Chris Chen
ChrisChen@u.northwestern.edu

Holliday Shuler
HollidayShuler2016@u.northwestern.edu

William Xiao
YuxuanXiao2017@u.northwestern.edu

Pitch Perfect 3

An Educational Pitch-Learning Tool

Northwestern

EECS 352
Machine Perception of Music & Audio

interactive audio lab

Bryan Pardo
Professor and Adviser

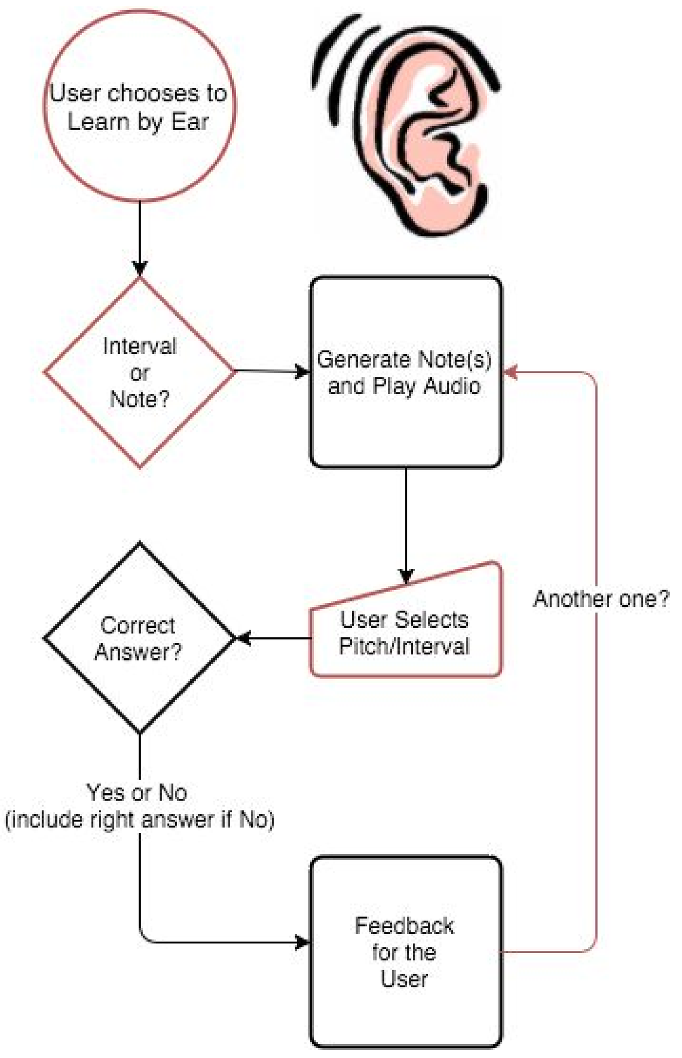
The Motivation

Ear training to recognize notes and intervals is a fundamental skill for musicians and very useful for the musically inclined. A solid understanding of intervals and pitches serves as the foundation for more advanced concepts like chords and harmonies.

