Lab 12 – Audio

Task 1

Add the AudioManager class and associated classes to your project. The AudioManager code files are available on Blackboard. Copy the files (AudioManager.cpp/h, Sound.cpp/h and MusicObject.cpp/h) into the Project Folder. In Visual Studio, you can add Existing Items to your project by Right-clicking on **Source Files** in the project view and selecting Add -> Existing Item (Figure 1). Select all the new code files to add them to your project.

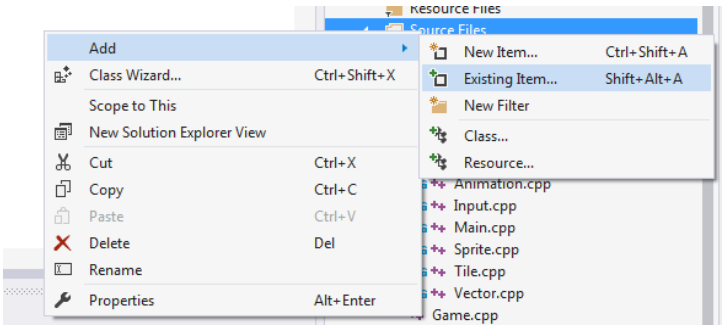


Figure 1 Add existing items

As discussed in the lecture you can create a filter inside Visual Studio to group the related files. For example Figure 2, below.

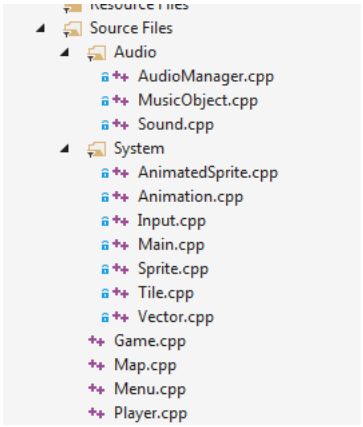


Figure 2 Created Audio filter

Before we can work with the Audio elements of SFML we need to add another library to our application. This is similar to the original library setup done in Lab 1; you may want to look over that Lab Sheet. In **Project Properties**, under **Linker, Input** we need to add an additional dependency: *sfml-audio-d.lib*. Figure 3 and 4 show screenshots of the process.

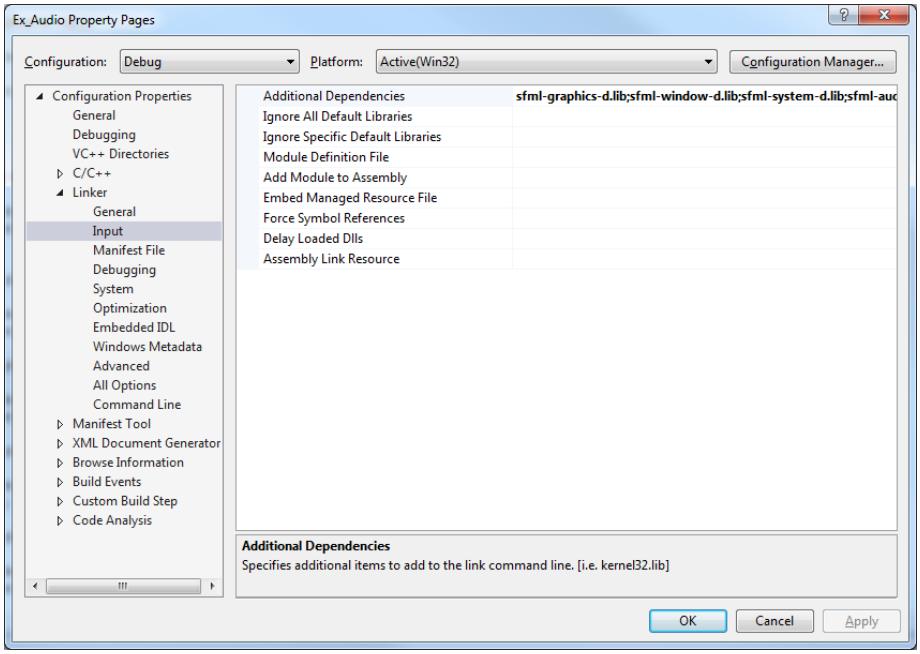


Figure 3 Project Properties

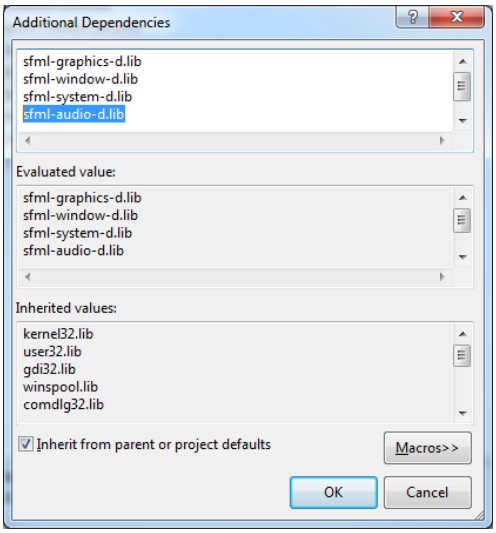


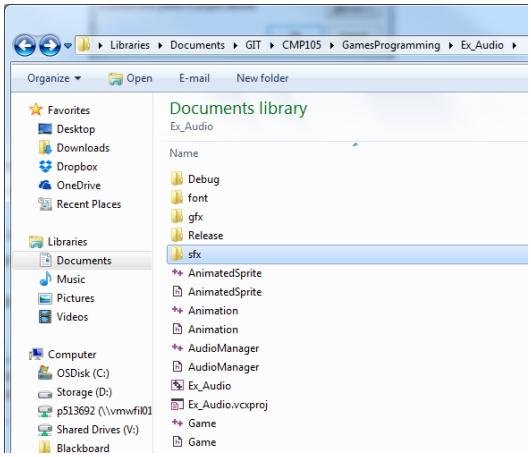
Figure 4 Adding Audio library to dependencies

This should have the project setup to produce audio.

Task 2

Build a simple example demonstrating audio, we will play some background music at the start of the scene and play sound effects on key presses.

Copy the provided audio files (available on Blackboard) into your project folder. These should go in a “sfx” folder to keep audio files organised.



You will need to add the AudioManager to the Game class. In Game.h you will need to include AudioManager.h,

```
#include "States.h"
#include "AudioManager.h"
```

and declare an audio manager object.

```
AudioManager audioMgr;
```

In the constructor in the Game class you will need to add files to the audio manager. For example;

```
audioMgr.addMusic("sfx/cantina.wav", "cantina");
audioMgr.addSound("sfx/SMB_jump-small.wav", "jump");
audioMgr.addSound("sfx/SMB_1-up.wav", "up");
```

Now you will need to add code to the application to start playing the music when the game starts and play sound effects when keys are pressed. In the lecture example, I played sounds when the number keys were pressed.

Task 3

Choose one of the following options to complete:
Option 1: Modify the application so the “Glass_break” sound effect is played when a player or computer controlled sprite collides with the edge of the screen.
Option 2: Modify the application so the “smb_jump-small” sound effect is played when a player controlled sprite jumps.

Task 4

Use remaining lab time to work on coursework.