Daniel Rusetski 219663590

EECS 1021

April 10th 2023

MAJOR PROJECT

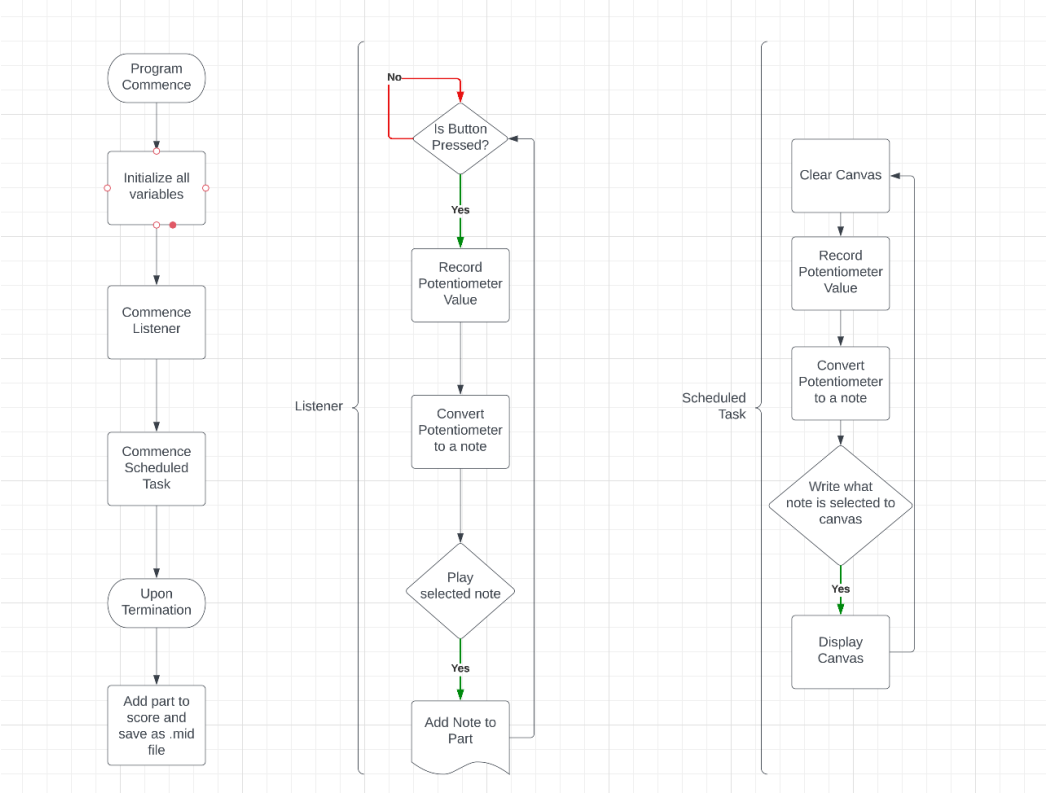
INTRODUCTION:

This minor project is based on the Grove Beginner Kit, specifically using the Potentiometer, pushbutton and OLED screen. The objective is to develop an idea for a prototype MIDI player that is capable of playing and recording notes that a user plays.

CONTEXT:

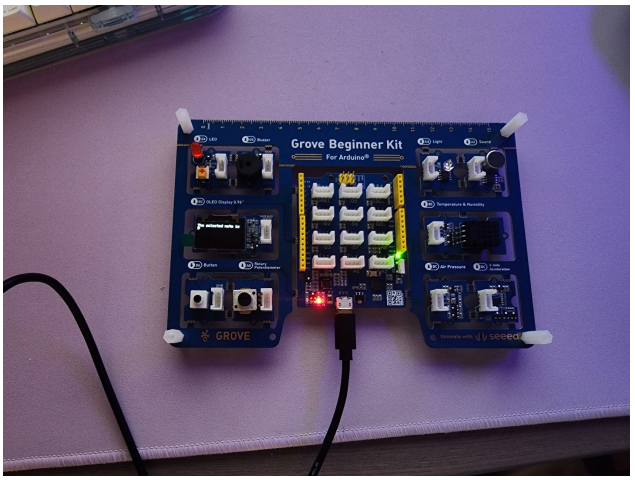
The context of this project was to create a prototype of a basic musical instrument that does not require any external hardware/software to play notes and record them into a file that can later be opened.

