

# uGIF User Manual

Version 1.0

## TABLE OF CONTENTS

<b>1. Introduction</b>	<b>2</b>
<b>2. Importing a GIF file</b>	<b>3</b>
<b>3. Using the Gif Player</b>	<b>6</b>
<b>4. Scripting Documentation</b>	<b>8</b>

# 1. Introduction

uGIF allows the use of GIF animations in Unity. The software is entirely integrated into Unity, you do not need to use any external tools. The editor allows you to view the decoded animation, modify its settings (frames per second etc.) and then export it to an asset that is usable at runtime.

## **About the Gif decoder:**

uGIF uses a slightly modified version of the libnsgif library ([original source](#)) which is a C library used in the NetSurf project (the modified source code is included in the package). The modified version has some functionality added to make it accessible to the C# scripts in Unity.

## **About the exported assets:**

After the Gif file has been decoded, it has to be exported to a Text Asset which contains the meta-data of the animation and all the frames in PNG format.

The reason why the Gif files have to be converted is so that the runtime loading can be done entirely in C# directly in Unity, this allows the animations to be compatible with every platform that Unity supports, without having to cross compile the C library.

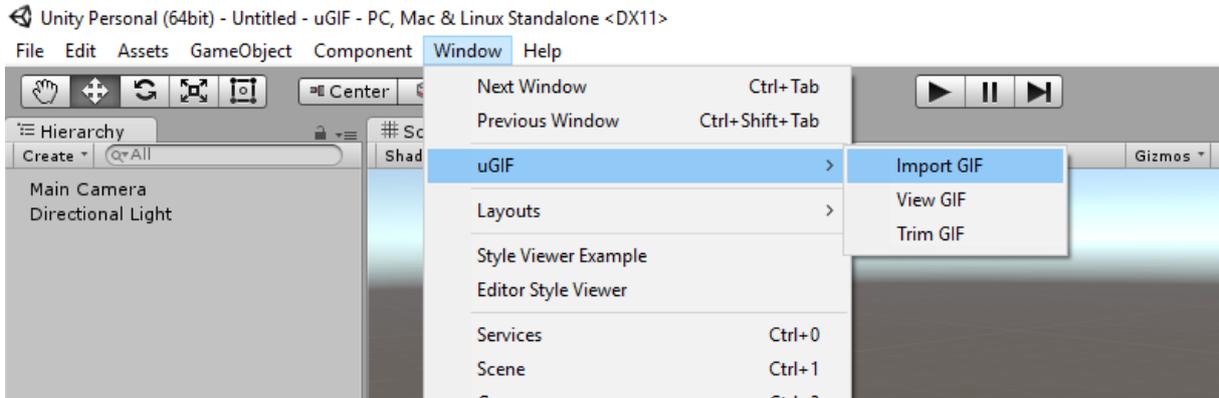
It is still possible to create custom code that will use the library to decode Gif files at runtime but uGIF does not currently provide this functionality because of eventual platform compatibility issues.

## **Backwards compatibility:**

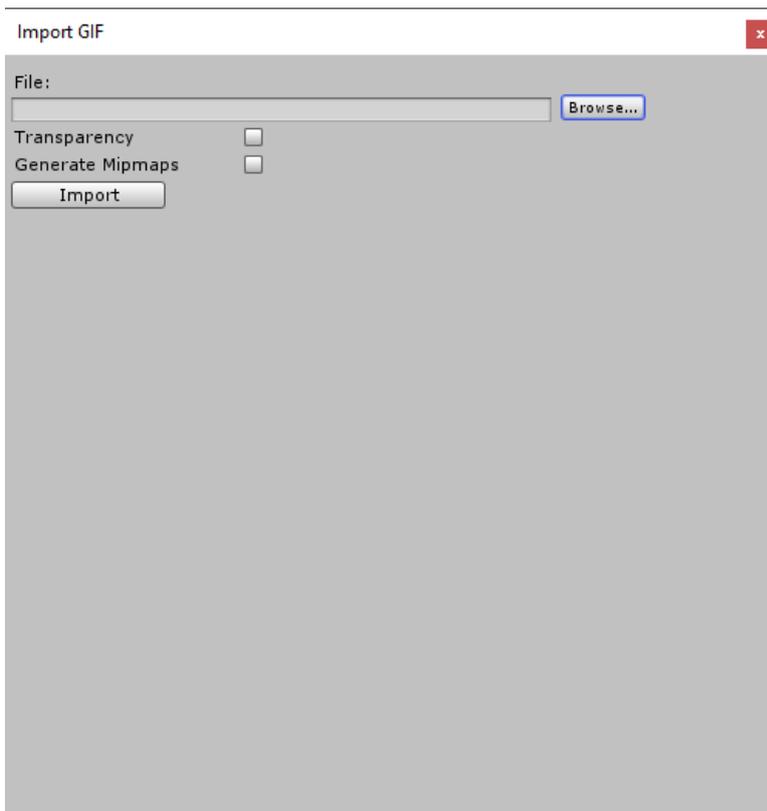
The package will fully support backwards compatibility of animations imported with eventual older versions of uGIF.

