

Table of contents:

0.0.0 Introduction (1)

1.0.0 First steps (1-3)

1.1.0 Place GlobalMusicSystem

1.1.1 GlobalMusicSystem

1.2.0 Play Music

1.2.1 Play Music via Nodes

1.2.2 Play Music via Trigger

2.0.0 Music Control (4-6)

2.1.0 Play & Stop Music

2.2.0 Music Settings

2.3.0 Get Music Data

3.0.0 Music Listener (6-7)

3.1.0 Interface: I_MusicSystem_Listener

3.2.0 Interface: Events

3.2.1 OnCurrentMusicTimeSignature

3.2.2 OnMusicFinished

3.2.3 OnMusicStart

4.0.0 Trigger (7-9)

4.1.0 Trigger Overview

4.2.0 Details Tab

4.2.1 Trigger: Music

4.2.2 Trigger: Collision

4.3.0 Override Trigger

5.0.0 DataTable DT_MusicSequences (9-12)

5.1.0 DataTable Overview

5.2.0 Sequence

5.2.1 Music Clip

5.3.0 Data Asset PDA_Music

==== 0.0.0 Introduction ====

This music system is just like this manual in ALPHA status! Feedback and suggestions for improvement are therefore very welcome. Feel free to contact me directly at: alex@audiolarks.com or via Discord: Audioaxel#2231

To achieve greater diversity, musicality and adaptability, the tracks included in the bundle are divided into various loops and transitions, called clips.

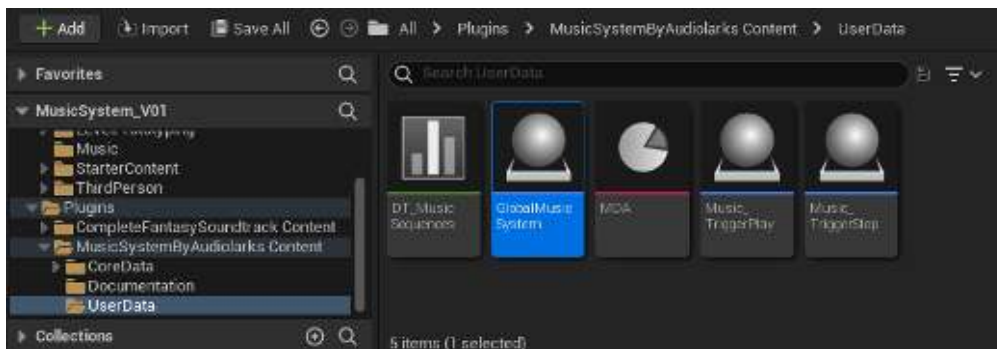
These clips are provided with different parameters in a DataTable and reassembled to Sequences or other tracks. Each sequence is assigned one or more DataAssets, which are then used to control the music via nodes.

The Music System serves first of all the fast and simple handling of the music contained in the bundle. However, it is possible to extend it with own music or to change preset properties. See especially point 5 – DataTable DT_MusicSequences.

==== 1.0.0 First Steps ====

--- 1.1.0 Place GlobalMusicSystem ---

Navigate to Plugins/MusicSystemByAudiolarks/UserData and place GlobalMusicSystem in the level. Make sure that this actor always remains present in the level.



--- 1.1.1 GlobalMusicSystem ---

In the details tab you have the possibility to activate the printing of debug and system information. Furthermore you can select a different DataTable (for the use of different bundles).

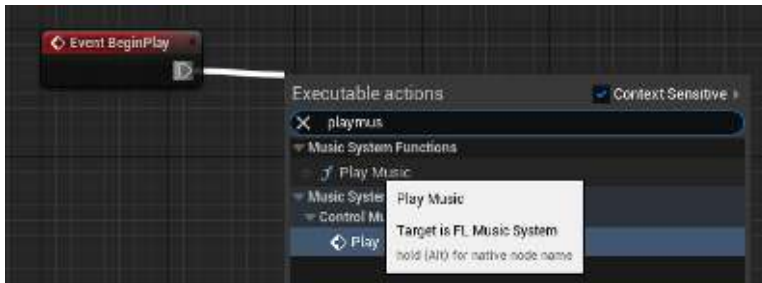
--- 1.2.0 Play Music ---

To play and control music you have the possibility to access various nodes in the Blueprints. You can find an overview under Plugins/MusicSystemByAudiolarks/CoreData/FL_MusicSystem