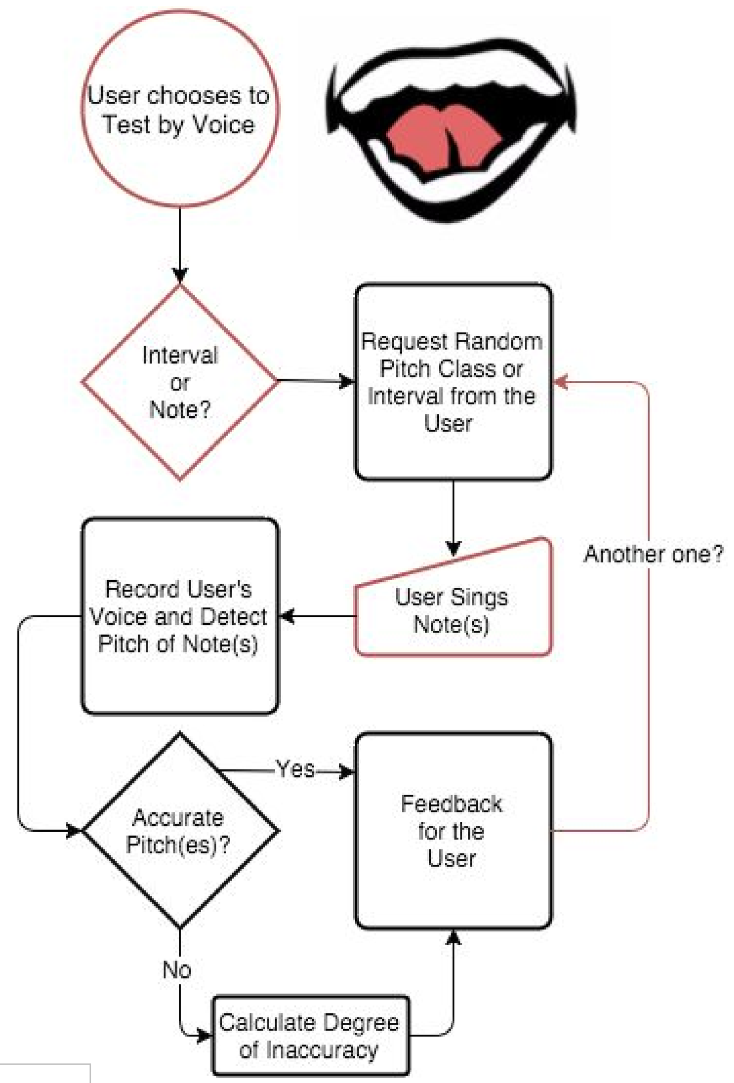
An easy-to-use learning tool built into a web browser would allow anyone to improve these skills in their free time without expensive lessons or an instrument.

Our Solution

We believe that the best way to develop one's listening and singing accuracy is to practice both skills. Repeatedly. So, we've created an application that lets people quickly and easily train both their pitch recognition and the pitch accuracy of their singing voice.



The flowchart above depicts the process by which a user interacts with the "learning" part of the app , along with the backend machine processing (mostly to generate the random notes).



The flowchart above depicts the process by which a user interacts with the "testing" part of the application. Notice that the machine processing implements the pitch-tracking behavior here.

How We Did It

For the **machine processing** portion of the project, we made extensive use of the Web Audio API for playback and recording of sounds. The Web Audio API provided an easy-to-use abstraction to access the user's microphone and speakers. All of our pitch tracking is performed directly in the browser through JavaScript, eliminating the need for server-side processing. We used an implementation of the McLeod Pitch Method algorithm implemented in the pitchfinder.js library.

Our web application's **user interface** is created directly in native HTML5, CSS, and JavaScript. We used the Materialize CSS framework to create responsive, interactive, and attractive pages.

The piano **audio samples** used in our application were obtained from Bigcat Instruments. The piano that produced the sound was a Baldwin Baby Grand.

Qualitative User Testing

The application's interface and its educational value were tested and evaluated by two user classes - those with little to no musical background, and those with significant musical background (i.e. multiple years of experience with playing a musical instrument).

