Tunerfish

by Gwen Tran, Gavin Werenczuk, Chris Greenup

Changes and Deviations

Tuner / Audio Analysis

At this rate, temperament will likely not be an option.

Note History

- Using Json instead of SQL
- Data type of Note in Note Class is changed to String

Metronome

- Tempo uses string instead of float.
- String is converted to int so that it can be divided, that way the user can input tempo in BPM rather than milliseconds.
- Division sum is stored as float, but converted back into int in order to be used for tempo.
- Using while loop instead of bool.

Note Player

- No longer needs Note and Note Player Class
- Notes can simply be played by using their frequency

What Have We Accomplished

What has been Completed

N/A

What is Partially Completed

Note Player:

Get some of the main notes work and playing in Window Form

Note History:

The backend of it is working. The only thing left is binding the data to DataGridView

Metronome:

The metronome feature is work and allows for BPM to be entered and changed while playing the user's current BPM entry. Still requires a GUI.

Audio Analysis:

The application successfully listens through the microphone and records it into a buffer. It can then use that buffer and, after some type changing, display the results in a graph

What Still Needs to be Implemented

Main Awaiting Features:

Tuner still needs to be made

Linking All the features of their pages together

Binding Data to DataGridView

General Things:

The GUI for the app needs to be made.

Unresolved Issues

Tuner/Audio Analysis

The assumption for how notes work was incorrect. The fundamental frequency isn't always the loudest frequency in a sound, which has increased the complexity of the problem. I'll have to take some form of sample of the spiking frequencies in the FFT and choose the lowest one. The problem with that is I'm unsure how many I'll need to do or how I'll even do that.

History

Still trying to figure out how to bind data to DataGridView