Animated GIF Player

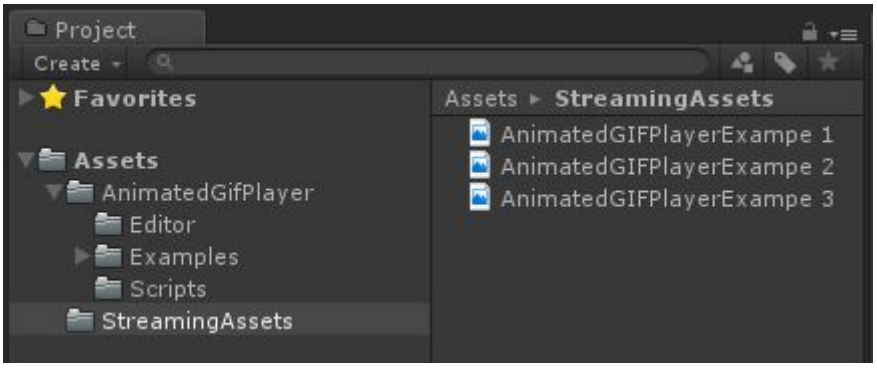
Animated GIF players allows Unity3d to play animated GIFs. No additional setup or coding is required. Also great for displaying short videos with no sound (for example tutorials). Videos are easily converted to animated gifs using any number of online tools like this one.

See a video that demonstrates the usage of Animated GIF Player here.

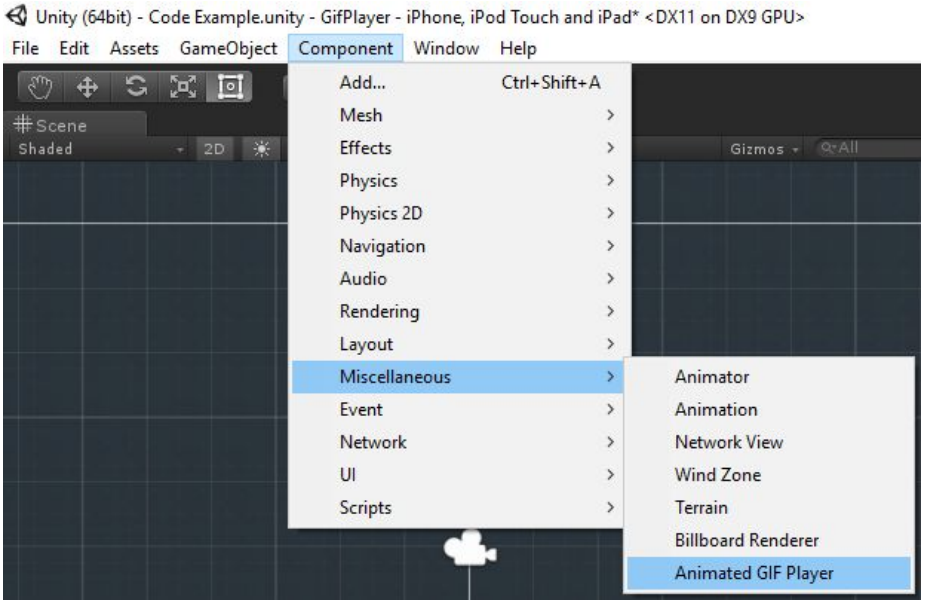
Basic Usage

Go through the following steps to start using animated GIFS in Unity3d:

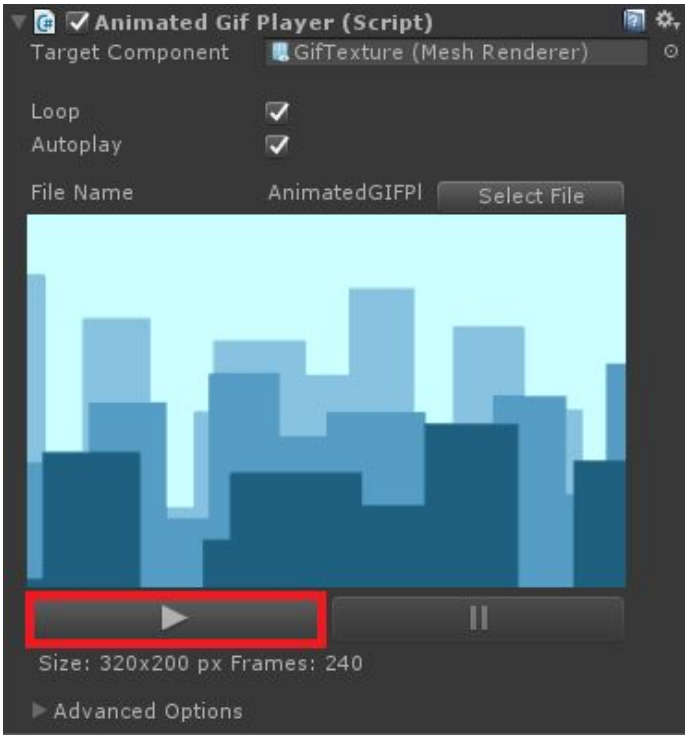
1. *Add an animated GIF to the **StreamingAssets** folder in the root of your Assets folder.*



