2015-09-24

The ELIAS Unity Plugin - Version 1.5

User's Guide

ELIAS Software AB

Table of Contents

Overview

Thank you for downloading the ELIAS Unity plugin!
ELIAS, Elastic Lightweight Integrated Audio System, is a music engine which attempts to change the way we perceive and experience interactive music in video games today.
With the ELIAS music engine, composers and game developers can easily collaborate to create musical scores that adapt based on the player's actions in the virtual world.
With the ELIAS engine, the music is just as dynamic and flexible as the corresponding composition would have been in a movie.

This package contains the following:
C# wrapper code to simplify usage of ELIAS within Unity. (Theme.cs, Track.cs and Elias.cs)
A DemoPlayer made to handle playback of ELIAS themes. (We suggest that you use this even outside our demo scene).
A Demo scene and associated scripts to provide an example of how ELIAS can be used and to test themes.
Native libraries for Mac, Windows, iOS and Android. (These are not the same as the ones in the ELIAS SDK downloadable from our website, as these have more functionality to simplify using ELIAS in Unity. (Read more in the chapter The ELIAS library files).
An example theme that can be used to test and demonstrate ELIAS, but may not be distributed. (How to get a license may be found in the chapter "Licensing"). If you need libraries for other platforms please contact us at support@eliassoftware.com.

For bug reporting please email us at bugs@eliassoftware.com, and other support questions should be directed to support@eliassoftware.com.
For licensing or other questions please email sales@eliassoftware.com.
If you have a question and you're not sure where to send it, don't worry, we will handle it even if it comes to the wrong place!
More themes and commercial licenses can be purchased at our website www.eliassoftware.com/shop

Changes

Version 1.5

- Added StartTheme functions so the theme can now be started from other scripts. Either with the pre-set parameters or sent in parameters.
- Added a StopTheme function to stop the theme from other scripts without having to destroy the component.
- Fixed a bug with the OS X elias.bundles.
- Minor documentation fixes.

Version 1.4 (Matches ELIAS SDK 1.7)

- Fixed a potential bug if using non ASCII characters, now all strings passed from the wrapper are guaranteed to be UTF-8. WARNING: This breaks Elias.Source.filename usage, it was previously a string, and is now an IntPtr. To get the string version use Elias.UTF8MarshallingHelpers.ConvertFromNativeUTF8(sourceObject.filename) .

- Added support for Action Presets. Both for creating and modifying them from code, and for calling them via the DemoPlayer.
- Action Presets can now be set in the Inspector to be used when the theme is started.
- Added the ability to not start the theme on Start. (This can be set from the inspector).
- Added the ability to set more starting parameters. The parameters:
    - Key (Always applied unless an action preset is used that overrides it)
    - Action Preset (Excludes the following)
    - Min Level
    - Max Level
    - Stinger
    - Rendezvous
    - (Already existing) Urgency
- Renamed DemoPlayer.urgency to DemoPlayer.startUrgency as that is how it is used in the DemoPlayer.
- Added Android support.
- Updated DemoPlayer code with support for reading the sound files from within the .apk file.
- Added interface code LoadSourcesFromApk(string theme_URI, Decoder_Flags flags) to make it easy to load sources from within an .apk file. Use this function when targeting an android device or emulator.
- Added interface code LoadSourcesFromArchive(string pathToArchive, string pathToThemeDirectory, Decoder_Flags flags) to handle when sound files are within an archive. PathToThemeDirectory is relative within the archive.
- Removed RegisterNotifications as it could potentially be causing crashes! (Due to being called from a different thread.)
- Getting PrebufferLengthMS now works properly.
- Added support to use Decoder_Flags when loading sources. (Options are: decoder_default, decoder_stream, decoder_preload and decoder_verify). The Decoder_Flags enum is located in Elias.cs
- Added StingerProgressions.Shuffle
- Demo player can now output a different audio configuration than what the theme uses. SetOutputChannels(int channels), and GetOutputChannels, both of these can be used while the theme is running.
- ELIAS will now handle a different sample rate compared to what the theme uses.