# 2. Importing a GIF file

The first step to start using a GIF animation in your project is to access the GIF importer window. To do so you can navigate in the editor to Window > uGIF > Import GIF.

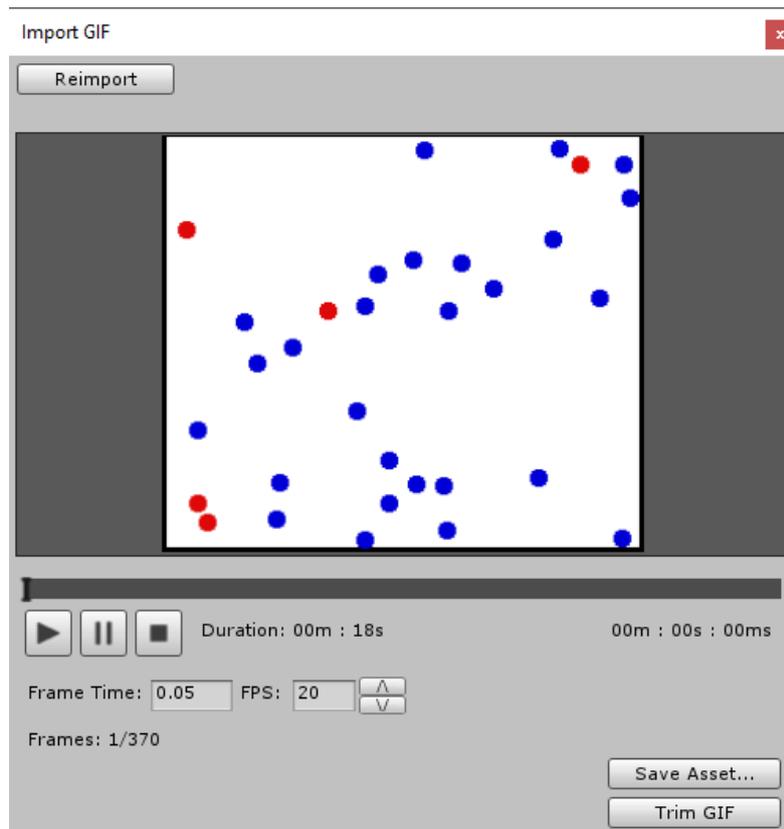


This window will appear:



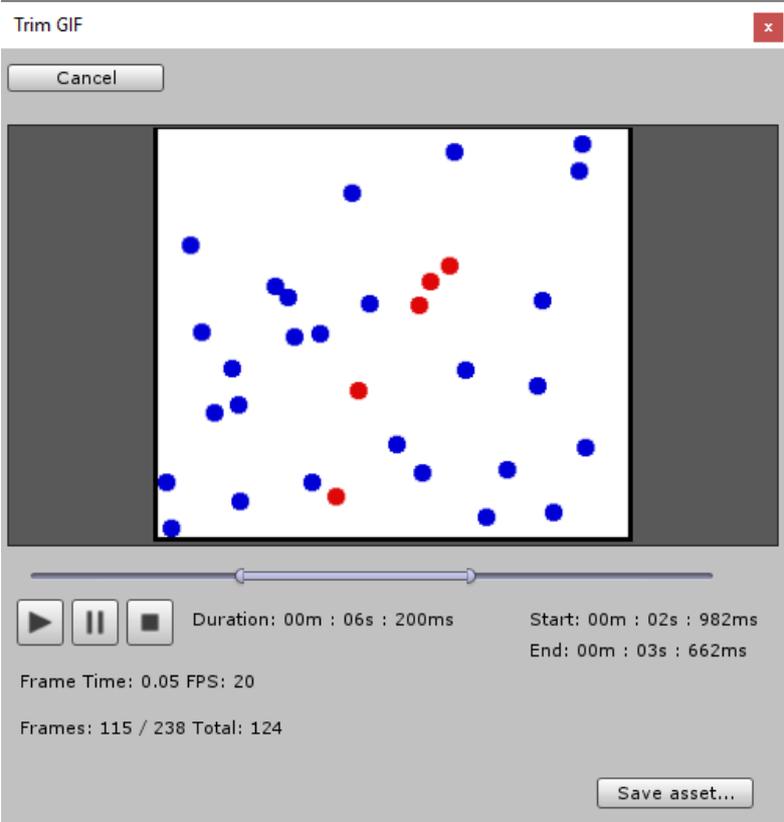
1. Select the .gif file you want to load
2. Allow or disable transparency (if transparency is disabled the alpha of every pixel will be set to 1)
3. Choose if the textures should generate mip maps