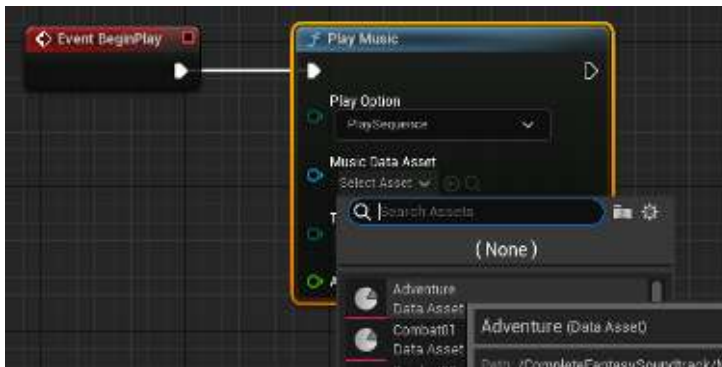
But you can also use the provided triggers directly under:Plugins/MusicSystemByAudiolarks/UserData/Music_TriggerPlay directly in the level.

--- 1.2.1 Play Music via Nodes ---

Open the Levelblueprint (or a BP of your choice) and search for PlayMusic.

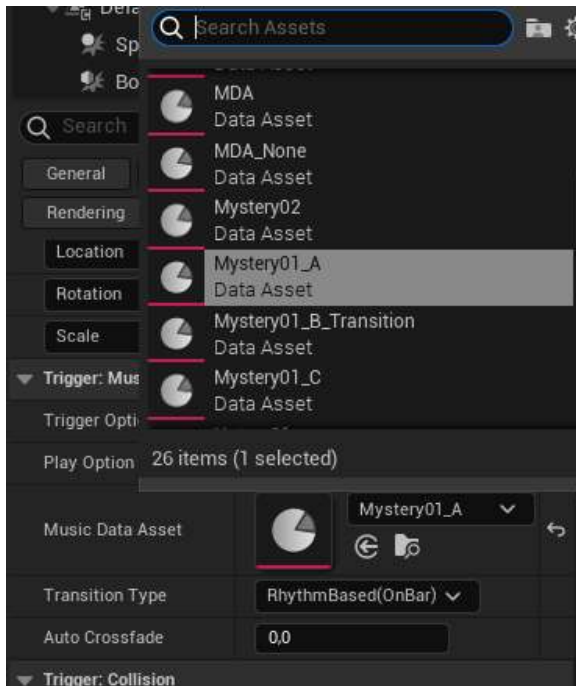


Now select a Track(DataAsset) under "MusicDataAsset". Save and compile the Blueprint and start the level. You should now hear your selected track.

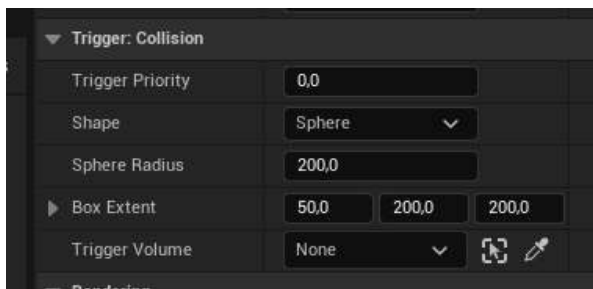


--- 1.2.2 Play Music via Trigger ---

Drag Music_TriggerPlay into your level and select a track (DataAsset) in the Details tab, section "Trigger: Music" under "MusicDataAsset".



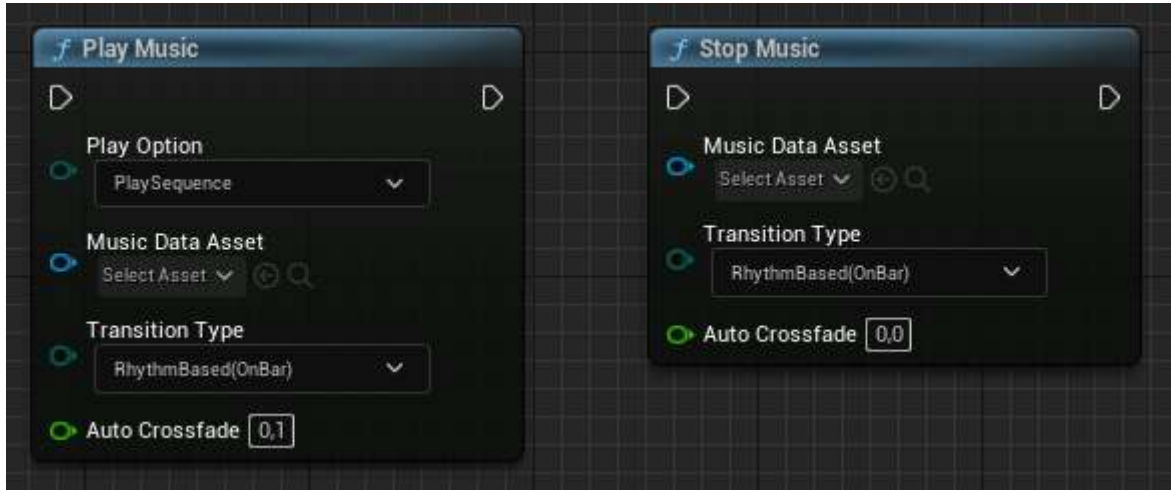
In the "Trigger: Collision" section you can now define the shape and size of the collider.