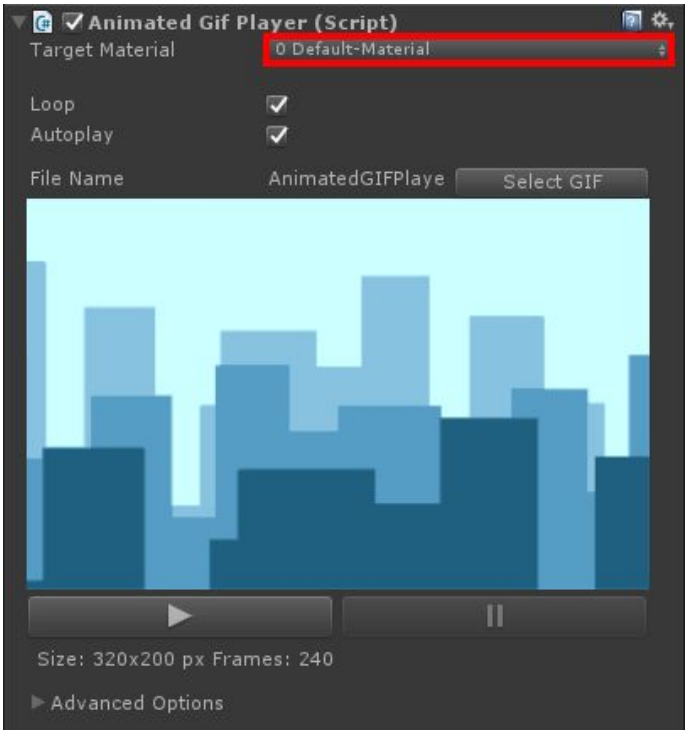
2. *Add the Animated Gif Player component to the gameobject that you want to show the GIF on.* For example an UI Raw Image element or a any gameobject with a Renderer, for example a quad. Do this by selecting the gameobject and then Component > Miscellaneous > Animated Gif Player or add it by clicking on “Add component” in the gameobject inspector and type Animated Gif Player in the search box.



3. *Press the “Select File” button to choose a GIF from your StreamingAssets folder.* You can now press the play button to test your GIF in the editor.



4. *If the renderer has more than one material you can use the dropdown to select which material to use for the GIF.* This option is not visible when there is only one material..



Advanced Options

In the inspector there are several advanced options visible:

Threaded Decoder: When enabled the player decodes each GIF frame in a separate thread for increased performance. (default: enabled)

Compatibility Mode: When enabled all dispose methods are supported by the decoder at the cost of memory and CPU usage. Use this when GIFs are not displayed correctly. In practice this is almost never needed. (default: disabled)

Cache Frames: When enabled each frame is cached in memory after decoding. Decoding is slightly slower. Use this when looping a GIF and you don't want to keep the decoder running. (default: disabled)

Buffer All Frames: When enabled all frames are loaded and cached immediately when the GIF player is initialized. Startup is a bit slower since playback can only start when all frames are decoded. When disabled each frame is decoded when needed. (default: disabled)

Override Time.timeScale: When disabled playback speed scales with time.timeScale. Use this when you want to change playback speed independent of the time scale. This can be useful when setting timeScale to 0 to pause the game but GIFs should still play at the same speed. (default: disabled)

How Does it Work

Animated GIF Player reads a GIF from the StreamingAssets folder and will read the header of the GIF. When finished a single Texture2D is created with the same width and height as the GIF. This Texture2D is then set as the texture to use for the gameobject the Animated GIF Player is attached to. Next, all frames are decoded in a separate thread (if enabled) and when needed (if buffering is disabled). Frames are uploaded to this Texture2D when needed using LoadRawTextureData.

Animated GIF Player doesn't use any plugins or any platform specific code so it *should* work on all devices.

Loading of GIF files from urls is also supported set the filename to an url (including http:// or https://).

Included Examples

Mesh Renderer Example: Usage with a 3D object with a Mesh Renderer.
Sprite Example: Usage with a sprite as the receiver of the animated GIF.
UI Example: Example with the GIF placed on a Canvas Raw Image element.
Code Example: Are more advanced example that shows loading and starting the animated GIF from code.

API

Variables

Loop	Set this to true to make the GIF loop (default: true).
AutoPlay	Set this to true to start playing the GIF when initialized.
Width	Returns the width of the GIF.
Height	Return the height of the GIF.
Path	Set the path to the GIF (default: StreamingAssetsPath).
FileName	The filename of the GIF in the StreamingAssets folder (including extension or url of the GIF image (e.g.: http://www.example.com/example.gif).
CacheFrames	Set this to true to cache each frame in memory. Set this to true to buffer and cache all frames after initialize.
BufferAllFrames	Set this to true to decode each frame in a separate thread.
UseThreadedDecoder	The target component for the GIF. Valid targets are RawImage, MeshRenderer and SpriteRenderer.
OverrideTimeScale	Set Gif playback speed independent of Time.timeScale. This is useful when you want to pause the game but want to continue playback of a Gif.
TimeScale	When OverrideTimeScale is set to true this sets the playback speeds of the Gif (e.g.: 2.0 plays the Gif at double speed).
TargetComponent	The target to which the Gif is written.
State	Returns the state of the player.
GifTexture	The texture which is used to display the GIF frames.
CompatibilityMode	Set this to true to allow support of more disposal methods. Use this when GIFs are not displayed correctly (default: false).

Public Functions

Init	Starts initialization of the GIF player and displays the first frame when ready.
Play	Starts playing the GIF.
Pause	Pauses playing of the GIF.
GetNumberOfFrames	Gets the number of frames in the GIF. Only returns the number of frames that have already been decoded.

Messages

OnReady	Called when the GIF has been initialized and is ready to play.
OnLoadError	Called when loading of the GIF has failed

Contact

For technical support and/or suggestions contact us at
animatedgifplayer@oldmoatgames.com

Rate us @:
Animated Gif Player for Unity3d
