Implementation Manual for Music Maker

Music Maker as a Java program that allows the user to create, save, and load songs with multiple instruments. The program implements GUI through the use of JavaFX.

Main Classes:

MusicMaker (Main Application):

This class is the main entry to the program that manages the GUI. This class also allows the user to change instruments and edit their notes.

The music maker class handles the UI elements of my project. It has two different scenes, a scene for viewing the overall song, and a scene for placing notes within an instrument. The music maker class contains the Pane inner class. A grid of Panes is used to create the note placement grid. Event Handlers are used to edit an instrument's note data when each pane is clicked.

Song:

The Song class contains the BPM (beats per minute) and calls each instrument to play notes at each 1/4th beat. It also includes the functions for saving and loading songs.

The Song class contains 8 different instruments that make up the song as well as a variable that determines the song's BPM (beats per minute). This class also manages the current position in the song. This class has functions that start and stop the song. When the play button is clicked, the Song class starts a timer to determine when to play the notes of the song. When the timer finishes, which occurs every 1/4th of a beat, each instrument checks to see what notes should be played at the current point in the song. When the stop function is called, the timer is canceled, and the song position variable (time) is set to 0. This class also manages saving and loading. When the song is saved, a new folder is created with the song's name that was typed into the text field. The song's BPM is saved to the "songdata.txt" file within that folder. Additionally, this class calls each instrument to save their note data to separate .txt files within the same folder. When a song is loaded, the note data and BPM are read from these text files.

Instrument:

The instrument class represents an individual instrument within the song. It plays .wav files contained in the instruments folder dependent on the location and pitches of each notes. This class also maintains note data (pitches and playtimes) and saves/loads this data using .txt files.

The instrument class loads .wav files in order to play sounds. This is done by using AudioInputStream and AudioSystem. Each instrument contains two integer arrays: *playtimes* and *pitches*. The *playtimes* array determines if a note is played on a given beat as well as what type of note is played (0 for no note, 1 for tap a note, 2 for hold a note, and 3 for a sustain note). The *pitches* array determines what pitch each note is, ranging two octaves (24 possible pitches). The folders that contain each instrument's sounds contain 24 different .wav files numbered from 1 to 24, each with a different note pitch. What pitch is played at each note is determined by the number that is contained within the pitches array at the current section of the song.

Folders and .wav files:

Audio files are located within the instruments/ folder. Each instrument has a folder that contains 24 separate .wav files, each with a different pitch ranging from two octaves.

Song data is saved in the songs/ folder. Within this folder, each song is saved in an additional folder named after the song's name and contain txt files that contain the song BPM and instrument note placements.

How to Run the Program:

To run this program, ensure you have Java Development Kit (version 8 or higher) installed to your computer as well as a compatible Java IDE. Additionally, you must be able to run JavaFX on your device.

Open the project in a Java IDE and ensure that your file directories are correct in the launch.json and settings.json files.

Finally, compile and run MusicMaker.java to start making your own music!

For information on how to use the program, read the User Manual.