SetOutputSampleRate(int sampleRate) and GetOutputSampleRate. Both of these can be used while the theme is running.

- DemoPlayer now listens to AudioSettings.OnAudioConfigurationChanged and will change ELIAS's output samerate and output channels whenever it is called.
- DemoPlayer now uses a different way of setting the Prebuffer and ProcessingPeriod from the inspector. This utilizes a multiple of Unity's DSP buffer, (settable in the projects AudioSettings), to provide minimal latency.
- Better error message when the DemoPlayer fails to start a theme.
- DemoPlayer now disables itself if something goes wrong, this is to prevent errors from constantly popping up.
- The "ExitOnPlayModeScriptCompile.cs" now only exits in play mode if ELIAS is using the streamed AudioClip solution.
- The DemoPlayer now only shows the "Pending Trigger Change" box if enabled from the inspector.

Version 1.3.1

- Fixed a major delay flaw with our implementation of sound playback. This removes approximately 600ms of latency from ELIAS!
  We now rely on the monobehavior of the DemoPlayer being a "post process" to the audio. This is done by implementing the function "OnAudioFilterRead" and not having any AudioClip on the AudioSource.
- If the output is not using the same number of channels as the theme is, we currently use a "streamed" audioClip and ReadSamplesBlocking to allow Unity to handle the problem which results in additional latency. This will be fixed in a later version!
- Changed from being able to set prebuffer and processingperiod in milliseconds to a simplified 4 option dropdown. Default is "Low Latency." When streaming a Wave (uncompressed) theme you will most likely get crackling due to limited disk I/O speed, but this should not be an issue when using Ogg Vorbis.
- Added an internal resampler to the engine, allowing it to deliver its output at any sample rate. Use SetOutputSampleRate(long sampleRate) to set it before starting the theme.
- Added ReadFloatSamplesBlocking which waits for the specified number of samples to be available before returning. This blocks the thread from which the function is invoked.

Version 1.3

- Changed the file structure so now all ELIAS scripts and demo assets are located inside the ELIAS folder.
- Added ExitPlayModeOnScriptCompile.cs, this will stop play mode if you attempt to do a hot-reload, as it may otherwise crash the editor.

- Fixed a bug where sounds that had “play on awake” would not play when the scene started.
- Fixed a bug where the clip length in the demo player was 1 second, instead of a more reasonable amount. It is now prebuffer length + processing period to reduce potential delay and memory usage.
- Fixed a bug with the demo player editor where if a folder in the Assets/StreamingAssets/ELIAS_Themes folder had a file with the same name, but was not an .epro file, it would still list it as a possible theme.
- Minor clean-up of the demo code.

Version 1.2

- Using version 1.6 of the ELIAS Engine. See section “Version History” in the supplied “manual.html” for details.

Version 1.0.1

- Minor fix of initial release.

Version 1.1

- Added a visual interface which simplifies the process of importing and configuring themes.
- Added an example scene which demonstrates how ELIAS may be integrated into a game.
- Fixed a bug in the binding that caused problems on Mac.
- Fixed some resource leaks.

Supported Platforms

Microsoft Windows
Mac OS X
Android: ArmV7 and x86, the same platforms that Unity currently supports and we have the plugins for more on our homepage.
iOS: Armv7, Armv7s, Arm64, x86 simulator, x86_64 simulator.

Supported Audio Formats

The plugin supports audio files in Wave format and Ogg Vorbis format. The supported sample bit depths are 16, 24 and 32.

Installation

The plugin is delivered as a unitypackage-file. Download it from our web site at www.eliassoftware.com/product/unity-plugin-all-platforms/ and save the file in a convenient place. In the Unity editor, create a new project and choose
Assets->Import Package->Custom Package...,
and choose the unitypackage-file.

Note that the inclusion of Unity's standard assets is only required for using the Demo Scene.

