Week 1:

1. I created a way to visually display MIDI files in a barbaric sort of fashion with a bar that roughly follows the music indicating the location the user is currently in. This took about 7 hours because it also included learning about using JavaFX and learning about using jMusic. I have also made so you can drag the bar that indicates where you are in the music (has no effect on the MIDI yet).

   I also spent around 3 hours on successfully making it so you can import MIDI files. These MIDI files do not yet play at the proper BPM but the MIDI data itself can be imported and is "serialized" in a way where the Note, the Note Duration, and the location of the note in the song are all gathered from the MIDI file itself, so it can be treated just as if it were any MIDI file that the user created.

2. In the coming week I hope to expand more on the UI by making a more complex window, one that hopefully includes measure numbers and beats marked per measure. I hope to synchronize the position indicator to properly align with the music. These two tasks should take around 3 hours. I also plan to somehow edit the display of the music to make it more readable. I hope to have the music display in a scrolling pane so that the entire song can be displayed if it is longer than the current window. This should take about an hour.

3. So far the only changes are that some things I might have to do differently, but no changes large changes that should have a giant impact on the final result.

Week 2:

1. This week was sad progress-wise. A lot of time was spent making a grid that allowed users to draw on and then these drawings were turned into to midi notes that could then be compiled into a song. This however proved to have a lot of lag, so this idea was scrapped. It was not entirely a waste of time because I had some good ideas come from it.

   I completed the goal of creating a scrolling window that allows for the user to scroll forward to see the music. Each midi part was transformed into a way where it can be drawn and serialized so that each track is drawn on a separate panel that is then added to the window. These two tasks took about four hours.

   I also learned about java synthesizers and loaded a high quality soundbank so that midi files can load faster and sound good.

   I added buttons to the UI that can play and write the current working midi and redid how music plays using the synthesizer in order to avoid strange lag that was occurring. This took about an hour and a half.

2. In the coming week I hope to have correctly imported and displayed midi files in the window this will take an unknown amount of time but shouldn't take too long because I think the problem are my for loops. I hope to have the bar that follows the music working, because I

broke it, I can't seem to get it compatible with the scrollbar. I tend to add perhaps another window to the UI where notes will be edited or something of the like.

3. The changes will still be on my part and maybe visually editing a midi file isn't feasible but perhaps they will be edited by text boxes? They will still be editable but the how is in progress because I had to scrap a lot.

Week 3:

1. My first problems I solved were that I fixed the importation of music so that all midi files are imported correctly, and their chords and notes into the right position. This took about an hour. Next, I redid a lot of code to make the program run faster by multithreading parts of the code. This took about four hours because I had to change a lot of methods and the way things work as well as add a class. Part of refactoring the code included testing that I can change a note's value and it will display this change visually using an observable list to know something has changed, but it also changes the note musically. This is also important because I made it so you can select each midi note by clicking on it. This took about 2 hours.

   I successfully imported information from the original midi and save it to the new midi file, including the instruments and what tracks the music was on.

   The next thing I did which took about 3 hours was to (#BringBacktheBar) bring back the bar that denotes the location of the music and have it follow the music seemingly correctly. JavaFX is a nightmare sometimes in adding elements… I successfully have a scrollable pane inside of a border pane that has a bar inside of the scroll pane. This is a major part of the UI, displaying the music correctly and showing the user where they are in the music.

2. In the coming week I hope to make it so users can edit MIDI files more easily and perhaps display some sort of measure bar or something. This will take an unknown amount of time because what is required to do this is unknown. I hope to add something that tells which instrument is at each music note and create a "standardized" way for users to edit notes (so they aren't just putting notes randomly?). This will take certainly several hours because basically all I have to work with are ticks and milliseconds for location.

3. There are no major changes to the functionality of the project so far.

Week 4:

1. This week I have added panes inside of the panes. I have extended the scroll bar to go the length of the music window. This took a small amount of time. I added a feature in order to import soundfonts, to change how the instruments sound. I did not end up adding measure bars, but instead I added a tempo text box that allows users to set the tempo of the song. This took about an hour. This is necessary work in order to add the measure bars. I added the feature to import a file from anywhere in the system by opening up the systems file chooser and then selecting a file and sending that file into my methods to extract the data. This took maybe three hours. I have a rudimentary delete, it turns out when I add a note, it

just adds that note, but also keeps the previous note. This error I am not sure how to solve yet. I experimented with different ways of viewing the notes and have slightly changed them so that individual notes can be seen, this took about an hour.

2. In this coming week I hope to fix the previously mentioned error with the notes, and to display the instruments names next to their proper parts. I hope to also begin working a view window to edit notes, or something like that, but I think the previously mentioned error is something big to fix but will make users creating music much easier to implement, I also plan to start testing ways to do this.

