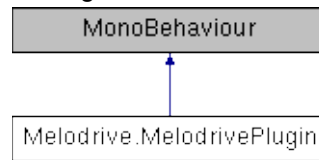


Melodrive.MelodrivePlugin Class Reference

Inheritance diagram for Melodrive.MelodrivePlugin:



Public Member Functions

delegate void	DebugHandler (string message)
delegate void	InitHandler (Style style, Emotion emotion)
delegate void	NoteOnHandler (string part, int num, int velocity)
delegate void	NoteOffHandler (string part, int num)
delegate void	ParamChangeHandler (string part, string param, float value)
delegate void	BarHandler (float bar)
delegate void	BeatHandler (float beat, float bar)
delegate void	BeatTickHandler (int tick, float beat, float bar)
delegate void	TempoChangeHandler (float bpm)
delegate void	NewMusicalSeedHandler (string name)
delegate void	CueChangeHandler (string cue, string seedName, Style style)
delegate void	ProjectLoadHandler ()
void	SetInstrumentsPathToStreamingAssets ()
void	SetInstrumentsPath (string path)
string	GetInstrumentsPath ()
void	Init (Style style=DEFAULT_STYLE, Emotion initEmotion=DEFAULT_EMOTION)
void	SaveProject (string filename)
void	LoadProject (string filename)
void	PreloadStyles ()
void	SetStyle (string style)
void	SetStyle (Style style)
Style	GetStyle ()
string	GetStyleString (Style style)
void	SetMusicalSeed (string seedName)
string	GetMusicalSeed ()
void	CreateMusicalSeed (bool setActive=false)
void	SaveMusicalSeed (string seedName, string filename)
void	LoadMusicalSeed (string filename)
string []	GetMusicalSeeds ()

string	GetCurrentEnsemble ()
string	GetCurrentEnsembleName ()
int	GetNumEnsembles ()
string	GetEnsembleID (int ensembleIndex)
string	GetEnsembleID (Ensemble ensemble)
string	GetEnsembleName (string ensembleID)
string	GetEnsembleName (Ensemble ensemble)
Dictionary< string, string >	GetEnsembles ()
void	SetEnsemble (string ensembleID)
void	SetEnsemble (Ensemble ensemble)
void	SetChiptuneMode (bool value)
bool	GetChiptuneMode ()
string	GetEmotionMode ()
void	SetEmotionMode (string mode)
float	GetEmotionalVelocity ()
void	SetEmotionalVelocity (float value)
string	GetEmotionString ()
Emotion	GetEmotion ()
void	SetEmotion (Emotion emotion)
void	SetEmotion (string emotion)
Vector2	GetVA ()
void	SetVA (Vector2 value)
int	AddEmotionalPoint (float x, float y, [MarshalAs(UnmanagedType.LPStr)] string emotion)
int	AddEmotionalPoint (float x, float y, Emotion emotion)
int	AddEmotionalPoint (EmotionalPoint ep)
void	RemoveEmotionalPoint (int id)
void	RemoveEmotionalPoint (EmotionalPoint ep)
void	SetEmotionalPointPosition (int id, float x, float y)
void	SetEmotionalPointPosition (EmotionalPoint ep)
void	SetEmotionAtPoint (int id, [MarshalAs(UnmanagedType.LPStr)] string emotion)
void	SetEmotionAtPoint (int id, Emotion emotion)
void	SetEmotionAtPoint (EmotionalPoint ep)
void	ClearEmotionalPoints ()
void	SetListenerPosition (float x, float y)
void	SetListenerPosition (MelodriveListener listener)
float	GetEmotionalStrength ()

void	SetEmotionalStrength (float value)
void	SetTempoScale (float value)
void	Play ()
void	Pause ()
void	Stop ()
void	SetCue (string cueName)
void	SetStateOptions (string style, string musicalSeed, string ensemble)
void	SetStateOptions (Style style, string musicalSeed, string ensemble)
void	SetStateOptions (State state)
void	SetMasterGain (float value)
void	SetLimiterEnabled (bool value)
float []	GetRMS ()

Public Attributes

const Style	DEFAULT_STYLE = Style.Piano
const Emotion	DEFAULT_EMOTION = Emotion.Neutral
bool	playOnStart = false
Style	initStyle = DEFAULT_STYLE
Emotion	initEmotion = DEFAULT_EMOTION
bool	chiptuneMode = false
bool	preloadStyles = false
EmotionalAxis	emotionalAxis = EmotionalAxis.XZ
const string	DLL_NAME = "AudioPlugin_Melodrive"

Events

DebugHandler	DebugLog
InitHandler	OnInit
NoteOnHandler	NoteOn
NoteOffHandler	NoteOff
ParamChangeHandler	ParamChange
BarHandler	Bar
BeatHandler	Beat
BeatTickHandler	BeatTick
TempoChangeHandler	TempoChange
NewMusicalSeedHandler	NewMusicalSeed
CueChangeHandler	CueChange
ProjectLoadHandler	ProjectLoad

Detailed Description

This is the **Melodrive** controller class. Other Game objects in Unity should control this class, or listen to the events fired.