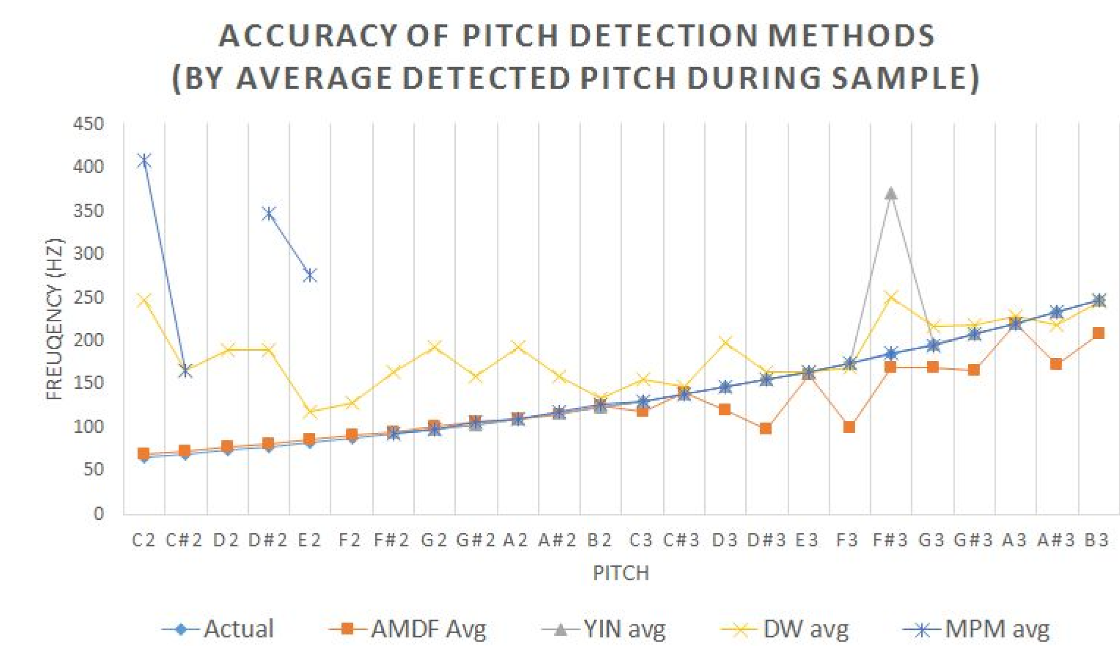
The originally tested **interfaces** can be seen in *Figures 1 and 2.* User feedback indicated that the tool would be more intuitive if the pitch class selections were laid out horizontally (like on a piano) rather than vertically. Furthermore, the "Submit" and "New Note" buttons needed to be set further apart to avoid user confusion.

When evaluating the project as an **educational tool**, users unanimously preferred helpful statistics (or even a scoring system) to track their improvement and more encouraging messages within the app.

However, they gave conflicting responses about how many tries a user should be given during learning and/or testing. Some testers thought that it would be most beneficial if the user were only given one shot to identify or sing the correct pitch (es), citing how much easier the exercise became when given a second chance. Other testers preferred being given more chances, especially for the "learning" portion of the app. Our final interfaces are displayed in *Figs. 3 and 4.*

Choosing a pitch tracker

We tested four pitch detection methods with pitch samples to choose the most accurate one. The methods tested were Average Magnitude Difference Function (AMDF), YIN, Dynamic Wavelet (DW), and McLeod Pitch Method (MPM). The graph below summarizes our results. The best pitch detector for the average human voice range was MPM.



Future Work and Improvements

Possible future additions to the application include testing for the recognition of musical scales, creating a scoring system, and better handling of audio signals that make it easier for users to record their singing.

Identifying scales and chords by key would be the logical next step in advancing the app's features. Also, a scoring system that awards points based on how closely a user identifies or sings notes would provide people with a better sense of their skill level and incentivize improvement.

Machine processing could also be improved by filtering out background noise and being able to parse two distinct pitches from a single recording.