3. There are no major changes to the functionality of the project so far.

Week 5:

1. This week I fixed inefficiencies in the code, such as opening a file twice. I modularized code so that their functions can be used in different tasks. This took about an hour of various things. I have discovered other bugs, so of which I do not know how to fix quite yet. I have a pane that allows users too zoom in and out that is scrollable. This pane whilst the music is playing locks with the vertical bar on the screen. This took about 10 hours of trying different configurations. I have measure numbers using the equation [Measures = (Duration * BPM) /(Numerator of time signature)] I create a measure indicator at the top that displays what measure the user is in. This right now can be clicked on and it the x position can be gathered. This x position can be translated into a microsecond time that can be sent to the sequencer that is the basis of the idea of changing where the music starts playing. This also took about 4 hours. I made it so the instrument is displayed on the piece of music itself, and that the instrument is name is retrieved dynamically from the sequencer. By each pane having an instrument value, the basis is there to allow users to start changing the instrument. I have also made it so users can select multiple notes at once by clicking on them so they can perform increment or delete operations on a selection of any size. These steps took about 3 hours because I had to learn about MIDI standards and how I can extract what instrument the MIDI is given only it's number.
2. Next week I plan to implement starting the song from a different location other than the beginning, this should take a few hours, but the basis is already there. I also plan to implement allowing users to change the instrument of tracks and begin testing creation of measures and notes.

3. There are changes about implementation, but no changes to the functionality of the project so far.

Week 6:

1. This week I made it so users can start the song from any place by clicking the location on the top measure bar. I also made it so users can skip around while the song is playing, though currently are still locked to the scrolling animation. This required some thought and took about 3 hours. The next thing which took about 5 hours was making it so users can drag any

note anywhere on the panel (limited to each panel) they can drag in both the x and y location which correctly changes the underlying elements in the score allowing users to edit any note that they choose. I improved the method to delete notes so that it should function more quickly. It took me about half an hour to create the underlying base for changing instruments of a part, right now it works, but there is no UI for it. The rest of this week has been bug testing, and I still get bugs of random index out of bounds, the console says source unknown, and I can't find what is causing these bugs.

2. Next week I will make it, so notes can be added, and deal with the idea of the adding of measures

3. No changes to the overall functionality of this product. (I talked to you about no drum machine, or loops, because this is now a different kind of program, they don't make sense)

Week 7:

Found the midi channel error

Week 8:

1. This week I messed with trying to see if I could allow users to use a midi device to input notes, that would take a very long time and I won't do that. I made it so users can add notes correctly, to the correct instrument, anywhere in the song. I also made it so that users can add tracks to the song. I've been working on an interesting issue with what midi channel each track is being sent on, but I think I finally have a solution, these all took around 9 hours. I also made it so users can select a MIDIPane, to add notes you must first select a midi pane.

2. Next week I will make it so you can create a score with a name from scratch and add command pattern for undo function and I will fix the channel problem. (probably by using a LinkedHashMap to retain order and by using the name of the instrument as the key while the value is the channel) I'm unsure how long this will take because I have been working on it for a while. I also need to make mechanism for editing note length.

3. No changes to the functionality of the project, but users can't use a midi device to input notes.

Week 9:

1. This week I have made it so users can save a file under whatever name they want, wherever they want to on their system. This took about 30 mins. After five hours I successfully made a "solution" to handle the problem I was having with MIDI channels. This puts each instrument on a separate channel, while putting the same instruments on the same channel, this however limits the instruments to 15 + drums. With more time I would try to find a better solution, but this was the best I could do. Along with created the needed map for midi channels I created a map for the pitch values. Users can now edit a single note and type in, for example, C4 and that will set the pitch of that note to middle C, or C in the 4^{th} octave.

In this same window users can modify the duration of a note by typing in a double value. Setting up this system and having it work correctly took about two hours. I have also made it so users can go to File -> New to create a song with just a piano 1 as it's instrument allowing them to add instruments. This took about three hours because there were some odds and ends that needed tying up, so I refactored my import method so that a New file is actually just importing a file named Untitled. If users named a file untitled then that file will be the one that loads when they hit new file. In about 5 minutes I made it so users can upload their own sf2 file (soundfont file) in order to customize the sound of their instruments. I made it so that the number of measures dynamically changes depending on the length of the song, right now it is calculated when a user hits the play button, but it successfully goes to the end.

2. In this final week I will make it so users can delete a track and various bug fixes and prime it for delivery, making sure the barebones to be functional (not necessarily non-tedious) are in place.

3. The changes include that tracks will not have images displaying what they are and this choice is because all the instruments are created dynamically, and depending on the system I don't know what instruments exist. Another change (which I noted before I believe) is that there are no loops or drum machines or patterns, because these would take too much time. There is no input from a midi device because this would take weeks. Time display probably won't make it into the final project (it's still in the works), there is a relative time display, relative to the end of the song built into the automatic scrolling mechanism. The "piano roll" exists just not traditionally it is all the panes that allow for the dragging and dropping of notes. Ultimately the command pattern is the biggest non-inclusion, I tried many ways of implementing this and the simplest still had to store a lot of information, for example if you changed a MNB you would have to store that object in a queue and also store a pointer to that object in a different queue so that the queue could be set to the object inside that queue, and anytime anything was changed about a note, you would have to store it. A fundamental problem with dragging the note then becomes that each of these movements will register as a change, and this would make the undo function useless/insanely tedious. It seems impossible without a lot more time because of the difficulty of so many changes instantly happening.