Readme - Pro GIF Recorder Toolkit 2.0.2

Pro GIF Recorder Toolkit is a powerful GIF recorder package for Unity, which supports multiple mobile and desktop platforms, with the unified APIs. Same with Pro GIF & Pro GIF Universal, the encoder is multi-threaded, brings the best performance GIF recorder for your indie and even commercial project!

This package provides the features to record animated GIFs / GIF Replay. Advanced, powerful GIF recorder, highly customizable settings(e.g. rotate, resolution, crop, transparency, fps, quality, memory usage optimization...). Much more useful stuff included that makes this package not only a recorder but also a toolkit for making & handling GIFs for games & applications!

All our code and examples are carefully designed to provide a clean, easy-to-use package. The Unified APIs allowing you to integrate once, runs on all supported platforms. GIF has never been so easy with Pro GIF series!

Highlights

- The Ultimate GIF recording solution and highly optimized encoder.
- Super fast, multi-threaded encoder.
- Instant preview, Ultra-low memory footprint.
- Save Reverse and Ping-pong play mode GIF.
- Advanced recorder features(Crop, Rotate, Transparent background GIF, Multiple recorder instance).

Index

(1) Features

<Core>

- Record GIF/GIF Replay (support transparent).
- Instant Preview/Play the newly recorded GIF.
- Rich settings: FPS, Duration, Quality, RepeatCount, Aspect Ratio, Resolution(support auto resize to fit any screen size), Transparent Color(for hiding a particular BG color).
- The encoding process runs in threads for better performance.

<Advanced>

- Support record GIF with multiple cameras.
- Super fast, multi-threaded encoder.
- Ultra-low (preview player) memory footprint, even for large number of GIF frames.
- Crop GIF (with a specific aspect ratio, e.g. 16:9, 3:2, 4:3, 1:1, etc.).
- Rotate GIF (90, -90, 180 degrees).
- Support save Reverse and Ping-pong play mode GIF.
- Support adding Comment-Extension(human readable metadata, e.g. image credit, description).
- Preview newly created GIF on Image, RawImage, Renderers(Meshes such as Cube, Plane, Sphere etc...), GuiTexture and any other material that supports Texture2D, Sprite, or RenderTexture.

<Extra>

- GIF API helper classes for uploading and getting uploaded GIFs from Giphy.
- Use your own GIF channels & API keys.
- Share on up-to 15 social platforms.
- Optimized Json tool(Newtonsoft.Json), work on mobile & desktop.
- The codeless Pro GIF recorder (editor extension) for recording GIF in both the Editor play mode and runtime App, without modifying your code, just drag and drop the script/prefab to your scene.
- Some more useful stuff.
- GIF libraries full source code.

Support: Android, iOS, Windows, Mac, Linux, Unity Editor.

(2) Reminders, Setup & Requirement

Build iOS: **.NET 2.0 or newer** is required for Newtonsoft.Json to work properly on iOS.

* Newtonsoft.Json is used with the API helper classes(i.e. Giphy API) in this asset. It will not be included in your build if you don't use the API.

Reminded, for newer Unity versions that Newtonsoft plugin is included by default, you can delete our Newtonsoft plugin (entire folder) at: ***Assets/SWAN Dev/Newtonsoft***

* The GIF library(recorder/encoder) can be used independently and compatible with .NET 2.0 Subset too. (.NET 2.0 Subset and newer compatible)

Setup for Scriptable Render Pipeline (SRP) : URP/LWRP/HDRP

If your project is using Scriptable Render Pipeline (e.g. URP/LWRP/HDRP), in order to let the recorder to record the frames you have to insert the define symbol **PRO_GIF_SRP** in the Unity Editor(***File > Build Settings > Player Settings > Other Settings > Scripting Define Symbols***).

GUITexture (obsolete)

This component is obsolete, so we adds a define symbol for it: **PRO_GIF_GUITEXTURE**
All the GUITexture related scripts in this asset will be hidden by default. But you can re-enable them by insert the define symbol in the Unity Editor(***File > Build Settings > Player Settings > Other Settings > Scripting Define Symbols***).

MobileMedia Plugin

The MobileMedia Plugin provides your apps the ability to save and pick media files to/from the Android and iOS device Gallery/Photos. Once you have the plugin installed, you may add the define symbol "**SDEV_MobileMedia**" in the Unity ***PlayerSettings > OtherSettings > Scripting Define Symbols***.
This define symbol enables the MobileMedia features for the CodelessProGifRecorder editor extension and certain Pro GIF demo scenes.
Check it out: https://www.swanob2.com/mobile-media-plugin

(3) PGif : Multiple Recorders (Encoder)

The **PGif** manager is recommended if you have multiple cameras for recording GIFs in the scene. The easiest way to record GIFs with multiple cameras. It is very simple as below:

Record multiple GIFs using PGif:
PGif.iStartRecord(Camera:camera, string:RecorderName, ...);

* For multiple recorder use case, just specify a unique name for each recorder.

