Music Visualizer - Project Summary

By: John Michael Hayde, John Zarnowski, Clare DuVal and Conner Schmidt

Project Description

- Develop a program which interprets music and produces a visual reflection of a song using the fast Fourier transform.

- The Fast Fourier Transform (FFT) can be used to separate one singular sound wave into all the frequencies that make it up. This transform would take in an audio file and produce different "waves" that reflect the audio's frequency and volume.

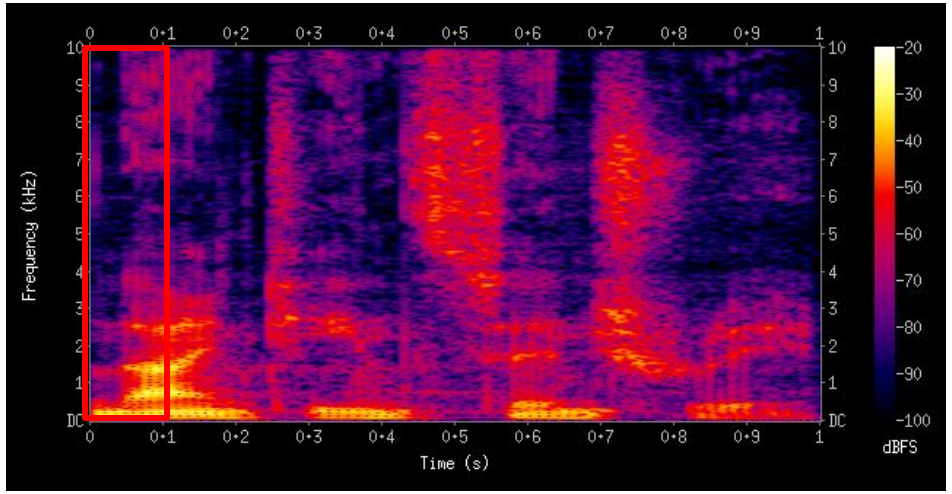
- Develop a neat graphic interface to show the visual reflection of the waves.

- Sync the program with Spotify and enable it to pull music from there.

Development Plan

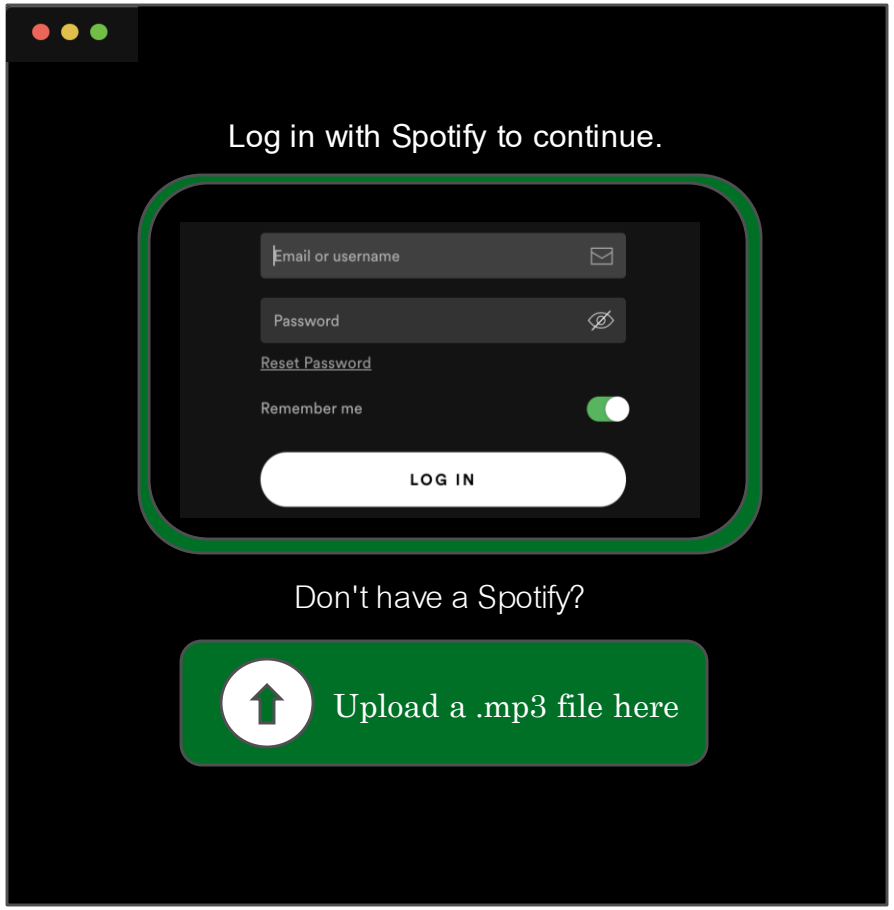
- Chosen language is C++
  - Chosen for its solid library support and its familiarity to the group
  - Make visualizer first, then connect to the Spotify API
  - Steps:
    - Create UI and song upload capability
    - Transform song into waves using FFT
    - Transmit those waves into graphics on the screen (spectrogram)
    - Make the graphics better, maybe let user alter them somehow
    - Connect to Spotify API and pull songs from there

Sketches



https://en.wikipedia.org/wiki/Spectrogram

Sketches (continued)



Requirements

- A machine that has a C++ Compiler
- A machine that can run C++ code

(These 2 requirements can be achieved either via command line or an IDE depending on the system you are running)

- Functional Requirements (user stories)
  - As a user, I can upload a song so the visualizer has a source.
  - As a user, I can observe the visualizer's images to enhance the music listening experience.
  - As a user, I can sync the program with Spotify to create an easy input source for the visualizer.
- Nonfunctional Requirements
  - The program will be created in C++
  - The program visuals will be created utilizing the OpenGL library

Questions?

- Does anyone know how to connect through the Spotify API to pull the .wav file? Or a .mp3 file?