Audio Core Documentation

This core uses the third-party library called FMOD, which can be downloaded from <http://www.fmod.org/> under the download section.

As of this write-up, the FMOD Ex version used is 4.44.07.

Once downloaded, install FMOD under default settings; this will default its directory to **C:\Program Files (x86)\FMOD SoundSystem**

After installation, has completed, open Visual Studio and create an empty Win32 console project.

Right-click on the project node in Solution Explorer, select Properties, choose VC++ Directories in the left-hand pane and add the following directory paths:

* Include Directories: **C:\Program Files (x86)\FMOD SoundSystem\FMOD Programmers API Windows\api\inc**
* Library Directories: **C:\Program Files (x86)\FMOD SoundSystem\FMOD Programmers API Windows\api\lib**

In the Linker -> Input section, add **fmodex\_vc.lib** to the Additional Dependencies.

When the application runs, fmodex.dll needs to be in the same folder as the exe file. For simplicity, you can add a post-build event to copy this automatically from the FMOD API directory to the target folder when the build succeeds.

Go to Build Events -> Post-build Event and set the Command Line as follows:

**copy /y “C:\Program Files (x86)\FMOD SoundSystem\FMOD Programmers API Windows\api\fmodex.dll” “$(OutputPath)”**

File to include: “CoreAudio.h”

class DevSoundManager

* The main hub for sound object interaction (playing, stopping, creating)

Methods:

CreateSound()

* Return type: Int
* Parameters: String
* Prerequisite: Must set up a fileName
* Purpose: Initialization for a specific sound that will not be looped using a string that represents the name of the file

CreateLoopedSound()

* Return type: Int
* Parameters: String
* Prerequisite: Must set up a fileName
* Purpose: Initialization for a specific sound that will be looped using a string that represents the name of the file

PlaySound()

* Return type: Void
* Parameters: Int, Int
* Prerequisite: Must set up a soundIndex and channelIndex
* Purpose: Plays a sound in a given channelIndex after a sound has been created and given a soundIndex

StopSound()

* Return type: Void
* Parameters: Int
* Prerequisite: Must set up a channelIndex
* Purpose: Stops a sound from playing in a given channelIndex

StopAllSounds()

* Return type: Void
* Parameters: none
* Prerequisite: none
* Purpose: Stops all sounds currently playing in any active channels

GetSoundLength()

* Return type: Float
* Parameters: Int
* Prerequisite: Must set up a soundIndex
* Purpose: Gets the length of a sound in seconds in a given soundIndex