Use that unique name to access and control the recorder(s):
PGif.iPauseRecord(string:RecorderName);
PGif.iResumeRecord(string:RecorderName);
PGif.iStopRecord(string:RecorderName);
PGif.iSaveRecord(string:RecorderName, string:optionalGifFilename);
PGif.iClearRecord(string:RecorderName);

Customize settings for recorder(encoder), call the below method before recorder start:
PGif.iSetRecordSettings(bool:autoAspect, int:width, int:height,
float:duration, int:fps, int:loop, int:quality);

PGif.iSetRecordSettings(Vector2:aspectRatio, int:width, int:height,
float:duration, int:fps, int:loop, int:quality);

Set the GIF rotation
PGif.iSetGifRotation(ImageRotator.Rotation:rotation);

Set the GIF transparent color (hide this color in the GIF)
PGif.iSetTransparent(Color32:color, byte colorRange);

Set the GIF play mode before Save (Normal, Reverse, Ping-pong)
PGif.iGetRecorder(string: recorderName).recorderCom.m_EncodePlayMode =
ProGifRecorderComponent.EncodePlayMode.Reverse;

Callbacks

The below callbacks are handled in the iStartRecord method, please register these callbacks if need, by assigning your methods/Actions to receive updates from them. Confuse? Don't worry, there are plenty of demo scenes included, we will show you step by step in the scenes.

Action<float>: onRecordProgress
Action: onRecordDurationMax
Action: onPreProcessingDone
Action<int, float>: onFileSaveProgress
Action<int, string>: onFileSaved

More parameters and methods available in PGif class or through the iGetRecorder and iGetPlayer method in the class.

(4) PGif : Multiple Players (GIF Preview)

The ProGif preview player support playback of multiple GIF recorder sources at the same time (in case you have set up multiple recorders). Also support Optimize Memory Usage option to minimize the memory consumption.

The **PGif** manager is recommended for displaying multiple newly created GIFs at a time.

Play multiple GIFs using PGif:
PGif.iPlayGif(ProGifRecorder:recorderSource, Image:playerImage, string:playerName,
Action<float>:onLoading);

* For the display target, supports Image, RawImage, Renderers(Meshes such as Plane, Cube, Sphere, etc, and GuiTexture).
* For multiple player use case, just specify a unique name for each player.

Use that name to access and control the player(s):
PGif.iPausePlayer(string:PlayerName);
PGif.iResumePlayer(string:PlayerName);
PGif.iStopPlayer(string:PlayerName);
PGif.iClearPlayer(string:PlayerName);

Set the flag to enable/disable Memory Usage Optimization:
PGif.iSetPlayerOptimization(bool:enable);

Set Ping-pong play mode:
PGif.iGetPlayer(string:PlayerName).PingPong();

Set Reverse play mode:
PGif.iGetPlayer(string:PlayerName).Reverse();

Callbacks

Reports the loading progress, instantly finish if play GIF using the recorder source.
PGif.iGetPlayer(string:PlayerName).SetLoadingCallback(Action<float>:onLoading);

The callback to be fired on every frame during play GIF. Pass a GifTexture each time.
PGif.iGetPlayer(string:PlayerName).SetOnPlayingCallback(
Action<GifTexture>:onPlaying);

More parameters and methods available in PGif class or through the iGetRecorder/iGetPlayer method in the class.

(5) PGif : Clean Up Memory

The GIF Manager handles memory clean up when a recorder or player restart, but in case you want to implement the Clear methods manually. You can use the below methods:

Clear the target recorder by recorder name:
PGif.iClearRecorder(string:recorderName);

Clear the target recorder and ensure the textures not being cleared too early(for the case the recorder source is in use by a gif player):
PGif.iClearRecord_Delay(string:RecorderName, string:playerName, Action<string>:onClear);

Clear the target player by player name:
PGif.iClearPlayer(string:playerName);

(6) CodelessProGifRecorder

This is a code-less GIF recording feature designed for recording GIFs without changing your codes. It provides a convenient way for recording GIFs in your app/game, with some simple setup in the scene. So you can create GIFs at any time when you need to.

How to use?

Use it in the Editor:
- Drop the prefab(**CodelessProGifRecorder**.prefab) to your scene,
- Run the scene, make some setting changes in the inspector if need,
- Click the 'Start Record' button to start the GIF recorder,
- Click the 'Save Record' button to save the stored frames as GIF.
- Wait for the save progress to finish.

Use it in your App at runtime:
- You can also reference the methods and dynamic parameters in the CodelessProGifRecorder to your UI components like Button, Slider, InputField, and Toggle, etc. in the scene, this allows you to record GIFs at runtime in your app.

