**Individual Capstone Assessment**

**JP Wang**

Programming and music have been core interests of mine for the past decade now. My team’s task is currently to develop a plugin for an audio workstation. The plugin, which will be a standard ‘VST’ extension, will accept a stream of notes from an audio source and deliver the closest key back to the user. There will be a GUI, DSP, and algorithmic component to this project. These are all areas I have not engaged with, or only learned in a limited manner such as a homework assignment. This senior design will be about taking on a larger project, broadening my programming knowledge into the music sphere, and learning to collaborate as part of a team that has more autonomy over decisions than I have received in my co-op.

The college experience that has most influenced my expectations for my senior design was my final project of my Software Engineering (EECE 3093C) class. My goal was to create a server-based audio workstation for real-time collaboration. I quickly realized that the project was far too large and created only a small python script that shared and played a midi track via a command line. This project, though small, did familiarize me with what was available in music software, as well as the midi format. I also expect my coursework in Intelligent Data Analysis (CS 5152) and Algorithms (CS 4071) to be useful. In my research into existing algorithmic solutions to key identification, most approaches analyze the output of a Fast Fourier Transform and perform analysis with anything from Markov chains to cosine similarity. Both are topics seen in my Algorithms and Intelligent Data Analysis classes.

My co-op will contribute to my project experience in other ways. I’ve spent all five of my rotations at Northrop Grumman as a Cyber Systems Intern. My main programming experiences have involved Python and C/C++. I also did some full-stack development for one rotation. I do expect my experience in C/C++ to come in handy as we will be developing in it. I also was gifted a textbook from a coworker on common design patterns after asking about what a “Singleton” is in our codebase. I intend to keep the book as a reference for any situation we encounter where we are searching for an overarching design. One non-technical skill I intend to use is knowing when to ask for help. For example, I may encounter some music topics that will be unfamiliar to me. I have better educated friends and team members that may be able to guide me in a better direction.

I am excited to work on this project because our solution is one I have wanted for myself. I usually use a plugin suite called Melodyne when I want to know what key something I played is in. Melodyne fully accomplishes this task but has drawbacks. It is paid software (over $100), not real time, slow to load, and its features go unused when I just want to know the key. I was given the initial idea from a musician friend of mine after I sent a social media post asking for ideas so I know other producers would like this tool, should it exist. The most common user in this case is likely to be a musician with the technical chops to play or design a melody, but who struggles to quickly communicate what they’ve done to a bandmate with more advanced music theory knowledge.

My own research into the design of this software has included guided examples in the JUCE framework. At this stage, my team members are at least able to build a basic “Hello World” project and have it load into a standard audio workstation. Other preliminary research includes reading papers for existing approaches to key identification. We have also searched for open-source solutions that tackle similar problems for reference, both within and not within the JUCE framework we are developing with. There are a few milestones we can hit depending on how difficult a problem we encounter is. My role will likely be the scale identification method. The bare-minimum solution should, if it is real-time, at least identify individual notes. This is likely to be the first milestone. If accomplished, I will consider it to be the minimum acceptable contribution from my input. A good job will identify major and minor scales in all keys in real time. If this is too complex, I will also be satisfied with a solution that is less than real time, or uses the more easily analyzed MIDI inputs.