

SID Player Documentation (v1.2.2)

by Mick @ GamePhase

(For Gamemaker Studio)

Introduction

With this extension you can use proper SID music (SID 6581 / 8580) in your games. The music is played back using audio buffers in GMS so no external audio instances are needed. Multi-speed songs are supported and also 2SID or 3SID songs are supported (played back in mono). The extension supports the following export formats: Windows and Android.

License

This extension is based on cSID-light by Mihaly Horvath (known as Hermit in the C64 scene). Read more about cSid here: <http://csdb.dk/release/?id=156587>. He has released the code under a "WTF: do what you want with this code but please give credits" license so you are allowed to use this extension in your games but please give credit to Mihaly Horvath if you do.

Quick Start

To use SID Player in your project, you need to add the SID Player extension to your project (from Marketplace / My Library in Gamemaker Studio). The object *obj_sid_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_sid_player* to your rooms, it's a persistent object and an instance of it will be added automatically the first time the script *SID_LoadSong(...)* is called.

```
sound_index = SID_LoadSong("Adventure.sid"); // Load a song
if(sound_index != noone) { // Check if song has been loaded
    // Initialise the default subtune for playback
    SID_InitSubtune(SID_DefaultSubtune());
    SID_Play(); // Play the subtune
}
```

For more advanced use, check *obj_sid_player_demo* and the documentation of scripts below.

Important notes for pre-gradle GMS1.4

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

Scripts

SID_LoadSong(filename)

Arguments: filename (string)

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.)

SID_NumSubtunes()

Arguments: none

Returns: The number of subtunes in the loaded song (integer)

Description: This script will return the number of subtunes in the loaded song, use with SID_InitSubtune() explained below.

SID_DefaultSubtune()

Arguments: none

Returns: The default subtune in the song (integer)

Description: Return the default subtune index defined in the loaded song.

SID_InitSubtune(subtune)

Arguments: subtune (integer)

Returns: 1 on success, 0 otherwise

Description: You need to set the subtune before calling SID_Play() or any other scripts below. You can get the number of tracks in the song with the script above.

SID_Play()

Arguments: none

Returns: nothing

Description: Play the loaded song / subtune.

SID_Pause(pause)

Arguments: pause (bool)

Returns: nothing

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

SID_Stop()

Arguments: none

Returns: nothing

Description: Stop the loaded song / subtune. Calling SID_Play() after this will restart the track from the beginning.

SID_GetTitle()

Arguments: none

Returns: The SID song title (string)

SID_GetAuthor()

Arguments: none

Returns: The SID song author (string)

SID_GetInfo()

Arguments: none

Returns: The SID song info field (copyright information etc.) (string)

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 SID 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_sid_player*.