Adding audio to your game engine

The following steps are required in order to add FMOD5 audio support to your game engine.

**Note**: **ALL** of your game projects (Incursion, Starship, all Protogames, and even MathUnitTests) are affected by this!

1. **Get to a good stopping point**.
   * Wait until you have Incursion building and running, and with no files checked out.
   * Consider adding Audio as one of the last features of your Incursion: Playable milestone.
2. **Unzip and copy files** from all 4 folders within this .ZIP to their appropriate locations:

**files for Engine\_Code\_Engine\_Audio**

* + - **AudioSystem.cpp**/**hpp** – the engine AudioSystem itself (simple, thin wrapper around FMOD5)
    - Copy these files to SD/Engine/Code/**Engine/Audio**

**files for Engine\_Code\_ThirdParty\_fmod**

* + - Includes all required header files (#include fmod.hpp) and static .LIBs for FMOD5
    - Copy these files to SD/Engine/Code/**ThirdParty/fmod**

**files for GameName\_Code\_Game (for ALL game projects)**

* + - **ALL** game projects must now have **Code/Game/EngineBuildPreferences.hpp**
    - Copy **EngineBuildPreferences.hpp** to the Code/Game folder for **all** of your game projects:
      1. SD/Starship/Code/Game
      2. SD/Protogame2D/Code/Game
      3. SD/Incursion/Code/Game
      4. SD/MathUnitTests/Code/Game
      5. …and anything/everything else, and from now on

**files for the “Run” folder (only for projects using audio!)**

* + - Copy these files to SD/**Protogame2D/Run** and SD/**Incursion/Run**:
    - **fmod.dll**/**fmod64.dll** (required, for 32/64-bit projects, i.e. Visual Studio platform “x86” or “x64”)
    - Only projects which actually require audio need these files (e.g. MathUnitTests does not)
    - These DLL files **must** be distributed with your game’s .EXE, in the same (“Run”) folder as the .EXE

1. **Add source files to projects**
   * Add audio files to your **Engine** project
     + In Visual Studio’s Solution Explorer, make an “Audio” filter (virtual folder) under your Engine project
     + Add existing files Engine/Audio/AudioSystem.cpp,hpp to your Engine project (in any game solution)
     + Add all 8 header files (.h and .hpp) from Engine/Code/ThirdParty/fmod to your Engine project also
   * Add EngineBuildPreferences.hpp to **every game project** (Incursion, Starship, MathUnitTests, all protogames)
2. **Add files to Perforce**
   * Add and Submit all FMOD files to P4 (fmod.dll & fmod64.dll in Run, all .lib/.h/.hpp files in Thirdparty/fmod)
   * Add and Submit all EngineBuildPreferences.hpp files (one per game project, in each game’s Code/Game)
3. **Edit MathUnitTests/Code/Game/EngineBuildPreferences.hpp**; uncomment this line:

#define ENGINE\_DISABLE\_AUDIO // (If uncommented) Disables AudioSystem code and fmod linkage.

* + Note: this causes most of AudioSystem.cpp to be skipped at compilation, so fmod.dll is no longer required
  + This game-specific game code header file is #included by engine code!
  + The engine finds this file ***for the game currently being compiled***, because $(SolutionDir)/Code is in our include path, and $(SolutionDir) resolves to a different folder for each game solution (even though they all share the same engine project!).

1. **Get all projects compiling**
   * For **every game project** (including all protogames), you will need to check (or change) the following setting:
     + Properties -> All Configurations & All Platforms -> Linker -> **Additional Library Directories** should be:

$(SolutionDir)Code/;$(SolutionDir)../Engine/Code/

* + For your Engine project, set the same setting (located under the “Librarian” settings instead of “Linker”)
  + One at a time – starting with Protogame2D – make sure you can now build and run each project
  + If a project does not need audio (e.g. MathUnitTests), you can disable audio in EngineBuildPreferences.hpp
  + Incursion and Protogame2D **must** support audio; Starship is nice but optional (you can add sounds later!)
  + Submit everything to Perforce!

1. **Add AudioSystem support** to Incursion
   * Find In Files (Ctrl-Shift-F in Visual Studio, with “Look in:” set to “Entire Solution”) for everything you’re doing with your InputSystem (e.g. new, Startup, BeginFrame, EndFrame, Shutdown, delete, nullptr), then:
   * AudioSystem is treated the same (created/managed/destroyed by App, global pointer in GameCommon…)
2. **Play a test sound** in Protogame:

SoundID testSound = g\_audio->**CreateOrGetSound**( "Data/Audio/TestSound.mp3" );

g\_audio->**PlaySound**( testSound );

1. **Update Protogame2D** with AudioSystem support. Make it so pressing a key plays a test sound.
2. **Test and submit everything!**