

MOBILE SENSING LEARNING

Mobile Sensing Video Assignment Two: **Transform It**

You are to complete this assignment in **groups of two or three**. Start with the unmodified Xcode project from the video lecture. Use the branch titled *FFTAudioGraphing*. Be sure to use lazy instantiation whenever possible in the following updates and be sure to deallocate any memory that is not managed by ARC. This handout is double sided, so be sure to turn it over!!

Part one: Table It

1. **[0.5 points]** Update the code to use a `TableViewController` as the main entry point of the app. Its fine to use static cells. When you click on the top cell, it should open the original `AudioLab` view controller that the app previously used as an entry point.
2. **[1 points]** Pause the `audioManager` object when leaving the view controller. Then start playing again (if needed) once the view controller is navigated to again by the user. Think critically about where to call these functions in terms of the view controller life cycle. You may need to write additional code in the `AudioModel` class to achieve the desired result.

Part Two: Equalize it

In this part of the assignment you will create a 20 point musical equalizer.

1. **[0.25 points]** Add another graph to the view that is 20 points long.
2. **[0.25 points]** Add another property to the `AudioModel` class that is a length 20 array.
3. **[1 points]** In the `AudioModel`, calculate the values of this new array by looping through the FFT magnitude array to take maxima of windows. Design the loop such that you can process 20 windows and these 20 windows span all the data in the FFT magnitude array.
 - For example, if the magnitude buffer was 100 points long, each batch would be $100/20 = 5$ points long. You would loop through the buffer in increments of 5 points, taking the max of each 5 point window.
 - Take the maximum in each window and save it to the array. For example, if you were getting the maximum of the 3rd window, you would save the maximum into the third element of your array.
 - It is possible, but NOT required, to implement this with the Accelerate Framework.
4. **[0.5 points]** Graph the 20 point array after you have filled it in by adding a graph using the `MetalGraph` class.

Part Three: A True Song Equalizer

1. **[1 points]** Change the functionality of the program to show an equalizer of a song playing (like the RollingStones satisfaction song) instead of the microphone. *Think about:* what code do you truly need to change to make this happen? Be sure also that the song also plays on the speakers. *Note:* the back of this handout has some example code you might be interested in.

[0.5 points Total] For thought:

- If you made the FFT Magnitude Buffer a larger array, would your program still work properly? If yes, why? If not, what would you need to change?
- Is pausing the `audioManager` object better than deallocating it when the view has disappeared (explain your reasoning)?

MOBILE SENSING LEARNING

What to turn in:

- Be sure to indicate who is part of your team.
- The Xcode project of the updated file (as a zipped/compressed file or GitHub link).
- Answer the questions posed above, upload in directory or in the canvas comments.

Helpful Code Hints:

See the code below in order to get an idea about the file reader class (for part three):

```
//=====
// MARK: Private Methods
private lazy var fileReader:AudioFileReader? = {

    if let url = Bundle.main.url(forResource: "satisfaction", withExtension: "mp3"){
        var tmpFileReader:AudioFileReader? = AudioFileReader.init(audioFileURL: url,
                                                                    samplingRate: Float(audioManager!.samplingRate),
                                                                    numChannels: audioManager!.numOutputChannels)

        tmpFileReader!.currentTime = 0.0
        print("Audio file succesfully loaded for \(url)")
        return tmpFileReader
    }else{
        print("Could not initialize audio input file")
        return nil
    }
}()

//=====
// MARK: Audiocard Callbacks
// in obj-C it was (^InputBlock)(float *data, UInt32 numFrames, UInt32 numChannels)
// and in swift this translates to:

private func handleSpeakerQueryWithAudioFile(data:Optional<UnsafeMutablePointer<Float>>,
numFrames:UInt32, numChannels: UInt32){
    if let file = self.fileReader{

        // read from file, loadng into data (a float pointer)
        file.retrieveFreshAudio(data,
                                numFrames: numFrames,
                                numChannels: numChannels)

    }
}
```