(7) ProGifManager : Recorder (Encoder)

To setup and start
Get/Create an instance for ProGifManager:
ProGifManager gifMgr = ProGifManager.Instance;

Call the methods like this:
gifMgr.MethodName(…);
or
ProGifManager.Instance.MethodName(…);

To make changes to the recorder settings:
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 (Camera.main will be used):
ProGifManager.Instance.StartRecord();
Or
ProGifManager.Instance.StartRecord(Action<float>:onRecordProgress,
Action:onRecordDurationMax);

Start gif recording with specific camera:
ProGifManager.Instance.StartRecord(Camera:camera,
Action<float>:onRecordProgress, Action:onRecordDurationMax);

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(string:optionalGifFilename);
Or
ProGifManager.Instance.SaveRecord(Action: onRecorderPreProcessingDone,
Action<int, float>:onFileSaveProgress, Action<int, string>:onFileSaved,
string:optionalGifFilename);

To stop and save stored frames to a gif file
Stop and save the recording:
ProGifManager.Instance.StopAndSaveRecord(string:optionalGifFilename);

Or
ProGifManager.Instance.StopAndSaveRecord(Action:onRecorderPreProcessingDone,
Action<int, float>:onFileSaveProgress, Action<int, string>:onFileSaved,
string:optionalGifFilename);

Set the GIF rotation
ProGifManager.Instance.SetGifRotation(ImageRotator.Rotation:rotation);

Set the GIF transparent color (hide this color in the GIF)
ProGifManager.Instance.SetTransparent(Color32:color, byte colorRange);

Set the GIF play mode before Save (Normal, Reverse, Ping-pong)
ProGifManager.Instance.m_GifRecorder.recorderCom.m_EncodePlayMode =
ProGifRecorderComponent.EncodePlayMode.Reverse;

Callbacks
The below callbacks are handled in the StartRecord, SaveRecord and StopAndSaveRecord methods, please register these callbacks if need, by assigning your methods/Actions to receive updates from them.

Action<float>: onRecordProgress
Action: onRecordDurationMax
Action: onPreProcessingDone
Action<int, float>: onFileSaveProgress
Action<int, string>: onFileSaved

Any confuse? Don't worry, there are plenty of demo scenes included, we will show you step by step in the scenes.

(8) ProGifManager : Player (GIF Preview)

To play gif after recording complete
Play the recorded gif frames stored in the recorder:
ProGifManager.Instance.PlayGif(Image:targetImage, Action<float>:onLoading);

* For the display target, supports Image, RawImage, Renderers(Meshes such as Plane, Cube, Sphere, etc, and GuiTexture).

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();

Set the flag to enable/disable Memory Usage Optimization:
ProGifManager.Instance.SetPlayerOptimization(bool:enable);

Set Ping-pong play mode:
ProGifManager.Instance.m_GifRecorder.PingPong();

Set Reverse play mode:
ProGifManager.Instance.m_GifPlayer.Reverse();

Callbacks

Reports the loading progress, instantly finish if play GIF using the recorder source.
ProGifManager.Instance.SetPlayerOnLoading(Action<float>:onLoading);

The callback to be fired on every frame during play GIF. Pass a GifTexture each time.
ProGifManager.Instance.SetPlayerOnPlaying(Action<GifTexture>:onPlaying);

(9) ProGifManager : Clean Up Memory

The GIF Manager handles memory clean up when the recorder or player restart, but in case you want to implement the Clear methods manually. You can use the below methods:

Clear the recorder and player:
ProGifManager.Instance.Clear();
Or
Clear the recorder:
ProGifManager.Instance.ClearRecorder();

Clear the recorder and ensure the textures not being cleared too early(for the case the recorder source is in use by the player):
ProGifManager.Instance.ClearRecord_Delay(Action<string>:onClear);

Clear the player:
ProGifManager.Instance.ClearPlayer();

More parameters and methods available in ProGifManager class or through the m_Recorder and m_GifPlayer variable in the class.

(10) Giphy API Helper

To use Giphy API, it requires a Giphy account to apply for the API keys.
APPLY HERE: https://developers.giphy.com/dashboard

How to USE? Run the demo scene for details!
Demo scene included:

1. **GifApi+ProGifPlayer Demo.unity,**
2. **GifApiDemo.unity**

You can also find our API documentation on our site:
https://www.swanob2.com/jsontool

(11) 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, QQZone, Weibo, Baidu, MySpace, LineMe, Skype).

Share GIF and/or text message:

```
GifSocialShare gifShare = new GifSocialShare();

gifShare.ShareTo(Social: socialPlatform, string: title, string: description,
        string: url1, string: url2, long: phoneNo, string: tags)
```

(12) Demo Scenes

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

ProGifDemo_Panels_Show_or_Hide_UI.unity
This scene shows the steps of record, playback and change settings with our UI templates. Also showing how to record GIF with or without UI, by changing the UI Canvas render mode.

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

GifApiDemo_GetById.unity
This scene shows how to get GIF from Giphy.com by id. And, share the GIF with its Bitly_Url/Url/Id responded by the API.

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

MobileMediaTest.unity
This scene simply shows how to pick and save media files to Android Gallery & iOS Photos.

ProGifDemo_TransparentSetupExample.unity
This scene shows how to create better quality transparent GIF, plus some tips in the scene.

THANK YOU

Thank you for using our assets!
For any problem and bug report please contact us at swan.ob2@gmail.com.
Remember to rate this asset on the Asset Store. Your review is always appreciated, and very important to the development of this asset!

Review And Rating

Visit our asset page to find out more!
https://www.swanob2.com/assets

SWAN DEV