

Audio

Playing music or sounds is an essential part of any game. GDevelop provides you with several conditions and actions to play audio files.

Playing a sound or a music file

Before a sound/music can be played, you have to first ensure that the player interacted with your game (a simple click/touch on the screen is ok to unlock audio). This is a limitation of browsers and cannot be worked around. In your final game, it's recommended to have on your first scene a "Press any key or touch the screen to continue" message.

The easiest way to play an audio file is to use the action "**Play a sound**" or "**Play a music file**". Choose the file to be played for the first parameter of both actions.

The supported audio file formats for GDevelop are Waveform Audio Format (wav), MPEG Layer 3 (mp3), Ogg Vorbis (ogg), and Advance Audio Coding (AAC)

.AAC files are much more broadly supported by different browsers and OSes, iOS included, so it is recommended to use .aac files.

These formats can all be used for music or sound effects, although it is strongly recommended to use either mp3 or ogg for music due to wav file size. A 320 kbps mp3, ogg, or AAC file will be nearly identical in audio quality to a maximum quality wav file, but take up 1/4th the file size.

If you distribute your game on the web, some browsers also support other file formats. For example ogg; ogg is a superior alternative to mp3 for music. Unfortunately, it's not supported by all browsers. See [more information on Wikipedia](#).

You can use the in-built sound effect creator tool called jfxr for making sound effects (This tool is only available for the desktop version) [Learn how to use jfxr here](#)

When you choose an audio file, it is added to the game's resources, to be loaded during game startup. To minimize your game's size, be sure to remove audio files that are no longer in use by your game actions.

Choosing between Sound or Music Events

GDevelop has two different methods of playing audio files. These are listed in the

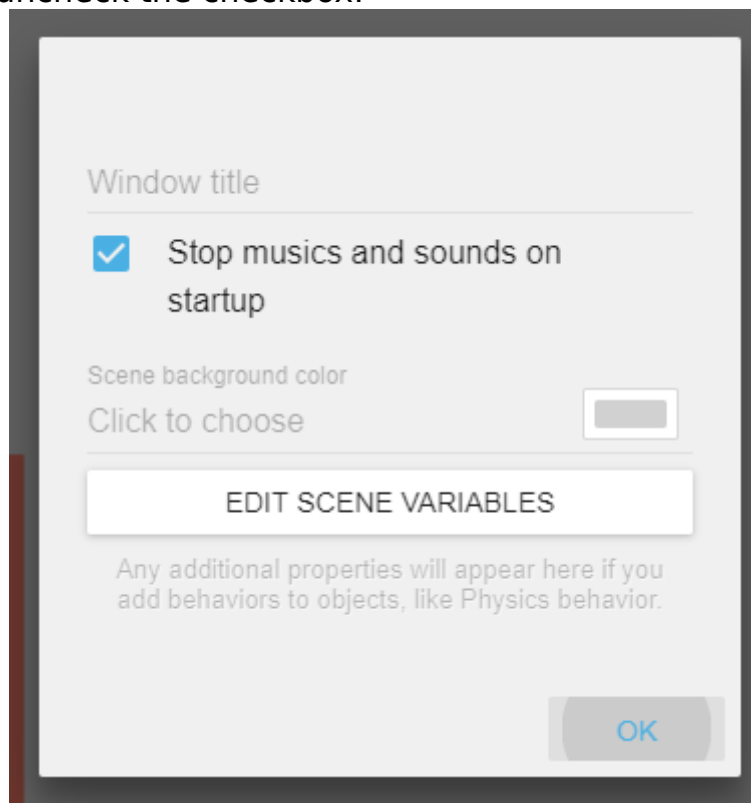
engine as **Play a Sound** or **Play a Music file** events, along with related events to these two different methods.

In the majority of use cases, audio should be played using the **Play a Sound** events. This is a newer method of playing audio, and will lead to less issues with looping sound files. Think of these events as playing a sound *file* and *not* playing a *sound effect* to avoid confusion.

The music events are an older method of playing sound files, and in most cases does not buffer the audio. Due to this, it can lead to a slight pause when looping a sound file. These events are mostly kept for backwards compatibility, and are not recommended for use when building a new game.

Keep music and sounds playing between scenes

When a new scene starts, by default, sounds and music are stopped. If you want to keep them playing, open the properties of the scene (right-click on the scene, in the scene editor) and uncheck the checkbox:



Using channels

When using the “Play a sound” or “Play a music file” action, audio files are played immediately. On completion, they are removed from memory. Looping is an exception. When you set the action to loop the audio file, it will be played forever (Endless Loop). You must use an action to stop all sounds or music.

If you want more control over sounds, use the action “**Play a sound on a channel**” or “**Play a music file on a channel.**” The action is used in the same way, except that you have to enter a channel number. This channel number can then be reused in other actions or conditions. A “channel number” can be used to check to see if a sound is being played on a channel, or it can be used to modify some properties of

the music being played. For example, you could use a “channel number” to update the music volume dynamically according to the presence of enemies around the player.

Volume

Sounds and music are by default played with a volume of 100%, which is the maximum. When starting a new sound/music, you can enter a different value (between 0 and 100).

You can also use the action “Game global volume” to change the entire game's audio volume. A value of 0 means that no sounds and or music can be heard. This action is convenient when allowing the player to mute or change the game's volume. For instance, you might have a settings screen in your game that accesses the game's volume control. Mobile games usually have a button to mute the sounds of the game too.

Performance considerations

The sound events are for playing any sound files, not just sound effects files, and should be used for all audio needs *including music*, especially if you are going to use looping audio. The music events use a legacy method of playing audio, and remain available to allow for backwards compatibility with games built using them.

The first time music or sound is played, there can be a lag while the audio file is being prepared if you do not preload the audio. If it's essential to avoid any lag, you can preload the audio files in the resources panel. Within the resource list, click on the sound file in question and choose the appropriate **Preload as Sound** or **Preload as Music** option depending on which events you are using.

Another option is to play the audio or music file at the beginning of the scene where it's used. Set the initial volume 0; this will force the game to load the sound or music. The audio will be cached in memory. It will quickly load when called using your action.

More about the state of a sound/music

When you use the action to **play a sound or music on a channel**, it's put in the state “**playing,**” even if still **loading** in memory.

The condition “A sound is being played” (on this channel) is then true, while “A sound is stopped” (on this channel) will be false, as well as “A sound is paused” (on this channel again). When the sound is fully loaded, it will start to be really played on your device (so you will hear the audio).

The sound will be stopped when:

- It reaches its **end** and is not configured to loop.
- There is **an error during the loading** (in which case it will be considered as playing for a few milliseconds, then will be deemed to be stopped as it was unable to load).
- Or you used **the action** to stop a sound or music on the channel.