Start the level. If you don't spawn inside the collision, run in with your character. You should now hear the selected track.

===== 2.0.0 Music Control =====

--- 2.1.0 Play & Stop Music ---



Play Option:

PlaySequence: Starts or switches to the selected sequence or clip (DataAsset).

SwitchToClip: Switches to the selected clip within a track.

NOTE: This option is only for switching within a currently playing track. If the system is not currently in that track, the system will not be affected!

Music Data Asset:

Select the DataAsset corresponding to the Clips and Sequences to play them.

NOTE: At "Stop Music" you have the possibility to select MDA_None. With this "DataAsset" you can stop the player completely. Even tracks with "Priority" are stopped by this.

Transition Type:

RhythmBased(OnBar): Delays an execution until the next bar of the current track.

Crossfade(Immediately): Execution starts immediately. A change of music takes place only through crossfades.

Auto Crossfade:

The Auto Crossfade is used to easily adjust the crossfade between two tracks, but can also be used only as FadeIn or FadeOut.

=0 deactivates the crossfade. The default values in DT_Music_Sequences_"BundleName" now apply to the tracks

<0 the FadeIn of the next track is used for the FadeOut of the current track. The set value is added to the current value.

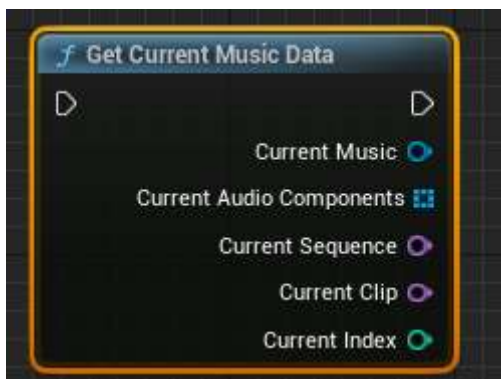
>0 the FadeIn is replaced by the absolute value of the negative Auto Crossfade.

--- 2.2.0 Music Settings ---



Allows to query and control the current volume defined between 0 and 1.

--- 2.3.0 Get Music Data ---



Allows to query current values of the music system.

Current Music:

Returns the currently played DataAsset

Current Audio Components:

Returns all currently playing audio components

Current Sequence:

Returns the current name of the sequence/track

Current Clip:

Returns the current clip name. This is composed as follows:
"sequenceName|"clipIndex".

Current Index:

Returns the current index of the current clip. The index is defined by the placement of the clip within a sequence in DT_MusicSequences.

==== 3.0.0 Music Listener ====

--- 3.1.0 Interface: I_MusicSystem_Listener ---

You have the possibility to execute functions on the rhythm of the track. To get access to it implement the interface:

"I_MusicSystem_Listener" in the corresponding blueprint.

To get information from the interface only when it is necessary, you have to subscribe to the music system with the node: "SubscribeToMusicSystem". Use "UnsubscribeToMusicSystem" to disconnect.



--- 3.2.0 Interface: Events ---

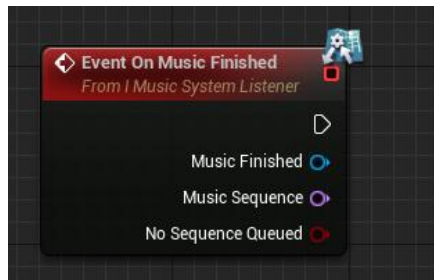
3.2.1 OnCurrentMusicTimeSignature

Provides the possibility to execute an execution on different time signatures of the current track. Furthermore, the event gives the current bar and the count time.