Member Function Documentation

◆ AddEmotionalPoint() [1/3]

```
int
Melodrive.MelodrivePlugin.AddEmotionalPoint ( float          x,
                                                float          y,
                                                [MarshalAs(UnmanagedType.LPStr)] string emotion
                                                )
```

inline

Add an emotional point in game-space.

Parameters

- float** x - the x co-ord
- float** y - the y co-ord
- string** emotion - the emotion of the point e.g. "happy"

Returns

int - the id of the created point

◆ AddEmotionalPoint() [2/3]

```
int Melodrive.MelodrivePlugin.AddEmotionalPoint ( float    x,
                                                    float    y,
                                                    Emotion emotion
                                                    )
```

inline

Add an emotional point in game-space.

Parameters

float x - the x co-ord

float y - the y co-ord

Emotion emotion - the emotion of the point

Returns

int - the id of the created point

◆ AddEmotionalPoint() [3/3]

```
int Melodrive.MelodrivePlugin.AddEmotionalPoint ( EmotionalPoint ep )
```

inline

Add an emotional point in game-space.

Parameters

float x - the x co-ord

float y - the y co-ord

Emotion emotion - the emotion of the point

Returns

int - the id of the created point

◆ ClearEmotionalPoints()

```
void Melodrive.MelodrivePlugin.ClearEmotionalPoints ( )
```

inline

Cleares the emotional points from the scene

◆ CreateMusicalSeed()

```
void Melodrive.MelodrivePlugin.CreateMusicalSeed ( bool setActive = false )
```

inline

Creates a new Musical Seed. This will trigger a NewMusicalSeed event when complete.

Parameters

bool setActive - set to true to trigger a Musical Seed change

◆ GetChiptuneMode()

```
bool Melodrive.MelodrivePlugin.GetChiptuneMode ( )
```

inline

Returns the current state of Chiptune mode.

◆ GetCurrentEnsemble()

```
string Melodrive.MelodrivePlugin.GetCurrentEnsemble ( )
```

inline

Returns the currently active Ensemble ID

◆ GetCurrentEnsembleName()

```
string Melodrive.MelodrivePlugin.GetCurrentEnsembleName ( )
```

inline

Returns the currently active Ensemble name

◆ GetEmotion()

```
Emotion Melodrive.MelodrivePlugin.GetEmotion ( )
```

inline

Returns the current emotion as an enum

◆ GetEmotionalStrength()

`float Melodrive.MelodrivePlugin.GetEmotionalStrength ()`

inline

Returns the emotional strength, which is how strong emotional points affect the emotion.

◆ GetEmotionalVelocity()

`float Melodrive.MelodrivePlugin.GetEmotionalVelocity ()`

inline

Returns the emotional velocity, which is how fast emotion changes happen.

◆ GetEmotionMode()

`string Melodrive.MelodrivePlugin.GetEmotionMode ()`

inline

Returns the current emotion control "mode". Options are "positional" (default), "discrete" or "direct". The mode changes automatically when you use the SetVA or SetEmotion methods, but you have to change back to "positional" mode if you'd like to use EmotionalPoints again.

◆ GetEmotionString()

`string Melodrive.MelodrivePlugin.GetEmotionString ()`

inline

Gets the current emotion as a string

◆ GetEnsembleID() [1/2]

`string Melodrive.MelodrivePlugin.GetEnsembleID (int ensembleIndex)`

inline

Returns an Ensemble ID in the list.

◆ GetEnsembleID() [2/2]

`string Melodrive.MelodrivePlugin.GetEnsembleID (Ensemble ensemble)`

inline

Returns an Ensemble ID in the list.

◆ GetEnsembleName() [1/2]

`string Melodrive.MelodrivePlugin.GetEnsembleName (string ensembleID)`

inline

Returns and Ensemble name by ID.

◆ GetEnsembleName() [2/2]

`string Melodrive.MelodrivePlugin.GetEnsembleName (Ensemble ensemble)`

inline

Returns and Ensemble name by ID.

◆ GetEnsembles()

`Dictionary<string, string> Melodrive.MelodrivePlugin.GetEnsembles ()`

inline

Returns a Dictionary of available Ensembles in the current style.

◆ GetMusicalSeeds()

`string [] Melodrive.MelodrivePlugin.GetMusicalSeeds ()`

inline

Returns a list of the Musical Seeds.

