

MOD Player Documentation (v1.3.2)

by Mick @ GamePhase

(For Gamemaker Studio)

Introduction

MOD Player is an extension that uses LibModPlug and audio buffers in GMS to play MOD music in 22 formats¹ (including MODs created with ProTracker, FastTracker, ScreamTracker and Impulse Tracker). MOD Player currently supports Windows and Android. Individual channels can be muted and tempo can be changed in real time. An example of usage is included in the extension.

License

The source for LibModPlug is released to the public domain so there is no restrictions in using this extension in your games.

Quick Start

To use MOD Player in your project, you need to add the MOD Player extension to your project (from Marketplace / My Library in Gamemaker Studio). The object *obj_mod_player_demo* is an example (for testing) and can safely be deleted from your project (it's recommended that you take a look at it first).

Once the extension is added to your project you can simply use the code below to load and play a song. You don't need to add *obj_mod_player* to your room(s), it's a persistent object and an instance of it will be added automatically the first time the script `MOD_LoadSong(...)` is called.

```
sound_index = MOD_LoadSong("second_pm.s3m"); // Load a song
if(sound_index != noone) { // Check if song has been loaded
    MOD_Play(); // Play the song
}
```

For more advanced use, check *obj_gme_demo* and the documentation of scripts on the next page and onwards.

Important notes for pre-gradle GMS1.4

If you have a version of GMS1.4 older than 1.4.1675, you need to remove the *libmodplug.jar* file from the following folder in your project: "yourproject.gmx\extensions\LibModPlug\AndroidSource\libs". If you don't delete that file, you will be unable to compile the project for Android.

¹ See the full list of supported formats at the end of this document.

Scripts

MOD_LoadSong(filename, repeat_count)

Arguments: filename (string), repeat_count (integer)

Returns: The index of the sound queue on success, *noone* otherwise.

Description: Load a song file with the given filename. If the file can't be found or another problem is encountered, the value *noone* will be returned, otherwise the index of the sound queue is returned. The returned sound queue index can be used with the normal audio_ functions (audio_sound_gain etc.). The second argument defines how many times the song should be repeated (looped). If you set this argument to 0 then no looping will happen, -1 will result in the song being looped indefinitely.

MOD_Play()

Arguments: none

Returns: nothing

Description: Play the loaded song.

MOD_Pause(pause)

Arguments: pause (bool)

Returns: nothing

Description: Pause / unpause the loaded song. Calling MOD_Play() after this will resume playback at the paused position.

MOD_Stop()

Arguments: none

Returns: nothing

Description: Stop the loaded song. Calling MOD_Play() after this will restart the song from the beginning.

MOD_GetTempo()

Arguments: none

Returns: The tempo of the loaded song (bpm)

Description: Returns the tempo (beats per minute) of the loaded song.

MOD_SetTempo(tempo)

Arguments: tempo (integer)

Returns: nothing

Description: Set the tempo of the loaded song in beats per minute (bpm)

MOD_NumChannels()

Arguments: none

Returns: The number of channels (integer)

Description: Returns the number of channels of the song.

MOD_MuteChannel(channel, mute)

Arguments: channel (integer), mute (bool)

Returns: nothing

Description: Mute or unmute a channel of currently loaded song. Setting the mute argument to true (or 1+) will mute the voice, setting it to false (or 0) will unmute the voice and set the channel volume to max (64). The voice argument is zero based, meaning the first channel is numbered 0 and so on.

MOD_GetChannelVolume(channel)

Arguments: channel (integer)

Returns: The channel volume (integer, 0..64)

Description: This script returns the channel volume (or gain). Max channel volume is 64.

MOD_SetChannelVolume(channel, volume)

Arguments: channel (integer), volume (integer, 0..64)

Returns: The channel volume (integer, 0..64)

Description: This script sets the channel volume. 0 = total silence, 64 = max volume. The new channel volume will be returned by the script. Note that if you have set a channel's volume to <64 and then use the MOD_MuteChannel(...) script to mute and unmute the channel, the channel volume will be reset to 64.

MOD_GetLength()

Arguments: none

Returns: The length of the MOD (integer)

Description: This script returns the length of the song in milliseconds.

MOD_GetName()

Arguments: none

Returns: The MOD name (string)

Supported Music Module Formats

Extension	Description
MOD	ProTracker (Amiga)
STM	ScreamTracker 2
S3M	ScreamTracker 3
XM	FastTracker 2
IT	Impulse Tracker
669	Composer 669
AMF	ASYLUM Music Format / DSMI Advanced Music Format
AMS	Extreme's Tracker / Velvet Studio
DBM	Digi Booster Pro
DMF	X-Tracker
DSM	DSIK Format
FAR	Farandole Composer
MDL	DigiTrakker
MED	OctaMED (Amiga)
MTM	MultiTracker
OKT	Oktalyzer
PTM	PolyTracker
ULT	UltraTracker
UMX	Unreal Music Package
MT2	MadTracker 2
PSM	Epic Megagames MASI

Troubleshooting

Do you experience noise in music playback?

If you have a room / game speed of lower than 60, you may need to increase the number of buffers used by the MOD Player extension. You can do this by increasing the value of the variable *buffer_count* to 20 (or even higher) in the create event of *obj_mod_player*.