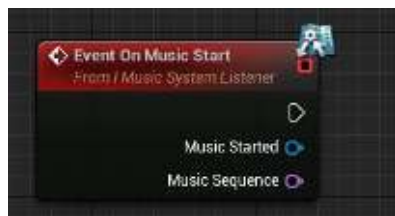
3.2.2 OnMusicFinished

Provides the possibility to execute an execution at the end of a track. Furthermore, the event outputs the DataAsset, the SequenceName and the information whether another track should be played.



3.2.3 OnMusicStart

Provides the possibility to execute an execution at the beginning of a track. Furthermore, the event outputs the DataAsset and the SequenceName.



==== 4.0.0 Music Listener ====

--- 4.1.0 Trigger Overview ---

The system provides 2 ready trigger blueprints.

"Music_TriggerPlay" & "Music_TriggerStop".

To use these, simply drag the trigger you want into your level and adjust it in the detail tab.

If you want to see these triggers in the game during the development process, you can set the bool "ShowTriggerInGame" to "true" in the construction script of the respective trigger.



--- 4.2.0 Details Tab ---

--- 4.2.1 Trigger: Music ---

In this section you will find exactly the same settings as for the nodes "PlayMusic" & "StopMusic" (see point 2.1.0 Play & Stop Music).

--- 4.2.2 Trigger: Collision ---

Trigger Priority:

This value decides which trigger should be used when different collisions overlap.

Shape:

Choose one of the implemented colliders. For complex rooms you have the option to replace the box or sphere with a trigger volume. With this selection you still have to assign the corresponding volume under "Trigger Volume".

Sphere Radius:

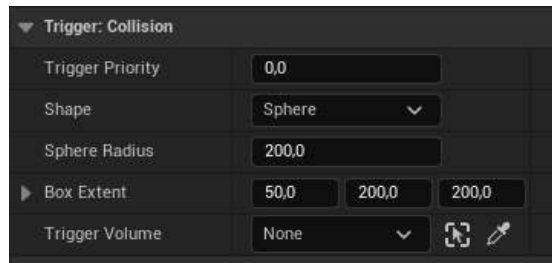
Increases or decreases the radius of the "Sphere Collision".

Box Extet:

Increases or decreases the volume of the "Box Collision" according to the x, y, z axis.

Trigger Volume:

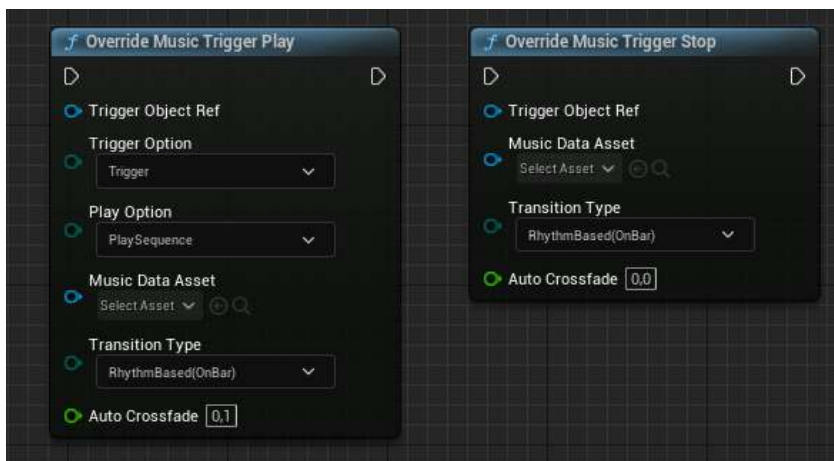
Use the dropdown or the eyedropper to select the corresponding "Trigger Volume". Make sure that "TriggerVolume" is also selected in "Shape".



--- 4.3.0 Override Trigger ---

The system offers the possibility to overwrite the selected values of the category "Trigger: Music" in Runtime.

To do this, select one of the corresponding nodes and reference the actor to be overwritten.

