COMP710: Studio Session 08 – Exercise:

EXERCISE NAME: *C++ – Middleware Integration FMOD*

FMOD **`fmodstudioapi20111win-installer.exe`** has been downloaded from the middleware vendor and using 7-Zip the **`.exe`** has been extracted and committed into SVN's **`shared\API-FMOD\`** folder.

Please note projects using FMOD must include an in-app FMOD logo. This is also available via SVN's **`shared\API-FMOD\FMOD_LOGOS\`** folder. Review the **`FMOD Logo Guidelines.pdf`** from within the **`FMOD_LOGOS`** folder carefully.

Inside the **`fmodstudioapi20111win-installer`** folder there are the following folders:

- **`api`**: Inside this folder is the core (low level) API:
  - **`api\core\inc\`**: Contains the **`.h`** files.
  - **`api\core\lib\`**: Contains the **`.lib`** and **`.dll`** files (x86 and x64).
- **`bin`**
- **`doc`**: Contains the FMOD API User Manual, **`doc\FMOD API User Manual\welcome.mtml`** API documentation.

Open the FMOD API User Manual documentation in the **`doc`** folder and review the "5. White Papers" section. Start with the "Getting Started" section, this details the usage of the FMOD API with C++.

Integrate the third-party middleware (FMOD) into your "GP Framework". The extracted zip has been committed into the **`shared\API-FMOD\`** folder of the SVN repository for you – you should link your project relative to this extracted version of the API.

In Visual Studio 2017, right click on the Framework Project, and choose "Properties". In the "C/C++" section, choose the "General" options. In the "Additional Include Directories" add an entry for the FMOD include headers. This will likely be:

```
$(SolutionDir)\..\..\..\..\shared\API-
FMOD\fmodstudioapi20111win-installer\api\core\inc\
```

This will allow you to **`#include "fmod.hpp"`** in your project's source code.

In the "Linker" section, choose the "General" options. In the "Additional Library Dependencies" add an entry for the FMOD include headers. This will likely be:

```
$(SolutionDir)\..\..\..\..\shared\API-
FMOD\fmodstudioapi20111win-installer\api\core\lib\x86\
```

Next, go to the "Linker", "Input" options. In the "Additional Dependencies" add an entry for the FMOD static library. This is named **`fmod_vc.lib`**.

Now the linker will be able to link to the FMOD middleware when the project is built.

When running the "GP Framework", it will now require the FMOD dynamic link library. Copy the **`fmod.dll`** file from the **`api\core\lib\x86\`** folder to your **`Game\`** folder (the one that the built executable is written to).

You will need to use at least the following functions and methods from the API:

- **`System_Create()`** which belongs to the **`FMOD`** namespace.
- **`init()`** which is a method of the **`FMOD::System`** type.
- **`createSound()`** which is a method of the **`FMOD::System`** type.
- **`playSound()`** which is a method of the **`FMOD::System`** type.
- **`update()`** which is a method of the **`FMOD::System`** type.
- **`release()`** which is a method of the **`FMOD::System`** type.

And the following types:

- **`FMOD::System`**
- **`FMOD::Sound`**
- **`FMOD::Channel`**

Review section "7. Core API Reference" including the "Common", "System" and "Sound" sections.

At appropriate locations in your "GP Framework" codebase, add calls to play sounds effects using FMOD.

Some hints for integration of this middleware into the GP Framework:

- Perhaps add a member of type **`FMOD::System*`** to the **`Game`** class.
- Create the FMOD System on initialisation of the **`Game`** object.
- Also initialise the FMOD System on initialisation of the **`Game`** object.
- Create sounds effects on initialisation of a scene.
- Ensure you update the **`FMOD:System`** on the Game's **`Process`**.
- Remember to release the FMOD resources.
- Finally, consider writing a wrapper class named **`SoundSystem`**, which abstracts the FMOD API, and manages resources similar to the design of the **`TextureManager`**, **`Renderer`** and **`Texture`** classes.

Finally, also note that the **`api/core/examples`** folder contains many examples in addition to the FMOD API documentation.