The operating system and architecture specific settings with respect to the ELIAS library file should already be correctly configured when you have imported the Unity package, but in case of any issue

you might want to check that they are okay. The following two sections give information on plugin settings for the native binary.

The ELIAS library files.

In the instructions below, the engine shared, (for non iOS, iOS uses a static), library file which you will use is referred to as "the ELIAS file". There are several different versions, and the one you should choose depends on your (target/working) operating system and architecture.
The plugin file is a shared library which means that on Microsoft Windows it is called elias.dll, on Mac OS X it's called elias.bundle, for Android it's called elias.so and on iOS it's called libelias.a

For both MS Windows and Mac OS X, the ELIAS file comes in both a 64- and 32-bit version.
For iOS it comes as a single “FAT” library where it contains armv7, armv7s, arm64, i386 (x86) and x86_64 architectures. (This can be split apart again if you wish to minimize the release size of your product using the lipo command line tool.)
For Android it comes as two libraries, however because **Unity3D currently only supports armeabi-v7 and x86**, so we are only shipping these with the Unity package, all of them are downloadable from our website.)
When you follow the instructions below, choose the version of the library that is applicable to your operating system and Unity editor version.

Unity Settings for the ELIAS library file

Navigate to the “Assets/Plugins” subfolder depending on what library you’re looking for in the Unity Project view, click on "the ELIAS file" (see the description in the previous section) and set the corresponding properties in the Import Settings in Inspector view.
Under "Select platforms for plugin", check "Standalone".
If you use the 32-bit version of the Unity editor, check "Editor" for elias.dll in the x86 folder.
If you use the 64-bit version of the Unity editor, check "Editor" for elias.dll/elias.bundle in the x86_64 folder.

Under "Platform settings", choose the correct CPU and OS. Use a similar approach if you need to set or change the other libraries manually.

Package Contents by Folder

In the description below, the folder names which are specific to ELIAS will be marked with (ELIAS).

Editor and StandardAssets

These are standard Unity asset folders and not part of the ELIAS integration. Here is for example the Unity 5 first person character.
They are included in this package just so we could provide a working test scene.

StreamingAssets (ELIAS)

This folder contains a subdirectory named **ELIAS_Themes**.
(**The following in this paragraph only applies if you’re using our DemoPlayer**)
In ELIAS_Themes you can drop one or several theme folders and the themes should then be selectable in the Unity interface via the ELIAS_Component GameObject. Note that the name of the theme folder must be the same as the name of the theme definition file (but without the .epro suffix)

in order for the Unity interface to find it and make it available as a suggestion in the drop down menu.
Example path to an ELIAS theme file: Assets/StreamingAssets/ELIAS_Themes/demo/demo.epro.

Important note: Any themes that you do not want to ship with your game **must be removed** from the StreamingAssets folder and any of it's subfolders. This also applies to the Demo Theme that we send along. If you wish to use it see the licensing part at the bottom.

More on this under the section about the ELIAS_Component below.

elias (ELIAS)

Here are the C# binding scripts to the ELIAS native library, together with an example player demonstrating how to interact with the ELIAS music engine (e.g. loading an ELIAS adaptive theme, starting and stopping the engine, issuing a trigger level change or playing a stinger).

Unity C# Scripts

In this section an overview of the role and functionality of the scripts related to ELIAS is given. The scripts have been put into two folders to make a distinction between core binding and application scripts, but the user is of course free to create their own structure.

elias/Elias.cs

ELIAS enumerations for e.g. track types and error codes, and a structure definition for what a “Source” is.

elias/Theme.cs

Wrapper around the native library, it uses exceptions and does the error checking for you. If you wish to bypass it you can make the ***extern*** functions public instead of private.

elias/Track.cs

Wrapper around the native library, it uses exceptions and does the error checking for you. If you wish to bypass it you can make the ***extern*** functions public instead of private.

elias/Editor/ExitOnPlayModeScriptCompile.cs

This is a solution for a bug where the Unity editor will crash if a hot reload is performed and ELIAS is utilizing a streamed AudioClip. It forces the editor to exit play mode before a compile takes place if ELIAS is using a streamed AudioClip.

