FMOD Studio Unity Integration

Introduction

The FMOD Studio Unity integration provides a streamlined workflow for Unity developers to create high quality game audio for their games. The integration is provided as a .unitypackage available from the FMOD downloads page. This document explains the various components of the integration. For more information visit the FMOD Studio Unity Integration forum.

Supported platforms

The current integration supports iOS, Android, Windows Phone, PC, Mac, Linux, PS4, WiiU, and XBox One. The Unity webplayer is not supported, as it does not provide the native plugin support that the integration requires.

Licensing

The integration itself is free, but you must have the appropriate FMOD License to release a title using FMOD Studio with Unity, as it is not covered by the Unity license alone. For more information about licensing FMOD see the FMOD sales page. The FMOD Studio integration is a native plugin which is available in the free iOS/Android Unity license but requires Pro for PC and Mac.

Contents

This section describes the contents of the integration package.

Example Scripts

Found in Assets/Plugins/FMOD

The integration package includes scripts which demonstrate usage of the FMOD Studio API. you can use these scripts and extend them, or write your own using the C# API directly. A brief description of each of the example scripts is listed below.

FMOD_Listener

Attach this to the camera in the scene. (there should only be one listener)

FMOD_StudioEventEmitter

Controls the playback of one event, use this to attach events to objects inside Unity. This is most suited to endless events which play for a long time. For short oneshot style events they can be fired off from scripts using the FMOD_StudioSystem.

FMOD_StudioSystem

Note: This is internal, don't attach it to any objects This is the central system which can be used to load banks and retrieve events. It has some convenience functions that demonstrate usage of the FMOD Studio API. It is a singleton so there is only ever one of these, it will get created automatically, you can access it by calling FMOD_StudioSystem.instance.

FMOD Studio Runtime C# API

Found in Assets/Plugins/FMOD/Wrapper

This is the FMOD Studio C# API. The wrapper exposes the same functions available in the C++ API for use inside Unity. For more information please refer to the programmer API documenation that comes with the FMOD Studio API installer.

FMOD Studio API usage

The content created in FMOD Studio is exported into .bank files. These files contain the encoded audio data, as well as the metadata for playback. They are loaded as loose files using the API (System.loadBankFile). The events are retrieved using GUIDs (System.parseID) or paths (System.lookupID). These functions provide an ID which can be used to retrieve the EventDescription, which is then used to create EventInstances. Event playback is controlled using the EventInstance.

Setting up the integration

1. Import the integration unitypackage.
2. Add the FMOD_Listener script to the Main Camera.

Importing .bank files into a Unity Project For the First Time

Inside FMOD Studio:

1. Assign your Events to banks.
2. Build all banks. (File > Build...)

Inside Unity:

3. Select FMOD > Import Bank and then select the /Builds folder inside your FMOD Studio project

The Import Bank step will copy the .bank files from the FMOD Studio project to the Unity project's Assets/StreamingAssets directory. Then it will create the Assets/FMODAssets directory and populate it with event assets for each event in the FMOD Studio project.

Updating .bank files in Unity after building in FMOD Studio

1. Select FMOD > Refresh Event List to re-import your banks into Unity.

Playing an event inside Unity

1. Add FMOD_StudioEventEmitter script to an object in the scene.
2. Locate the event you want to play inside Assets/FMODAssets and assign it to the FMOD_StudioEventEmitter.

Scripting

FMOD_StudioSystem contains some convenience functions which make it easy to control events from your Unity scripts.

The following examples demonstrate using the API with C# but are applicable to the other scripting languages available in Unity.

PlayOneShot

"fire and forget" oneshot events. Good for short sounds like footsteps and gunshots

Example: play a gun shot sound at the current position

```
FMOD_StudioSystem.instance.PlayOneShot(
  "event:/Weapons/Single-Shot",
  transform.position);
```

GetEvent

Retrieve an event this can be used when you want more control, such as modifying the game parameters.

Example: get an engine event and update the RPM parameter each frame

```
FMOD.Studio.EventInstance engine;
FMOD.Studio.ParameterInstance engineRPM;
void Start()
{
  engine = FMOD_StudioSystem.instance.getEvent("event:/Vehicles/Car Engine");
  engine.start();
  engine.getParameter("RPM", out engineRPM);
}
void Update()
{
  // get a RPM value from the game's car engine
  engineRPM.setValue(rpm);
}
void OnDisable()
{
  engine.stop();
  engine.release();
}
```

Note: When controlling the event instance in code, you must call release when you are finished with it to ensure the resources are cleaned up.

Loading Plugin Effects and Plugin Instruments

The FMOD Plugin Effects like Distance Filter and Gain are free, the third party effects like the ML1 Limiter and AudioWeather have to be licensed separately.There is licensing information available on the FMOD sales page.

To load the plugins using Unity you add the plugin to the FMOD_Listener's plugin list. Then just copy the plugin file into appropriate subdirectory of the Assets/Plugins directory (inside your Unity project) for the platform you are targeting. For Windows 32bit it would be /Assets/Plugins/x86. The directory for each platform is listed here.

To get the free FMOD plugins, you can copy the files inside the /Plugins directory where FMOD Studio is installed.

Configuration Options

There are options features which can be enabled which can help during development. These options are useful during development. To enable these add the specified keyword to the Scripting Define Symbols, inside Unity Player Settings (Edit > Project Settings > Player)

Connecting to the tool with Live Update

Option: FMOD_LIVEUPDATE

Live Update allows you to make changes inside FMOD Studio have have them mirrored on the game side. This is particularly useful for mixing and tweaking attenuation curves.

1. Add FMOD_LIVEUPDATE to the Scripting Define Symbols in your Unity project.

This will enable Studio to connect to Unity while the game is running.

2. Start running the game, either by building and executing on the target device, or by running the game inside the Unity editor.

3. Inside FMOD Studio open up the Connect to Game dialog (File > Connect to Game...)

4. Enter the IP address of the target device which is running game (if it is running on the same machine as FMOD Studio you just enter localhost or 127.0.0.1)

Studio should now be connected to the game and modifications made in Studio will be mirrored as long as the game is running.

Note: When the game stops all of the changes made via Live Update will be lost, at that point it is advised to rebuild the banks and import them into Unity to keep everything in sync.

Additional Debug information

Option: FMOD_DEBUG

This will print additional information to the Unity console to help diagnose issues if something has gone wrong with the integration.