"Before" Screenshots

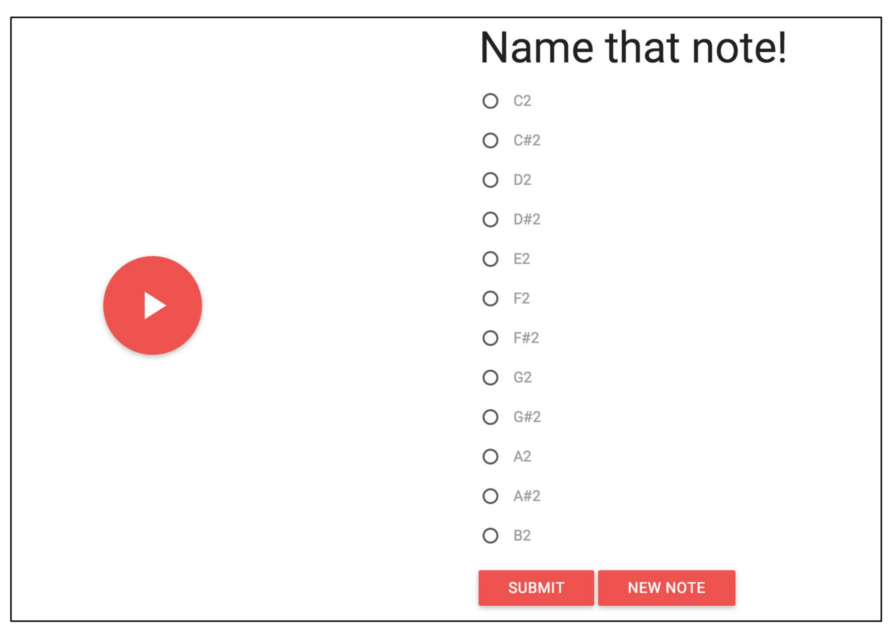


Figure 1. A look at the original interface for the pitch learning tool. Notice that the radio buttons are vertically arranged, and the Submit button is very close in proximity and appearance to the entirely opposite New Note button. Users encountered the most issues with these two design decisions.

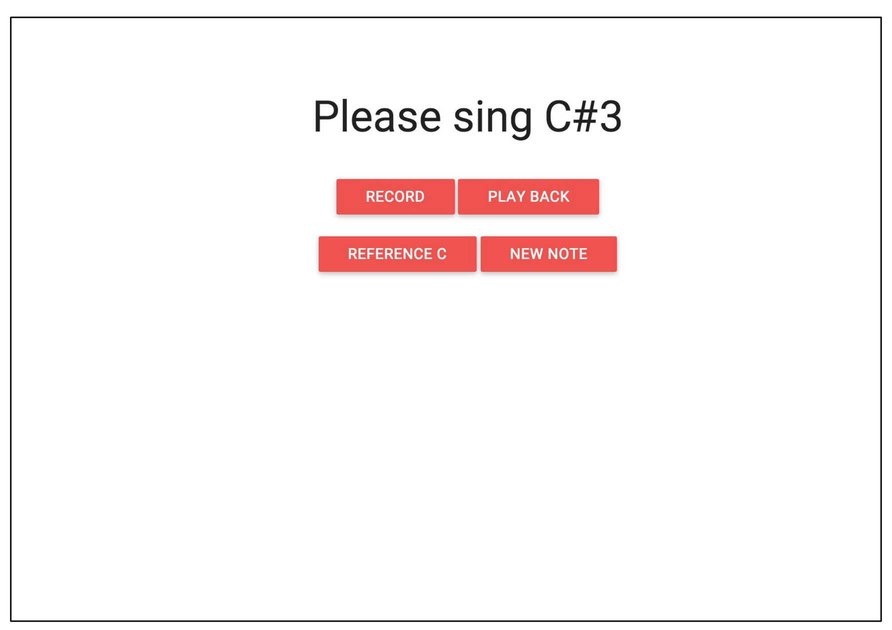


Figure 2. The original interface of the pitch testing tool. Users lamented the lack of differentiation among the buttons and the close proximity between them. The interface also lacked a way of notifying the user exactly when he or she should start and stop singing.

