Sprint #3 Review

Features Implemented

- Octave switching
- Flask back-end framework
- Midi support

Issues fixed

- Picked and setup a distribution method, Docker, and built a Docker image that contains all application dependencies, that can be auto-rebuilt when new releases are made.

Successes

- Generally did better with keeping Daily scrums

Problems/Solutions

- Needed to find a commonly used, easy to use, method of distributing the application bundled with it's dependencies. We chose Docker, as it seemed like the choice that made the most sense.

Changes made

- Integrated existing static website into Flask, built docker-based Dev workflow as well as a production-style Flask container.
- Added Midi support, so Midi instruments can be used to as an input mechanism.

Next Sprint

- Again, a priority is more synth sound modes and wave types to give the synthesizer a more versatile feature-set.
- More new GUI features: chord-capture mode, which saves the chord playing when a specific key is pressed. This is a priority because it is a first step for chord progression recognition.
- More backend features: build out Google sign-in, start work on implementing user sessions.

Scrum Review

- Daily scrums have gone better, in general they have been more short and collaborative.
- Organization-wise, many team members are not consistent at tracking their progress in GitHub/Zenhub, making it difficult for the team to keep organized and in sync. This also makes it much easier to determine what needs work next.