Check-in

The progress on my project so far has mainly been in successfully setting up the environment and finishing the audio processing capabilities, so I’ll be able to work on purely the graphics implementations during the rest of the time until the project is due. The main issues I was having involved setting up the FFT libraries for audio processing on Windows, since these were primarily meant to be used in Linux, and setting up the dll’s and environment in Visual Studio was some of the most painful stuff I have experienced. Also it took me several hours to realize that when using a file stream in C++ I needed to use a flag indicating a binary file, since the stream kept terminating early thinking it had read the end of file character very early into the file. However, now my code can do the full audio pre-processing to have the audio features ready for the visualizer.

What it currently does is take in an audio file, analyze the metadata to make sure it’s in the expected format (stereo, 16-bit depth, PCM), and reads samples into a buffer. A fourier transform of size 4096 is processed for every window with a spacing of 256 samples, (e.g. a transform for 0-4096 is calculated, then 256-4352, etc), then stored in a 2D matrix representing the chromagram of the entire song, as shown below (the example used is the opening of “Yeah” by Mac Miller, I wrote the processed data into an output file and visualized it using a Jupyter notebook to debug and make sure everything was working as expected).

As of now, the processing code detects the main transients by differentiating the spectrogram data to find sudden changes and generates sample indices where these occur to allow the visualizer to be able to trigger the corresponding behavior once the same location is reached in the audio playback. I haven’t implemented pitch detection yet, but this should be relatively straightforward by taking each bin, finding the closest corresponding pitch using a frequency table, and adding it into one of the corresponding 12 pitch bins.