"After" Screenshots

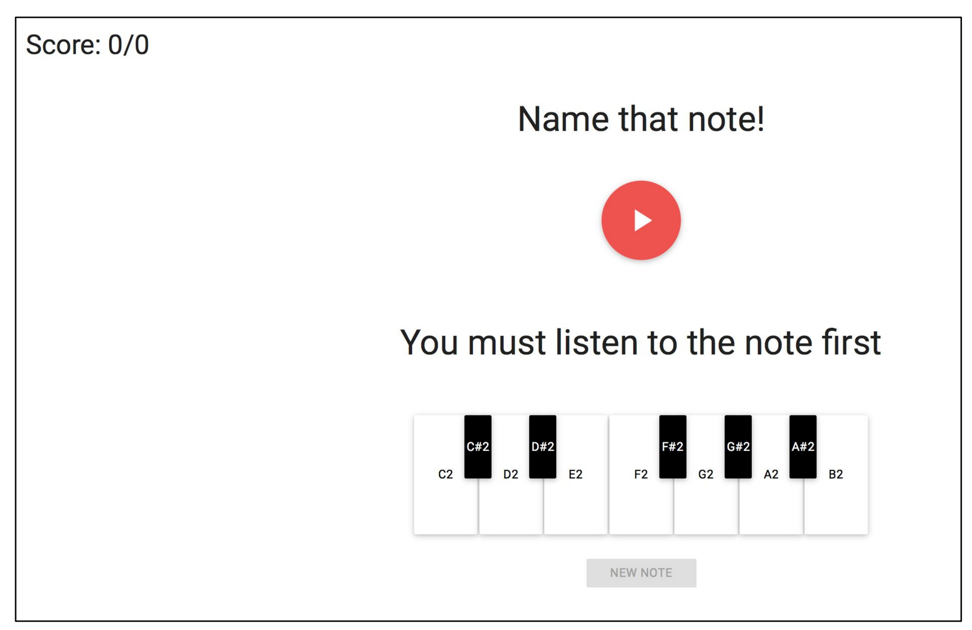


Figure 3. The new pitch learning interface. In response to user feedback, the pitch class choices have been converted from radio buttons to real piano keys. The Submit button has also been eliminated entirely, so that users need only press a piano key to check their answer.

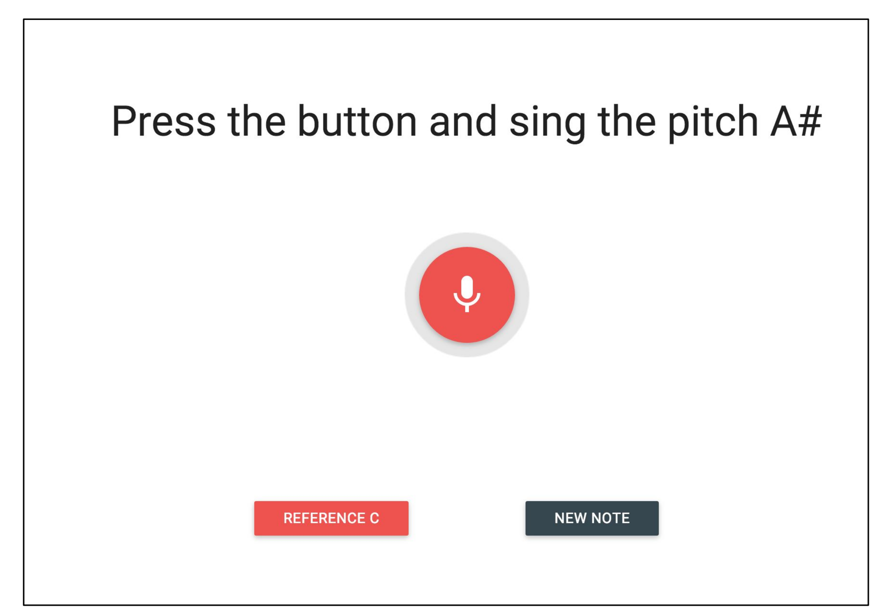


Figure 4. The new pitch testing interface. When the microphone button is pressed, the app gives the user a countdown for when he or she should begin singing. A progress bar surrounding the button informs users of how much recording time they have before the app processes and grades their singing.