John Botonakis
CS-410
03/29/25

Lab 2: Bell Choir

Explanation of the Program:

When running the ANT build, you are executing the main method in the Tone class, which serves as the entry point for everything. First, the Tone class initializes the audio output system and loads a musical score from a file, which can be specified via a command-line argument, and validates that the file has no error notes. After validating, it breaks up the notes and their lengths, storing them in the "loadedSong" Array List. It then creates a Member thread for each unique note in the musical score, excluding rests, with each Member responsible for playing its assigned note. Once all the Members are created, it makes and begins a Conductor thread, which orchestrates the performance. The Conductor iterates through the loadedSong list, instructing each Member thread to play its note at the correct time and duration. The Member threads, write audio samples to the shared SourceDataLine, producing the musical output. The simulation runs until the Conductor has played all notes in the score, at which point it terminates, and the program ends.

Meeting the Requirements

I believe that I meet all the requirements for this program. With the loadSong method in Tone reading the song from a file, I am able to not only read a file containing a list of Bell Notes, but also validate that they are not broken. Each member is assigned only one Note, and once the note is assigned to that member, it is only played by that member. My Conductor class controls the tempo through the use of the tempoBPM, and ensures that only one note can be played at a time using the cues for each member. Finally, each member being instantiated on their own separate threads ensures each note is on a separate thread, and the notes are played in order with the appropriate timing due to the length of each note being defined with noteLength.timeMs().

Challenges

Starting out, I actually thought I had this in the bag within a week, because I was able to create the Conductor and the Member classes relatively quickly! However, as I continued on, I separated BellNote from Tone, I started making Test classes that ended up not effectively testing what I had, and thought WAY too much on how exactly it was supposed to work. Ultimately I had to reset to a point where I was comfortable, and had to take it slow, building each functionality piece by piece. I knew what the end product should've been, but the finer details, as usual, escaped me. All in all, I am pretty proud of my here and it was pretty similar to the Juice Bottling lab. I can definitely see the appeal of multi-threading in apps like these, though I'm still a little bit unsure of when to prefer it over standard methods.