



For more information : <http://harmony.baptistebillet.com/>

Introduction

[Harmony](#) is a unity plugin and tool usefull for managing the sounds aspects of a project. It was created for the project The Key.

SetUp

Unpack [Harmony](#) plugin on your Unity game.

You need to place the "Editor" content from the [Harmony](#) plugin into your "Editor" folder if you have already one.

You need an "AudioMixers" and an "AudioClips" folder for running [Harmony](#). If you don't have one, running [Harmony](#) for the first time will create them.

Features of Harmony in Editor

Actualize

The Actualize function actualize the SoundManager gameobject of the Sounds folder with all the sounds of this folder. Alternatively, if there is no Sounds folder or SoundManager, it create them.

Clean

Clean all the clips or sources of the SoundManager.

New Sound Trigger

Create a new sound trigger in the scene. You can define before creating it:

- If the trigger will be destroyed after playing (can't loop so).
- The type of Collider you want to use (Sphere, Box, Capsule or Mesh).
- If the collider work only once, for the first correct thing who enter in.

After, you can change directly on the component the size of the collider, all the parameters of the audio source (set the clip by example), and the variables of the [Harmony](#) Sound Trigger script. You can define the category of elements who can be triggered and the lists to fill it. These lists are public, so you can also access them by script.

Features of Harmony in script

Base

The Base section is usefull for the basic and general functions of [Harmony](#). Play, Pause, UnPause, Stop...

Attach and Return

The Attach and Return functions exist in order to place some sources at some places. It can be useful to place some 3D sound sources. The return function bring them back to [Harmony](#).

Playlist

All the sources can be use as playlist in order to change clip easily in runtime and play differents clips, one after the other. The Playlist can't be fully empty, the playlist would automatically add a unique null clip.

Bugs and Issues

You can check all Bugs and Issues in the [Bug List](#)

PatchNote

v0.2 : No more overflow functions. No more automatic sources. No more name functions. All [Harmony](#) now work with AudioSource component.

Harmony Class Reference

The [Harmony](#) class hub of fonctionnalities.

Inherits MonoBehaviour.

Static Public Member Functions

static void [Play](#) (AudioSource _source, bool _destroyAfterPlaying=false, float _delay=0)

Play the source [More...](#)

static void [Pause](#) (AudioSource _source, float _delay=0)

Pause the source [More...](#)

static void [UnPause](#) (AudioSource _source, float _delay=0)

UnPause the source [More...](#)

static void [Stop](#) (AudioSource _source, float _delay=0)

Stop playing the source [More...](#)

static void [StopAll](#) (float _delay=0)

Stop all the clip which are playing from the source list [More...](#)

static void [StopAllBut](#) (AudioSource _source, float _delay=0)

Stop all the clip which are playing, except the _source [More...](#)

static AudioSource [SetSource](#) (string _clipName)

Set an AudioSource and return it [More...](#)

static void [DestroySource](#) (AudioSource _source)

Destroy the Source [More...](#)

static void [AttachTo](#) (AudioSource _source, Transform _transform)

Make _transform the parent of the gameobject of _source [More...](#)

static void [ReturnToHarmony](#) (AudioSource _source)

Make [Harmony](#) the new parent of the _source [More...](#)

static void [EnabledPlayList](#) (AudioSource _source, bool _enabled)

Enable or disable the PlayList mode of a _source [More...](#)

static void [EnabledPlaylistLoop](#) (AudioSource _source, bool _enabled)

Enable or disable the loop PlayList mode of a _source [More...](#)

static void [EnabledPlaylistRandom](#) (AudioSource _source, bool _enabled)

Enable of disable the random mode of the playlist of _source
[More...](#)

static void [NextClip](#) (AudioSource _source, float _delay=0)

Set the next clip in the playlist of _source [More...](#)

static void [PreviousClip](#) (AudioSource _source, float _delay=0)

Set the previous clip in the playlist of _source [More...](#)

static void [SetClipInPlaylist](#) (AudioSource _source, int _clipIndex, float
_delay=0)

