)

Or

- [ProGifManager.Instance.StopAndSaveRecord\(Action:
onRecorderPreProcessingDoneCallback, Action<int, float>:
onFileSaveProgressCallback, Action<int, string>: onFileSavedCallback\);](#)

———— ProGifManager : Player ————

To play gif after recording complete

Play the recorded gif frames stored in recorder:

- [ProGifManager.Instance.PlayGif\(Image: targetImage, Action<float>:
onLoading\);](#)

To play gif with filePath or Url

Load and decode a gif file and play it:

- [ProGifManager.Instance.PlayGif\(string: gifPath, Image: targetImage,
Action<float>: onLoading, bool: shouldSaveFromWeb\);](#)

To pause / resume / stop gif player when a gif is playing

Pause the player, the player will be paused at current frame:

- [ProGifManager.Instance.PausePlayer\(\);](#)

Resume the player, continue to play from current frame:

- [ProGifManager.Instance.ResumePlayer\(\);](#)

Stop the player, the player will be stopped and reset to first frame:

- [ProGifManager.Instance.StopPlayer\(\);](#)

———— ProGifManager : Clean Up Memory ————

Clear recorder and player:

- `ProGifManager.Instance.Clear();`

Or

Clear recorder and player separately:

- `ProGifManager.Instance.ClearRecorder();`
- `ProGifManager.Instance.ClearPlayer();`

More parameters and functions available in ProGifManager class. Well commented.

———— JPG/PNG/Texture2D to GIF ————

Use ProGifTexture2DsToGIF to load images(JPG, PNG) from specific directory within your application.

Load the images as Texture2D and convert them to GIF with rich settings.

It is very simple. Just prepare your textures in a list.

Then create an ProGifTexture2DsToGIF object to do the magic for you!

- `ProGifTexture2DsToGIF.Instance.Save(...);`

You can prepare your texture list yourself, or use the LoadImages method to load images from a specific directory:

(JPG and PNG are tested)

- `ProGifTexture2DsToGIF.Instance.LoadImages(...);`

You may want to set the image format to load into the texture list before LoadImages:

- `ProGifTexture2DsToGIF.Instance.SetFileExtension(...);`

———— Giphy API ————

To use Giphy API, it requires a Giphy account to create API KEY and Upload API Key. You need to request a production key for the Upload API as well.

APPLY HERE: <https://developers.giphy.com/dashboard>

Simply Create/Get the instance of GiphyManager with 1 line:

- `GiphyManager giphyMgr = GiphyManager.Instance;`

Call the methods like this:

- `giphyMgr.Method(...);`

or

- `GiphyManager.Instance.Method(...);`

Set your channel UserName and Keys:

- `GiphyManager.Instance.SetChannelAuthentication(string: userName, string: apiKey, string: uploadApiKey);`

—— [Upload API] ——

Upload GIF:

- `GiphyManager.Instance.Upload(string: filePath, Action<GiphyUpload.Response>: onComplete, Action<float>: onProgress = null);`

Upload with Tags (helps describe the GIF and allows it to be found again by browsing or searching):

- `GiphyManager.Instance.Upload(string: filePath, List<string>: tags, Action<GiphyUpload.Response>: onComplete, Action<float>: onProgress = null);`

—— [GIF API] ——

Get GIF By Id:

- `GiphyManager.Instance.GetById(string: giphyGifId, Action<GiphyGetById.Response>: onComplete);`

Get GIFs By Ids:

- `GiphyManager.Instance.GetByIds(List<string>: giphyGifIds, Action<GiphyGetByIds.Response>: onComplete);`

Search GIFs (for words or phrases):

- `GiphyManager.Instance.Search(List<string>: keyWords, Action<GiphySearch.Response>: onComplete);`

Random GIF (get a gif by random):

- `GiphyManager.Instance.Random(Action<GiphyRandom.Response>: onComplete);`

Random GIF (get a random GIF limited by tag):

- `GiphyManager.Instance.Random(string: tag, Action<GiphyRandom.Response>: onComplete);`

Translate GIF (translating word and phrase to GIF):

- `GiphyManager.Instance.Translate(string: term, Action<GiphyTranslate.Response>: onComplete);`

Trending GIFs (hot GIFs hand curated by the Giphy editorial team):

- `GiphyManager.Instance.Trending(Action<GiphyTrending.Response>: onComplete);`

—— [GIF Sticker API] ——

Search Stickers (for words or phrases):

- `GiphyManager.Instance.Search_Sticker(List<string>: keyWords, Action<GiphyStickerSearch.Response>: onComplete);`

Random Sticker (get a random GIF sticker):

- `GiphyManager.Instance.Random_Sticker(Action<GiphyStickerRandom.Response>: onComplete);`

Random Sticker (get a random GIF sticker limited by tag):

- `GiphyManager.Instance.Random_Sticker(string: tag, Action<GiphyStickerRandom.Response>: onComplete);`

Translate GIF Stickers (translating word and phrase to GIF sticker):

- `GiphyManager.Instance.Translate_Sticker(string: term, Action<GiphyStickerTranslate.Response>: onComplete);`

Trending GIF Stickers (hot GIFs hand curated by the Giphy editorial team):

- `GiphyManager.Instance.Trending_Sticker(Action<GiphyStickerTrending.Response>: onComplete);`

———— Social Share ————

Share GIF/image Url(s) return by the Giphy APIs. GIF preview and playback depends on the social platform. Support up to 15 social platforms (Facebook, Twitter, Tumblr, VK, Pinterest, LinkedIn, Odnoklassniki, Reddit, GooglePlus, QQZone, Weibo, Baidu, MySpace, LineMe, Skype).

Share GIF and/or text message:

- `GifSocialShare gifShare = new GifSocialShare();`
- `gifShare.ShareTo(Social: socialPlatformType, string: title, string: description, string: url1, string: url2);`

———— Demo Scenes ————

SimpleStartRecordDemo.unity

This scene shows the simplest steps to start, change settings and stop/save a recording for a game.

ProGifDemo_Panels_HideUI.unity

This scene shows the steps of record, playback and change settings with our UI templates. Pay attention to the canvas setting that let this example exclude UI from camera.

ProGifDemo_Panels_ShowUI.unity

This scene shows the steps of record, playback and change settings with our UI templates. Pay attention to the canvas setting that let this example include UI from camera.

ProGifDemo_SpecificCamera.unity

This scene demonstrates the ability to record GIF with specific camera.

GiphyApiDemo_GetById.unity

This scene shows how to get GIF from [Giphy.com](https://giphy.com) by id. And, share the GIF with its Bitly_Url/Url/Id responded by Giphy. You can test max 15 social platforms here.

ProGifDemo_MultipleCamera.unity

This scene demonstrates how to record GIFs with multiple cameras using different GIF settings.

Texture2DToGIFDemo.unity

This scene demonstrates how to load and convert images(JPG, PNG > Texture2D) to GIF.

ProGifDemo_PlayerRenderer.unity

This scene demonstrates displaying gif with different renderers(i.e. cube, sphere, capsule, cylinder, plane) at a time.

GiphyApi+ProGifPlayer Demo.unity

This scene demonstrates the combined use case of Giphy API with Pro GIF. Download and display multiple gifs at a time.

Thank you for buying this package!

Any problem and bug report please contact us at swan.ob2@gmail.com. Remember to rate this asset on the AssetStore if you like it. Your review is always appreciated!

Download/Purchase on the Asset Store Now!!

Visit our asset page for more information.

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

SwanDEV