Platform specifics

iOS

If you encounter link errors, please try the following:

Go to the project `Build Settings` tab

- Search for `Other Linker Flags`
- Add `-lz` as a linker flag (see screenshot)
- In the `Product` menu, select `Clean` before building again

Callbacks work differently on iOS, to be allowed to call a managed function from unmanaged code the function must have the Attribute **AOT.MonoPInvokeCallback** to a static function, ex: **[AOT.MonoPInvokeCallback(typeof(Elias.Theme.SourceEnumerationCallback))]** . Mono's information: http://developer.xamarin.com/guides/ios/advanced_topics/limitations/#Reverse_Callbacks

Due to some issues with lagging on mobile devices the processing period will be equal to the prebuffer if using an unmodified DemoPlayer.

The library file used for iOS contains all different configurations you might need. You may want to split it apart and put only the versions you need together. This is due to a limitation with Unity.

The library file used for iOS is a lot larger on disk before creating the end product then it will be when the end product is created. This is due to symbol hiding/stripping.

Android

On Android, all data will be packed into either the .apk file, or the .obb (in case of split binary). The ELIAS Unity plugin handles this by using LoadSourcesFromApk instead of normal **LoadSources** and sending in the URI that Unity is supposed to provide with Application.streamingAssetsPath. This will also work for files packed into the .obb files. In case you are getting the path in a different manner, you can use LoadSourcesFromArchive. **LoadSourcesFromArchive** takes 2 arguments, the first is the path to the archive itself, including the archive name, and the second argument is the path to the theme folder from “inside” the archive.

Due to some issues with lagging on mobile devices the processing period will be equal to the prebuffer if using an unmodified DemoPlayer.

Demo

Materials and Textures

These are customary folders with expected content for a Unity developer. The floor texture was acquired from the Pixar One Twenty Eight library at https://community.renderman.pixar.com/article/114/library-pixar-one-twenty-eight.html.

Prefabs (ELIAS)

This folder contains prefabs exemplifying how a Unity GameObject can be used for triggering an ELIAS level change or stinger. How the ELIAS engine is controlled through its API is of course entirely up to you in the end; this is just an example. For a complete description of the ELIAS API, see the SDK manual in Assets/elias/doc. (Note that all the functions are wrapped in a C# version, so once you find the function you want, check in Theme.cs, Track.cs or Elias.cs to find the C# wrapper.)

elias/Demo/DemoPlayer.cs

This class is an implementation to play ELIAS themes, it will work just as well outside of the demo scene, and we recommend that you use it.

elias/Demo/Scripts/ELIASMusicSwitcher.cs

Implements some very simple logic for controlling trigger levels and which stingers to play. The Switch method listens to zone trigger events in the ExampleTrigger script, and the ThemeLoaded method is executed upon successful loading of a theme.

elias/Demo/Scripts/ExampleTrigger.cs

This script is attached to the trigger geometry in the scene. When the player character causes a trigger event to fire by intersecting the collision geometry, the subscribed methods will be called (e.g Switch in MusicSwitcher).

elias/Demo/Scripts/ThemeGeometry.cs

Creates geometry for visualisation and collision detection. The ExampleTrigger script is attached to the trigger geometry in the scene. The script also defines simple movement of the geometry via its Update method.