TECHNICAL REQUIREMENTS / SPECIFICATIONS:



The program should be able to transpose the potentiometer position to a note on a one-octave scale, and play the selected note on the push (but not the pull off) of the button,

and avoid any repeated inputs that may arise from how arduino or firmata function. The program will then feed the notes played into a score that will be saved and can be played back.



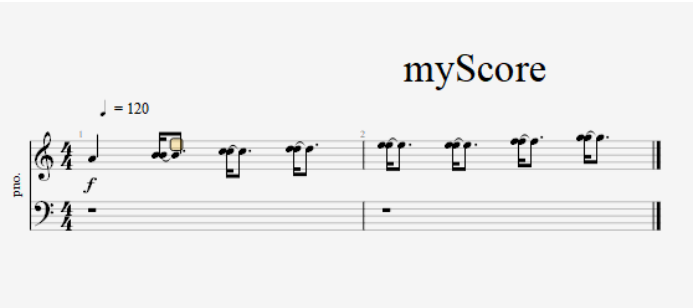
COMPONENTS LIST:

- Grove Beginner Kit

PROCEDURE:

To begin the project, imported all the required components and defined all pins and created all objects. Next I needed to create a listener that only reacted to the button depress, and a way to have the OLED screen relay the position of the potentiometer. I originally implemented the OLED screen on a separate listener that would refresh when the potentiometer moved alongside the button listener, but this resulted in a significant bug in the system where the listeners would interfere with each other, causing the note selection and playback to be inconsistent. To avoid this, I created a Timer Task that actively refreshed based on the potentiometer's position. This alleviated the previous issue and the program ran smoothly. When the program is exited, a shutdown hook adds the part to the score, then saves the score as a .mid file to the project directory. This was actually a feature I added at the very end as I realised that it was fully possible to do so, and provided far more use to a user as it avoided the need for any additional software.

TEST:



In order to verify that the program works, I first made a barebones program that just played notes on the press of the button, then created another program that just transposed the values of the potentiometer to a note. Once I verified that

both work, I combined them. The last part was displaying the selected note to the OLED screen. As mentioned previously, having 2 listeners resulted in an unresponsive program, so I had to make a new timer task to put the code into. After that, I realised that sometimes the code would take ghost inputs. In order to avoid this, I created a small delay in the system using Thread.sleep in order to ensure the code doesn't take an input as two or more accidentally.

LEARNING OUTCOMES:

- **CLO1:** To ensure the program worked, I first tested each component individually. Although I could use listeners to control the OLED screen, using a TimerTask proved to be more efficient.
- **CLO2:** By using various APIs such as Firmata4j and JMusic, I was able to create a program that met the given requirements of the assignment.
- **CLO3:** In order to record the score, I recorded each inputted note into a Phrase, which is upon exit plugged into the score.
- **CLO4:** The program used a listener to dictate when to play the notes, and a state-machine to decide which note to play.
- **CLO5:** I initialised various objects throughout the code such as the Grove Board, display, and score, as well as revising the basis of my Screen Display to use a TimerTask instead of a listener and adding in the feature to record as a .mid.

CONTINGENCY:

The one concept that I could not implement was variable timing for the notes. Originally I wanted the note to be as long as the button is pressed, but I could not get the concept to work within a listener and the existing loops. For the essence of time, the feature had to be scrapped. Next time I would have to put forth more effort to learn the quirks and lesser known information of FirmataJ and how to use listeners and loops with it.

ADDITIONAL MATERIAL:

Although the project does not have anything to do with the selected topics, I took this as an opportunity to create the basis for a unique widget. Music recording is an expensive and overly complex process, especially charting it. This project aims to solve that by automatically recording the notes played WITHIN the instrument. This means the user requires less cables, less software, and less hassle.

CONCLUSION:

Overall the project successfully met the requirements given in the assignment with the use of listeners, a state machine, and a score to input recorded values that are later saved.