◆ GetNumEnsembles()

`int Melodrive.MelodrivePlugin.GetNumEnsembles ()`

inline

Returns the number of ensembles in the loaded style

◆ GetStyle()

Style Melodrive.MelodrivePlugin.GetStyle ()

inline

Returns the currently playing Style.

◆ GetStyleString()

string Melodrive.MelodrivePlugin.GetStyleString (Style style)

inline

Returns the string value for style enum.

◆ GetVA()

Vector2 Melodrive.MelodrivePlugin.GetVA ()

inline

Returns the current Valence (x) Arousal (y) point

◆ Init()

```
void Melodrive.MelodrivePlugin.Init ( Style style = DEFAULT_STYLE,  
                                         Emotion initEmotion = DEFAULT_EMOTION  
                                         )
```

inline

Initialises **Melodrive** with a given style and emotion

Parameters

string style - the style

string initEmotion - the initial emotion

◆ LoadMusicalSeed()

`void Melodrive.MelodrivePlugin.LoadMusicalSeed (string filename)`

inline

Loads a Musical Seed into the current project.

◆ LoadProject()

`void Melodrive.MelodrivePlugin.LoadProject (string filename)`

inline

Loads a project from the specified path

◆ Pause()

`void Melodrive.MelodrivePlugin.Pause ()`

inline

Pauses playback

◆ Play()

`void Melodrive.MelodrivePlugin.Play ()`

inline

Starts playback

◆ PreloadStyles()

`void Melodrive.MelodrivePlugin.PreloadStyles ()`

inline

Preloads all styles to avoid audio artifacts

◆ RemoveEmotionalPoint() [1/2]

```
void Melodrive.MelodrivePlugin.RemoveEmotionalPoint ( int id )
```

inline

Remove an emotional point from **Melodrive**.

Parameters

int id - the id of the point to remove

◆ RemoveEmotionalPoint() [2 / 2]

```
void Melodrive.MelodrivePlugin.RemoveEmotionalPoint ( EmotionalPoint ep )
```

inline

Remove an emotional point from **Melodrive**.

Parameters

int id - the id of the point to remove

◆ SaveMusicalSeed()

```
void Melodrive.MelodrivePlugin.SaveMusicalSeed ( string seedName,  
                                                    string filename  
                                                    )
```

inline

Saves a Given Musical Seed to a file for later loading.

◆ SaveProject()

```
void Melodrive.MelodrivePlugin.SaveProject ( string filename )
```

inline

Saves a project to the specified path

◆ SetChiptuneMode()

`void Melodrive.MelodrivePlugin.SetChiptuneMode (bool value)`

inline

Activates/deactivates **Melodrive**'s Chiptune mode. While active, all the ensembles will be Chiptune until deactivated. Note Chiptune mode is deactivated if SetEnsemble is called with a non-chiptune ensemble.

◆ SetCue()

`void Melodrive.MelodrivePlugin.SetCue (string cueName)`

inline

Activates a Cue in **Melodrive**

◆ SetEmotion() [1/2]

`void Melodrive.MelodrivePlugin.SetEmotion (Emotion emotion)`

inline

Sets the current emotion as an Emotion enum. This will also set the emotion mode to "discrete".

◆ SetEmotion() [2/2]

`void Melodrive.MelodrivePlugin.SetEmotion (string emotion)`

inline

Sets the current emotion as a string e.g. "happy" "sad". This will also set the emotion mode to "discrete".

◆ SetEmotionalPointPosition() [1/2]

```
void Melodrive.MelodrivePlugin.SetEmotionalPointPosition ( int    id,  
                                                             float x,  
                                                             float y  
                                                             )
```

inline

Update an emotional point's position in game-space.4

Parameters

int id - Melodrive's ID for the point

float x - the x co-ord

float y - the y co-ord

◆ SetEmotionalPointPosition() [2/2]

```
void Melodrive.MelodrivePlugin.SetEmotionalPointPosition ( EmotionalPoint ep )
```

inline

Update an emotional point's position in game-space.4

Parameters

int id - Melodrive's ID for the point

float x - the x co-ord

float y - the y co-ord

◆ SetEmotionalStrength()

```
void Melodrive.MelodrivePlugin.SetEmotionalStrength ( float value )
```

inline

Sets the emotional strength, which is how strong emotional points affect the emotion.

◆ SetEmotionalVelocity()

```
void Melodrive.MelodrivePlugin.SetEmotionalVelocity ( float value )
```

inline

Sets the emotional velocity, which is how fast emotion changes happen.

