Miles Grossenbacher
4/25/19

For my final project I would like to make a creative audio effects plugin within the JUCE framework. For the base effect of my plugin, I would like to make a feedback delay, but I would also like to process the delayed audio through filters and/or saturation depending on how deep into this project I am able to go within the time constraints. At this point, I'm not entirely sure of the scope of my project, but as I begin working it should become clear what I can hope to accomplish. At the very least I want to make something I find usable and not too generic. I would also like to build all of the GUI controls that are expected for a commercially viable audio plugin within JUCE. Obviously, the library I would be working with is JUCE, and conveniently, the JUCE application framework separates the source files into an audio processing file called *PluginProcessor.cpp* and a GUI editor file called *PluginEditor*. Within the *PluginProcessor.cpp* file there are predefined methods that serve as the foundational functional blocks of the program. The main one, for example, is the processBlock, which seems to essentially work the same as the callBack function in Port Audio. From the process block I will call to a method that interacts with a circular buffer to create the delay effect. Within that block, I will also call to other methods that will process audio in a variety of ways. Separate from audio processing, I will make a real-time, reactive GUI using the JUCE library. This will consist of sliders to set the parameters of the audio processing.

In terms of the platform, I believe JUCE makes cross-platform applications, but I will definitely be making this plugin to work with MAC OS X.