Set the _clipIndex clip in the playlist of _source [More...](#)

static List< AudioClip

> [GetPlaylist](#) (AudioSource _source)

Get the Playlist as a list of AudioClip [More...](#)

static void [AddClip](#) (AudioSource _source, AudioClip _clip=null)

Add the _clip to the Playlist (at the end of) [More...](#)

static void [SubClip](#) (AudioSource _source, AudioClip _clip=null)

Remove the _clip from the Playlist [More...](#)

static void [DeleteClip](#) (AudioSource _source, AudioClip _clip)

Remove the _clip from the Playlist [More...](#)

static void [ResetPlaylist](#) (AudioSource _source)

Clear all the Playlist, then add a null clip [More...](#)

static void [SwapClip](#) (AudioSource _source, AudioClip _clip1, AudioClip
_clip2)

Swap the two clip in the playlist list [More...](#)

static void [SwapUpClip](#) (AudioSource _source, AudioClip _clip)

Swap the _clip with the clip above [More...](#)

static void [SwapDownClip](#) (AudioSource _source, AudioClip _clip)

Swap the _clip with the clip below [More...](#)

)

Add the `_clip` to the Playlist (at the end of)

Parameters

`_source` The source of the playlist

`_clip` The clip to add

§ AttachTo()

```
static void Harmony.AttachTo ( AudioSource _source,  
                                Transform _transform static  
                                )
```

Make `_transform` the parent of the gameobject of `_source`

Parameters

`_source` The AudioSource

`_transform` The new parent

§ CleanPlaylist()

```
static void Harmony.CleanPlaylist ( AudioSource _source ) static
```

Remove all the clip which are null from the playlist

Parameters

`_source` The source of the playlist

§ ClearPlaylist()

```
static void Harmony.ClearPlaylist ( AudioSource _source ) static
```

Reset the playlist but keep the first clip from the playlist

Parameters

`_source` The source of the playlist

[§](#) DeleteClip()

```
static void Harmony.DeleteClip ( AudioSource _source,  
                                AudioClip _clip      static  
                                )
```

Remove the `_clip` from the Playlist

Note

Only the first one found

Parameters

`_source` The source of the playlist

`_clip` The clip to delete

[§](#) DestroySource()

```
static void Harmony.DestroySource ( AudioSource _source ) static
```

Destroy the Source

Parameters

`_source`

[§](#) EnabledPlayList()

```
static void Harmony.EnabledPlayList ( AudioSource _source,  
                                      bool _enabled static  
                                      )
```

Enable or disable the PlayList mode of a `_source`

Parameters

`_source` The source to change
`_enabled` Enable the Playlist mode

[§](#) **EnabledPlaylistLoop()**

```
static void Harmony.EnabledPlaylistLoop ( AudioSource _source,  
                                          bool _enabled static  
                                          )
```

Enable or disable the loop PlayList mode of a `_source`

Parameters

`_source` The source to change
`_enabled` Enable the Playlist Loop mode

[§](#) **EnabledPlaylistRandom()**

```
static void Harmony.EnabledPlaylistRandom ( AudioSource _source,  
                                             bool _enabled static  
                                             )
```

Enable of disable the random mode of the playlist of `_source`

Parameters

`_source` The source to change
`_enabled` Enable or disable

[§](#) **GetPlaylist()**

```
static List<AudioClip> Harmony.GetPlaylist ( AudioSource _source ) static
```

Get the Playlist as a list of AudioClip

Parameters

`_source` The source of the playlist
`_clipIndex` The index of the clip to set
`_delay` The action can be delayed

§ NextClip()

```
static void Harmony.NextClip ( AudioSource _source,  
                               float         _delay = 0  static  
                               )
```

Set the next clip in the playlist of `_source`

Parameters

`_source` The source to set the next clip
`_delay` The action can be delayed

§ Pause()

```
static void Harmony.Pause ( AudioSource _source,  
                            float         _delay = 0  static  
                            )
```