ELIAS Unity GameObjects

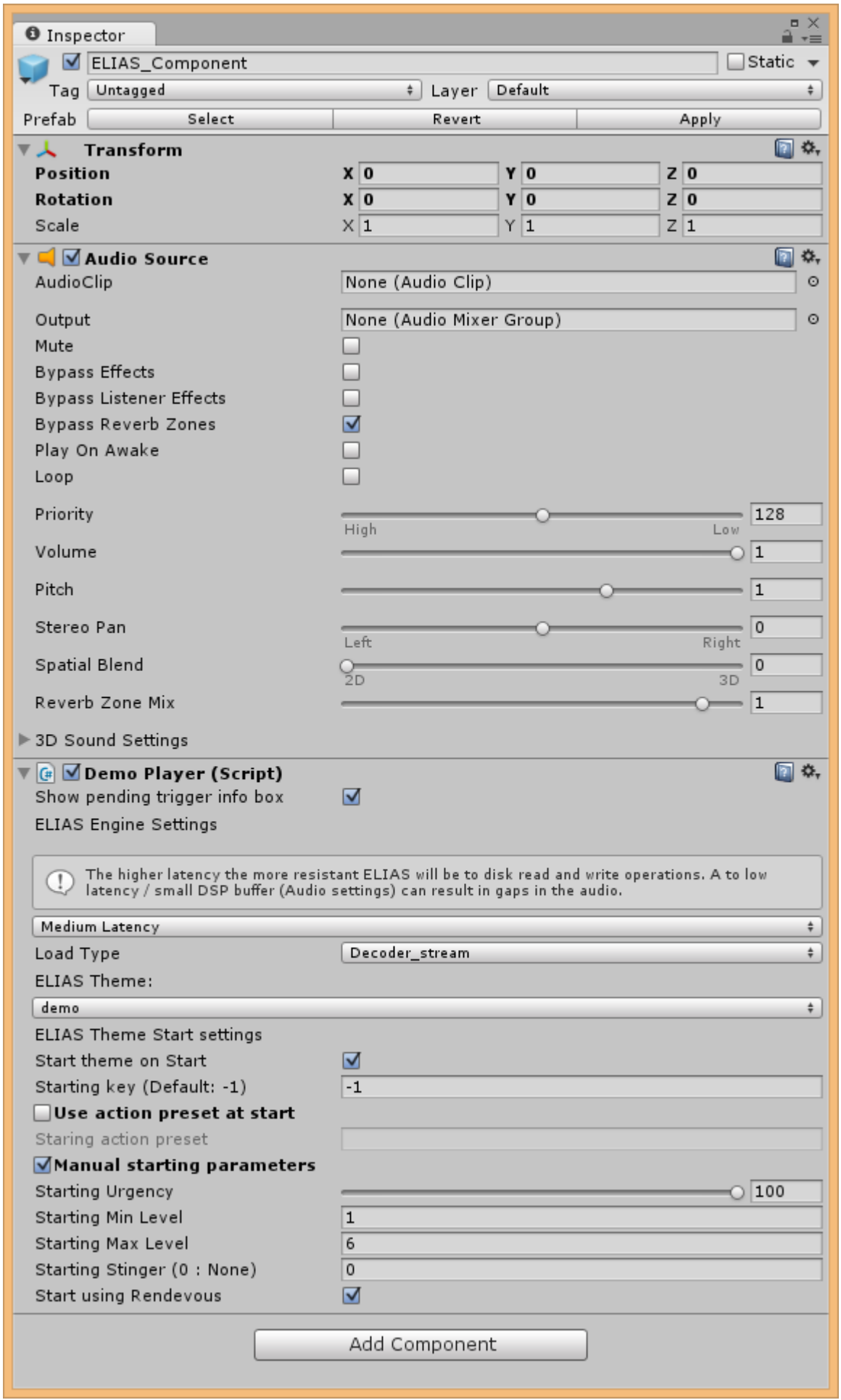
The example scene contains a few GameObjects which demonstrate how ELIAS can be used with Unity.

MusicSwitcher

This GameObject carries an example script demonstrating how a Unity application can control the ELIAS engine.

