Keefer Rourke
Mr. Fryer-Davis

4 May 2015
ICS4UI

Fast Fourier Transforms and Frequency Visualisation of Audio Files - Requirements

A discrete Fourier transform (DST) is a mathematical method for breaking any waveform into its component sinusoidal parts. That is to say, all waveforms are really just sums of sinusoids. The Fast Fourier Transform (FFT) is a computer algorithm that can perform a DST very quickly, thus making DSTs more useful in computing where things are typically expected to go more quickly.

Using FFT, one can isolate the component frequencies of a sound for instance. This is useful for the analysis of audio, and in particular, visualisation of audio by plotting the component frequencies on a graph, or by performing the visualisation in some other manner using a graphics library such as OpenGL.

For this project to be successful, I will need to be able open and read meaningful data from the PCM packets found in the uncompressed wave audio, and apply a Fourier transform to get the component sine waves of the recorded waveforms. From there, a graph of the frequencies can be produced using OpenGL, as the program collects the relevant data in real time.

This project will most likely consist of three major parts:

- Playing and parsing the audio files
- the FFT implementation
- visualisation with OpenGL

Approximate Schedule:

May 4 – May 8	Finish planning	• Planning and requirements document • Design document • Finish initial research
May 9 – May 15	Open and play WAV files	• Get file name from user • Open file safely • Read from file • Play file
May 16 – May 22	Read data from PCM packets	• Get meaningful data from PCM packets • Determine what data is required for FFT • Figure out how to process the data
May 23 – May 29	Decide on an implementation of FFT	• There are many FFT algorithms and approaches for its implementation • Pick one and begin implementing it
May 30 – June 5	Work on FFT implementation	• Try to finish the FFT part of the project
June 6 – June 12	Learn OpenGL	• Experiment with and learn how to use OpenGL to visually represent data

June 12 – June 19	Final push	• Add OpenGL visualisation to the project • Do lots of testing and debugging • Finish documentation: post-mortem document, testing document