Pause the source

Parameters

`_source` The AudioSource of the wanted clip
`_delay` Delay before proceed

§ Play()

```
static void Harmony.Play ( AudioSource _source,  
                           bool         _destroyAfterPlaying = false, static  
                           float        _delay = 0
```

)

Play the source

Parameters

<code>_source</code>	The AudioSource of the wanted clip
<code>_destroyAfterPlaying</code>	If the source gameobject must destroy itself after playing
<code>_delay</code>	Delay before proceed

[§](#) PreviousClip()

```
static void Harmony.PreviousClip ( AudioSource _source,  
                                float _delay = 0 static  
                                )
```

Set the previous clip in the playlist of `_source`

Parameters

<code>_source</code>	The source to set the previous clip
<code>_delay</code>	The action can be delayed

[§](#) ResetPlaylist()

```
static void Harmony.ResetPlaylist ( AudioSource _source ) static
```

Clear all the Playlist, then add a null clip

Parameters

<code>_source</code>	The source of the playlist
----------------------	----------------------------

[§](#) ReturnToHarmony()

```
static void Harmony.ReturnToHarmony ( AudioSource _source ) static
```

Make [Harmony](#) the new parent of the `_source`

Parameters

`_source` The source

[§](#) **SetClipInPlaylist()**

```
static void Harmony.SetClipInPlaylist ( AudioSource _source,  
                                       int          _clipIndex,  
                                       float         _delay = 0  static  
                                       )
```

Set the `_clipIndex` clip in the playlist of `_source`

Parameters

`_source` The source to setclip

`_clipIndex` The index of the clip to set

`_delay` The action can be delayed

[§](#) **SetSource()**

```
static AudioSource Harmony.SetSource ( string _clipName ) static
```

Set an AudioSource and return it

Parameters

`_clipName` The clip name to research

Returns

[§](#) **Stop()**

```
static void Harmony.Stop ( AudioSource _source,  
                           float        _delay = 0  static  
                           )
```

Stop playing the source

Parameters

`_source` The AudioSource of the wanted clip

`_delay` Delay before proceed

§ StopAll()

```
static void Harmony.StopAll ( float _delay = 0 ) static
```

Stop all the clip which are playing from the source list

Parameters

`_delay` Delay before proceed

§ StopAllBut()

```
static void Harmony.StopAllBut ( AudioSource _source,  
                                float _delay = 0 static  
                                )
```

Stop all the clip which are playing, except the `_source`

Parameters

`_source` The AudioSource of the wanted clip

`_delay` Delay before proceed

§ SubClip()

```
static void Harmony.SubClip ( AudioSource _source,  
                             AudioClip _clip = null static  
                             )
```

Remove the `_clip` from the Playlist

Note

Only the first one found

If the `_clip` is null, remove the last one on the Playlist

Parameters

`_source` The source of the playlist

`_clip` The clip to sub

[§](#) **SwapClip()**

```
static void Harmony.SwapClip ( AudioSource _source,
                                AudioClip _clip1,
                                AudioClip _clip2,      static
                                )
```

Swap the two clip in the playlist list

Note

If one of the clip is currently played in the playlist, it will continue playing

The next clip to be played will be the next on the list from the ancient index of the current clip

Parameters

`_source` The source of the playlist

`_clip1` The first clip to swap

`_clip2` The second clip to swap

[§](#) **SwapDownClip()**

```
static void Harmony.SwapDownClip ( AudioSource _source,
                                    AudioClip _clip,      static
                                    )
```

Swap the `_clip` with the clip below

Parameters

`_source` The source of the playlist

`_clip` The clip to swap

§ **SwapUpClip()**

```
static void Harmony.SwapUpClip ( AudioSource _source,  
                                AudioClip _clip     static  
                                )
```

Swap the `_clip` with the clip above

Parameters

`_source` The source of the playlist

`_clip` The clip to swap

§ **UnPause()**

```
static void Harmony.UnPause ( AudioSource _source,  
                              float _delay = 0     static  
                              )
```

UnPause the source

Parameters

`_source` The AudioSource of the wanted clip

`_delay` Delay before proceed