After clicking the Import button the animation will be loaded and you will see the following screen:



In this window you can play the animation, view any frame and change the speed settings. If you are satisfied with the animation you can click "Save Asset", however, if you only want to use a part of the animation you can click on "Trim GIF" this will close the current window and open the animation in the GIF Trimmer.

The GIF Trim window allows you to select a starting and ending point for the animation and then export it to a separate asset.



# 3. Using the Gif Player

After you have exported a gif to an asset, you can use the Gif Player script to play the animation at runtime.

Assign the “File” field to the asset you previously saved.

Now you can choose between three **playback outputs**:

1. **UI**: This mode will use a RawImage component to display the animation
2. **Material**: Use a material to play the animation, you can also specify the name of the material property you wish to use.
3. **LegacyGUI**: Use the old GUI system.

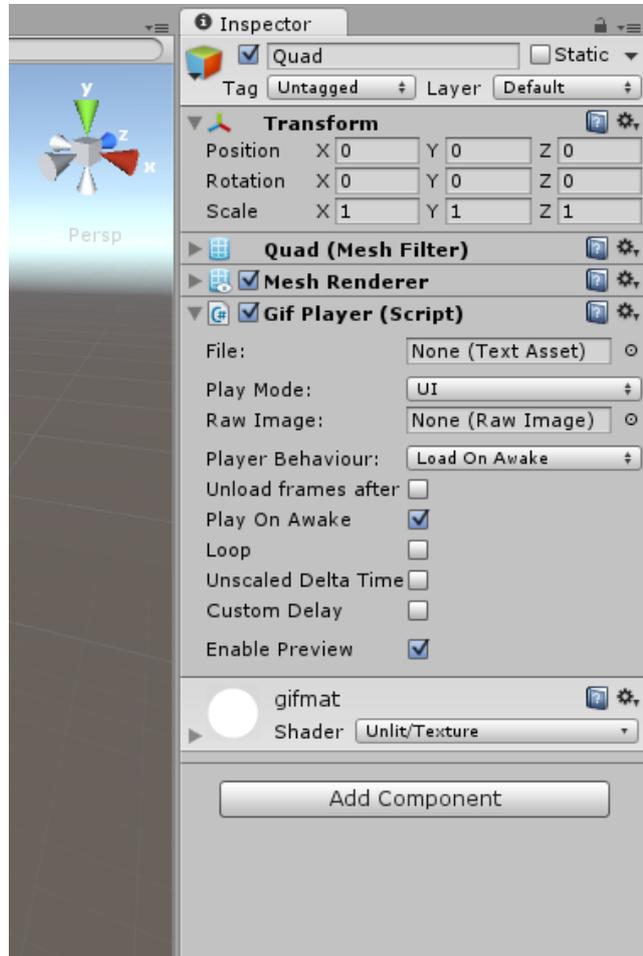
The **Player Behaviour** options lets you choose between:

1. **Load On Awake**: Loads the whole animation on the Awake() event.
2. **Load On Start**: Loads the whole animation on the Start() event.
3. **Load Progressively**: While the previous options will most likely lock up the thread, this option will load one Texture every Update(), this method does not lockup the thread and the Player can start the playback immediately. **This is the recommended option.**
4. **None**: The player will not load anything.

**Unload frames after playback**: When this option is enabled the frames of the animation that have already been played will be unloaded. This is useful for animations that need to be played once in the scene and never be used again.

Note that Unity does not unload the textures immediately, it may take some time before they are effectively unloaded from memory.

Additionally, the Loop option will not work when this option is active.



**Play On Awake:** This option will start the animation playback as soon as the game starts.

**Loop:** This option will make the animation loop infinitely.

**Unscaled Delta Time:** When this option is enabled, the player will use the unscaled delta time instead of deltaTime. This will cause it to ignore the setting of Time.timeScale.

**Custom Delay:** Lets you override the frame time of the animation and set a custom one.

**Enable Preview:** When enabled the UI or Material will display the first frame in edit mode.

# 4. Scripting Documentation

The scripting documentation can be found in the package at the path  
Assets/uGIF/Documentation/Scripting  
or at <http://www.orbitsoft.ugu.pl/ugif/docs/index.html>