Introduction to Programming II Project Log.

Project title: Music Visualisation
Topic: Lesson 10.3 Finishing touches.

1. **Progress in this topic.**
    a. I performed testing on the last both feature and visualisation.
        i. This test can be viewed in the test folder on the file “Sandstorm and Effects Pad testing”.
    b. I performed usability testing with 2 different users.
        i. These tests can also be viewed in the test folder.
    c. I implement the feature of the Sandstorm visualisation going backwards and forwards with a “Reverse” button.
    d. I addressed some bugs found on the “Sandstorm and Effects Pad testing”.
        i. I corrected the problem that seems the visualisation perform slow.
        ii. I corrected some colouring problems when the user enter the Ridge Plot visualisation, caused by the lack of push() and pop() functions in the draw function.
        iii. I decided to take out the Reverb effect on the Effects Pad as it was not performing correctly.
2. **Problems faced and solutions.**
    a. The bug of the Sandstorm visualisation seemed the visualisation to perform slow.
        i. At first, I thought that this was caused to the high number of particles that the visualisation required at a specific time.
        ii. However, I found out that the problem was in my code:
            1. When the particle list passes some threshold, I returned the function, but I never update the particles.
                a. I implement this with the idea that the visualisation would perform poorly with a lot of particles in it.
            2. This caused that, at some time, the x-axis of the particles was updated (when the threshold was not passed) and at some time the x-axis was not updated.
            3. Once the bug was found, it was simple to correct.
    b. The bug on the Effects Pad.
        i. The colouring effect bug was addressed by looking at the code and assure all drawing functions perform a push() and pop() state.
            1. These functions are especially important to not carry a drawing state to another entity.
        ii. I could not address the bug that the Reverb effect was not behaving as expected.
            1. The p5.js documentation is not clear about how to make de desired behaviour.
                a. Basically, I want the effect to behave the same way as the Delay effect:

        i. When the user clicks and holds a button the effect starts, and it remains until the user stops pressing the button.
    b. Both classes, Delay and Reverb, extend the p5.Effect object, so I thought this would be straightforward.
2. As it was a problem I was facing for a couple of weeks now, I decided to leave the Reverb Effect out of the final deliverable.
    a. The Effects Pad would only contain the Delay Effect, which I could implement correctly.

3. Plans over the next few weeks.

a. Final tweaks to address:
    i. Address the lack of “loading” text while uploading a sound file.
        1. I believe is a bad implementation of the whileLoading function callback in the loadSound() p5 function.
        2. But, of course, as I am using the loadSound() function in more places than the preload (as is expected) I can encounter these not desired behaviors.
    ii. Increase the particle threshold in the Sandstorm visualisation.
        1. As I catch that the “performance” problem was not in the number of particles, I can increase the threshold of particles.
            a. I quote performance because, as I stated, It was a problem in my code.
    iii. Add helper text for the user to interact with the MIDI keyboard and the Effects Pad.
        1. As I learned from the usability testing, these features are not so obvious for the user to interact with.
        2. The MIDI keyboard is practically hidden, the user could only access it if someone else tells him to do so, or if he accidentally presses any key from a to l.
    iv. Optional: Hide some controls when the user enters fullscreen.
        1. I am not sure if I would have the time to implement this feature, but it would be a nice thing to add.
        2. The controls cover a lot of the screen, and the visualisations tend to hide behind the controls.
        3. The idea is, that if the user is in full-screen mode, some or all of the controls are hidden for the user to watch the whole visualisation.
b. Prepare the report.
    i. Organize the files and folders of the deliverable.
    ii. Comment on the code and ensure it is clean and readable.