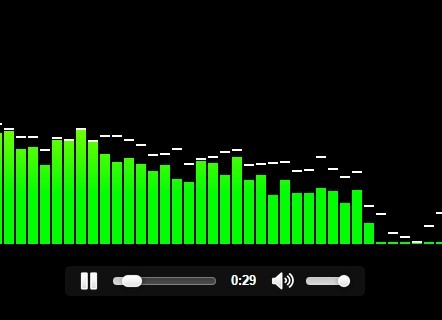
**Complex Game System Brief**

**Overview**I am wanting to make an audio visualiser. The audio visualiser will be able to play audio such as music, sounds, etc, and display the sound being played by a visual queue for example a change of colour, moving bars, making shapes pulse, an image bouncing to the beat of the bass or a 3d object bouncing, etc. It should be able to load audio files such as mp3, wav, etc. It will have the capability to get the audios frequency/volume levels. The audio visualiser will be displayed with the AIE bootstrap. overall in a quick summary it should load audio, play, stop, and display the audio visually while being played. I have included an image below is similar to what this may look like.



**Test Application**The test application will be the bootstrap as it will display the audio being played. The bootstrap will use the audio visualiser DLL to play audio, load, pause it, and get the data about the audio to make the visuals move.

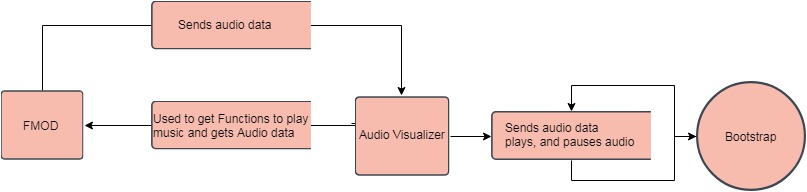
**Functionality/Features**  
The system will have a list of features which have been briefly stated earlier. These features include:

* It will allow the user to load music from the user’s computer as mp3, mp4, and wav.
* There will be the ability to pause and play the audio by clicking a button.
* The interface for the program will use imgui to run all the functions such as playing, pausing, etc.
* There will be the feature to get the audio frequency, for example if the user wanted to just use the low bass frequency there will be a function to get the frequency range between 20-40 Hz which is low bass.

**Extra features**  
some extra features which could be added.

* The ability to change the pitch, volume, channel, mute, and loop of the audio while it is playing and not playing.
* Change the number of bars being visualised while it plays.
* A slider which will allow the user to scroll through the audio.
* Play multiple audio files at the same time.
* Have gain value that can be changed so the bars can disappear/change in size whether it is small, and the sound represented in minimal or big and the bars are going through the roof.
* The ability to select certain frequencies to be played like bass, midrange, and high end of the frequency spectrum.

**Minimum viable product/base goal**The minimum viable product will be able to load music from the user’s computer. It will play the audio from start to end with the option to pause the audio. It will display audio as it plays based of the sound waves, frequencies, and volume with bars raising up and down to the volume of that frequency area, these areas will be representing different sounds. The system will be made both a statically-linked (.lib) and dynamic link library (.DLL).

**UML**

**Constraints, Risks, and Concerns**Some of the Constraints, Risks, and Concerns that I will face while undertaking this project are: for starters I am new to using FMOD and programming with audio, so I am not sure how to go about much of what I am wanting to do. I also am unsure on how difficult this will be as I have found little research to help with making an audio visualiser in C++ with FMOD, however this does give me an opportunity to learn how to use FMOD and all it offers, also it gives some experience with it which may be helpful in the future. There are also some threats with the project such as time as we have a couple weeks to do this and some of the time is going to be preoccupied by other activities such as the gam jam, which takes a week away from working on this. However, this can be overcome with if I put in hard work and talk to any of the teachers which can help with my project which can be classified as a strength as they have a wide range of knowledge which can help. So, in the end I may not end up with as much of a complex game system with extra features that I would have liked.

**Post Mortem**Throughout the assignment there were a couple problems which I ran into and had to overcome to complete the assignment. The first problem I encountered was setting up fmod into the project. I spent a little too much time at the start figuring out how to link it into the project. After I finally set fmod up into the project I did not realise that my git ignore had ignored the fmod files and cost me a fair amount of time. So, setting up the project took me about one to two days which was a little longer then I wanted. Once I was started working on the program it was going well until I ran into another problem with getting the spectrum data which I got suck on for a couple days. The problem was majority of the tutorials or examples I looked at used the outdated version of how to get access to that data and the other recent examples did not work. However, once I figured it out I completed the program but ran into more problems when making it into a .lib and .dll. Although I had trouble with that I just had to spend some time getting the linking correct and moving the UI of the system for the .dll out of the library and into the test app. Those are the main problems I faced. What I would change next time would be making more room for myself to setup a project with other libraries, etc that I have not used before. I would do a little more research around what it is I am doing/using as I thought I did enough research about fmod to get the spectrum data, but it turned out that it was outdated. The main thing is I would try and leave myself extra time for things when using a new library, software, etc just so I don’t end up over scoping and wasting for time then I need on unimportant things. All together I am happy with how the program turned out, it works exactly how I imaged looks better than I thought. The playing and loading works well however I had to change it last second as the old way it loaded only work in debug mode. Besides that the overall program works well and as intended.

**External Libraries**This system will require the use of the following external libraries to function.  
FMOD, FMOD will be used in this project so I can access and use audio within this project.   
imgui, imgui will be used for the UI of the project for example the play and pause buttons.

**Resources/Referencing**<https://www.fmod.com>

<https://www.airtightinteractive.com/2013/10/making-audio-reactive-visuals/>

<https://www.cssscript.com/audio-visualizer-with-html5-audio-element/>

<https://www.cg.tuwien.ac.at/courses/Seminar/WS2010/processing.pdf>