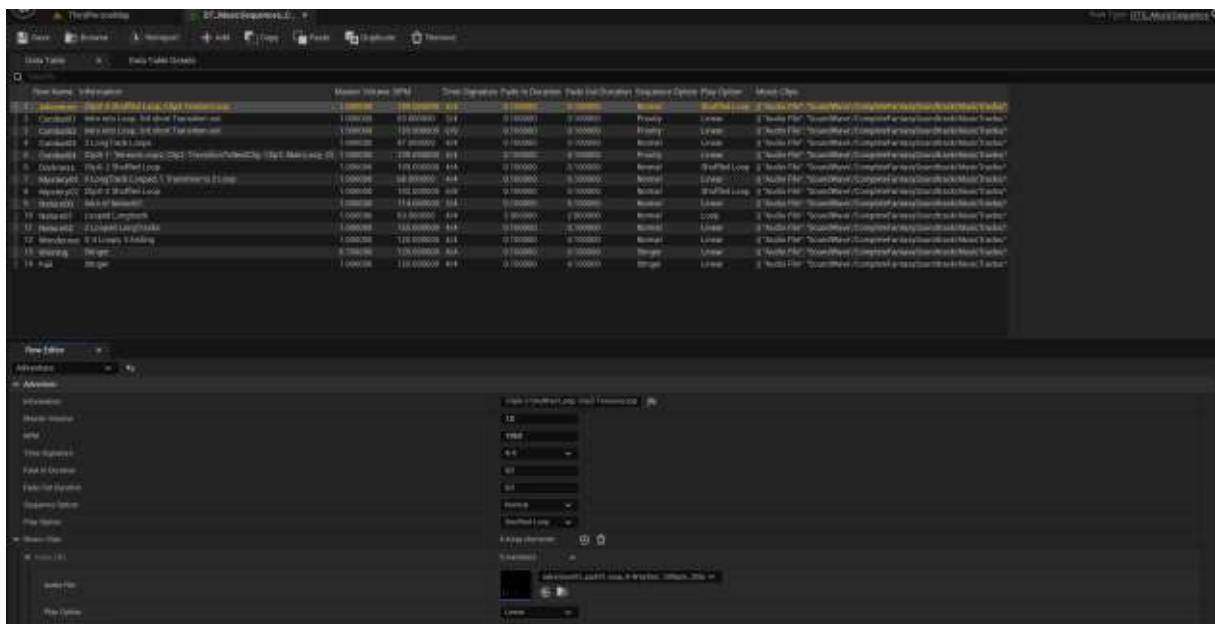


==== 5.0.0 Data Table DT_MusicSequences ====

--- 5.1.0 DataTable Overview ---

As already described in the introduction, in this "DataTable" the individual clips are put together again to form entire tracks and also provided with parameters.

A "DataTableRow" corresponds to a sequence or an entire track.



--- 5.2.0 Sequence ---

Information:

A text field without influence on the system. It is only used to save notes.

Master Volume:

Specifies the output volume of the entire sequence with a value between 0 and 1.

BPM:

An integer representing the speed of the song.

Time Signature:

In the dropdown you can select the time signature of the song.

Fade In Duration:

The time as a float that the track takes to reach its full volume (if auto crossfade is turned off!).

Fade Out Duration:

The time as a float that the track takes to fade out completely. (if Auto Crossfade is switched off!)

Sequence Option:

Normal:

Default setting of a Sequence.

Priority:

The Sequence must be explicitly terminated before switching to another Sequence is possible.

Stinger:

A musical interjection that plays independently of the current Sequence. Works well for short "winning jingles" or similar.

Play Option:

Linear:

The Sequence plays the Music Clips in a given order and does not repeat.

Loop:

The sequence plays the music clips in a predefined order and starts from the beginning as soon as the last clip is over.

ShuffledLoop:

The sequence plays all contained tracks that do NOT have the option "ExcludeFromShuffle" or "TransitionOut" in random order.

--- 5.2.1 Music Clip ---

The music clip is the smallest musical component in a sequence. Within a Sequence there are 1 or more Clips in a corresponding Array.

Audio File:

Wave or Sound Cue.

Play Option:

Linear:

The clip plays through once.

Loop:

The clip loops until you end it.

ExcludeFromShuffle:

The clip behaves like "Linear" except that it is ignored by the shuffle function.

TransitionOut:

This clip is played as the final clip when you exit (OnBar). If you switch to Immediately, this clip will be ignored.

Pre Roll Length (Bars):

The length of the intro or prelude in bars as an integer.

Body Length (Bars):

The length of the clip (without intro) in bars as an integer.

NOTE:

I would recommend a minimum length of 2 as the system may currently have errors.

PDA_Music:

The corresponding DataAsset through which the Sequence or also the Clip should be addressed.

--- 5.3.0 Data Asset PDA_Music ---

The Primary Data Asset `_Music` is the Base Class for all DataAssets that are used to control Sequences and Clips. If you want to add your own song, you have to create at least one DataAsset and assign it to the clip.

Music Sequence:

Must match the corresponding DataTableRowName.

Music Clip Index:

Must match the corresponding Clip Index. Here it is also possible to expand the array. This is for the Shuffle function to get a random entry into one of the clips. The DataAsset must then be assigned to each of the clips.