◆ SetEmotionAtPoint() [1/3]

```
void  
Melodrive.MelodrivePlugin.SetEmotionAtPoint ( int id,  
                                                [MarshalAs(UnmanagedType.LPStr)] string emotion  
                                                )
```

inline

Updates an emotional point's emotion

Parameters

int id - **Melodrive's** ID for the point
string emotion - the new emotion

◆ SetEmotionAtPoint() [2/3]

```
void Melodrive.MelodrivePlugin.SetEmotionAtPoint ( int id,  
                                                    Emotion emotion  
                                                    )
```

inline

Updates an emotional points emotion

Parameters

int id - **Melodrive's** ID for the point
Emotion emotion - the new emotion

◆ SetEmotionAtPoint() [3/3]

```
void Melodrive.MelodrivePlugin.SetEmotionAtPoint ( EmotionalPoint ep )
```

inline

Updates an emotional points emotion

Parameters

int id - **Melodrive's** ID for the point
Emotion emotion - the new emotion

◆ SetEmotionMode()

```
void Melodrive.MelodrivePlugin.SetEmotionMode ( string mode )
```

inline

Sets the current emotion mode. Options: "positional" - will listen to the listener position and the emotional points to determine emotion "discrete" - set when using SetEmotion. Emotion values map to discrete points "direct" - set when using SetVA. Allows direct control over the VA space

◆ SetEnsemble() [1/2]

```
void Melodrive.MelodrivePlugin.SetEnsemble ( string ensembleID )
```

inline

Sets the active Ensemble by ID. This will Trigger a CueChange.

◆ SetEnsemble() [2/2]

```
void Melodrive.MelodrivePlugin.SetEnsemble ( Ensemble ensemble )
```

inline

Sets the active Ensemble by ID. This will Trigger a CueChange.

◆ SetInstrumentsPath()

```
void Melodrive.MelodrivePlugin.SetInstrumentsPath ( string path )
```

inline

Sets the **Melodrive** AudioPlugin to load instruments from the specified folder

◆ SetInstrumentsPathToStreamingAssets()

```
void Melodrive.MelodrivePlugin.SetInstrumentsPathToStreamingAssets ( )
```

inline

Sets the **Melodrive** AudioPlugin to load instruments from the StreamingAssets folder

◆ SetLimiterEnabled()

```
void Melodrive.MelodrivePlugin.SetLimiterEnabled ( bool value )
```

inline

Turns the limiter on/off

◆ SetListenerPosition() [1/2]

```
void Melodrive.MelodrivePlugin.SetListenerPosition ( float x,  
                                                    float y  
                                                    )
```

inline

Updates the position of the listener in world space.

Parameters

float x

float y

◆ SetListenerPosition() [2/2]

```
void Melodrive.MelodrivePlugin.SetListenerPosition ( MelodriveListener listener )
```

inline

Updates the position of the listener in world space.

Parameters

float x

float y

◆ SetMasterGain()

```
void Melodrive.MelodrivePlugin.SetMasterGain ( float value )
```

inline

Sets the gain of the Master bus

◆ SetMusicalSeed()


```
void Melodrive.MelodrivePlugin.SetMusicalSeed ( string seedName )
```

inline

Sets the Musical Seed in **Melodrive**. This will trigger a CueChange event.

◆ SetStateOptions() [1/3]

```
void Melodrive.MelodrivePlugin.SetStateOptions ( string style,  
                                                  string musicalSeed,  
                                                  string ensemble  
                                                  )
```

inline

Sets the playback state in **Melodrive**

◆ SetStateOptions() [2/3]

```
void Melodrive.MelodrivePlugin.SetStateOptions ( Style style,  
                                                  string musicalSeed,  
                                                  string ensemble  
                                                  )
```

inline

Sets the playback state in **Melodrive**

◆ SetStateOptions() [3/3]

```
void Melodrive.MelodrivePlugin.SetStateOptions ( State state )
```

inline

Sets the playback state in **Melodrive**

◆ SetStyle() [1/2]

```
void Melodrive.MelodrivePlugin.SetStyle ( string style )
```

inline

Sets the current style

◆ SetStyle() [2/2]

```
void Melodrive.MelodrivePlugin.SetStyle ( Style style )
```

inline

Sets the current style

◆ SetTempoScale()

```
void Melodrive.MelodrivePlugin.SetTempoScale ( float value )
```

inline

Sets the tempo "scale", which is a multiplier on the base tempo that **Melodrive** chooses.

◆ SetVA()

```
void Melodrive.MelodrivePlugin.SetVA ( Vector2 value )
```

inline

Set the Valence/Arousal point. This will also set the emotion mode to "direct".

◆ Stop()

```
void Melodrive.MelodrivePlugin.Stop ( )
```

inline

Stops playback

The documentation for this class was generated from the following file:

- unity/MelodrivePackage/Assets/Melodrive/Scripts/MelodrivePlugin.cs