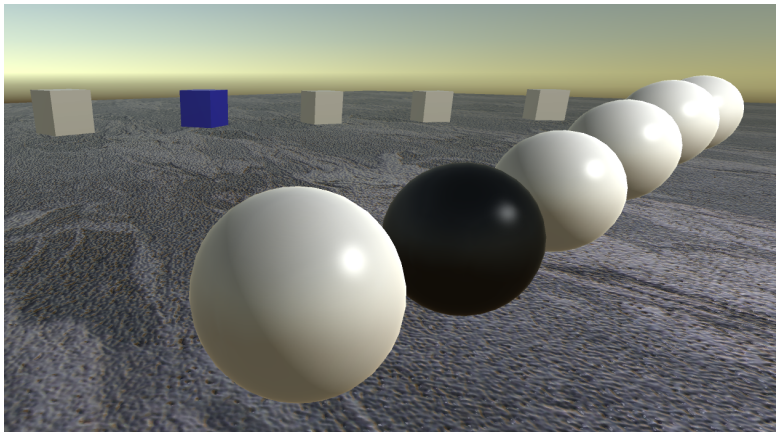
ELIAS_Component

This GameObject carries an AudioSource component for audio playback and the DemoPlayer script for interfacing with ELIAS at the highest level.



About the Demo Scene

Spheres and boxes will be generated once the simulation has started. The spheres trigger stingers, and the boxes trigger level changes in an objective theme. These are activated once you step into them.



The capsule in the sign turns green when a theme is loaded, which will generally be almost instantly after "play" has been pressed.

The sign with its curved arrow suggests a starting point and direction when visiting the sequence of boxes. They are generated in a counter clockwise fashion with the lowest theme level trigger box at the start. If the theme has subtle changes between levels you can skip a few boxes at a time for faster progression through the theme.

Current Limitations

The DemoPlayer is currently slightly limited in how it works, and can be extended or replaced if necessary. For most users this should not be required.

The ELIAS Unity plugin can currently only read its audio files from disk separate from how Unity handles files. This is the reason for the requirement to place the files in the Streaming Assets folder, the downside of this being that the audio compression settings that Unity otherwise allows will have to be done outside of Unity. Elias Composer's Studio has a menu option called “Export delivery package” which allows the user to export all the audio that is used in a theme as Ogg Vorbis or Wave, (for a shipped game we strongly recommend that you use Ogg Vorbis!).

The test scene only shows simple usage of ELIAS, there is much more that you can do! The actual binding to the ELIAS library offers a much wider set of low level features, so reading the elias manual and experimenting yourself is encouraged if you want to do things that the provided DemoPlayer can't. Note that it will generally be easier to make modifications to a theme or its settings using the ELIAS Composer's Studio, compared to doing the same in code.
All of the functions in the C reference have corresponding counterparts in the Unity plugin (see Elias.cs, Track.cs and Theme.cs for more information).

Further documentation

In the folder “Assets/elias/doc” you can find the accompanying manual for the standalone ELIAS

SDK. It contains a thorough description of all the core concepts in the ELIAS music engine, as well as the C API reference. As a Unity plugin user you do not need to worry about the data callback mentioned in the C reference.

All of the functions in the C reference have corresponding counterparts in the Unity plugin (see Elias.cs, Track.cs and Theme.cs for more information). These wrapper also handle string encoding conversion to and from UTF-8 that ELIAS uses.

Licensing

Demo Theme

The package includes an adaptive theme that is used as part of the example scene to demonstrate the functionality of the ELIAS engine. This theme may not be used for any purpose, commercial or otherwise, except as part of the example scene when evaluating ELIAS. However, it may be licensed for production use from our website. The title of the theme on the website is "Beneath Fog City".

ELIAS and third-party licenses

To learn more about the ELIAS music engine and its licensing terms, as well as the authoring tool for composers, visit www.eliassoftware.com/elias-engine.

Before distributing ELIAS as part of a game, see the file "copying.txt" which is found in the “Assets/elias/doc” directory.

Music, Composer's Studio and the web site

If you need music to go along with the ELIAS engine in your next title, please visit http://www.eliassoftware.com/shop/

We offer a high quality comprehensive library of ready to use adaptive themes that you can drop right into your production.

We also develop and distribute an authoring tool for creating your own adaptive themes, modifying themes and previewing themes, the ELIAS Composer's studio.

We hope you find the ELIAS engine useful, and we welcome your feedback. support@eliassoftware.com.

Have fun!