

CS446 Deliverable 3 - Demo Documentation

Project Title: Pitch-Perfectly-Accurately Practice

Irvine Yao (b6yao)
Jialin Shan (j6shan)
Alex Lai (x7lai)
Johnny Gao (z53gao)

1 Status Report

For the current stage of development, our team was able to finalize the basic logic of our application. Our application will utilize an open source library called Tarsos, a library allowing us to detect the frequency of sound input from the microphone of the device. So far, we implemented the basic UI of our application, the backend logic, and Note Practice mode. In our backend, we have a model class to store states of the program and handle all the question generation, UI views update and answer comparison. After user launches the app, they are automatically in Note Practice mode, and our backend model will generate questions by creating a corresponding question object based on, for now, a set of candidate notes, and then update the view of the front end respectively. When user start to sing, Tarsos will automatically detect sound input from the microphone of the device and return a double, which is the frequency, back to the model class which will determine whether the user was singing the right notes.

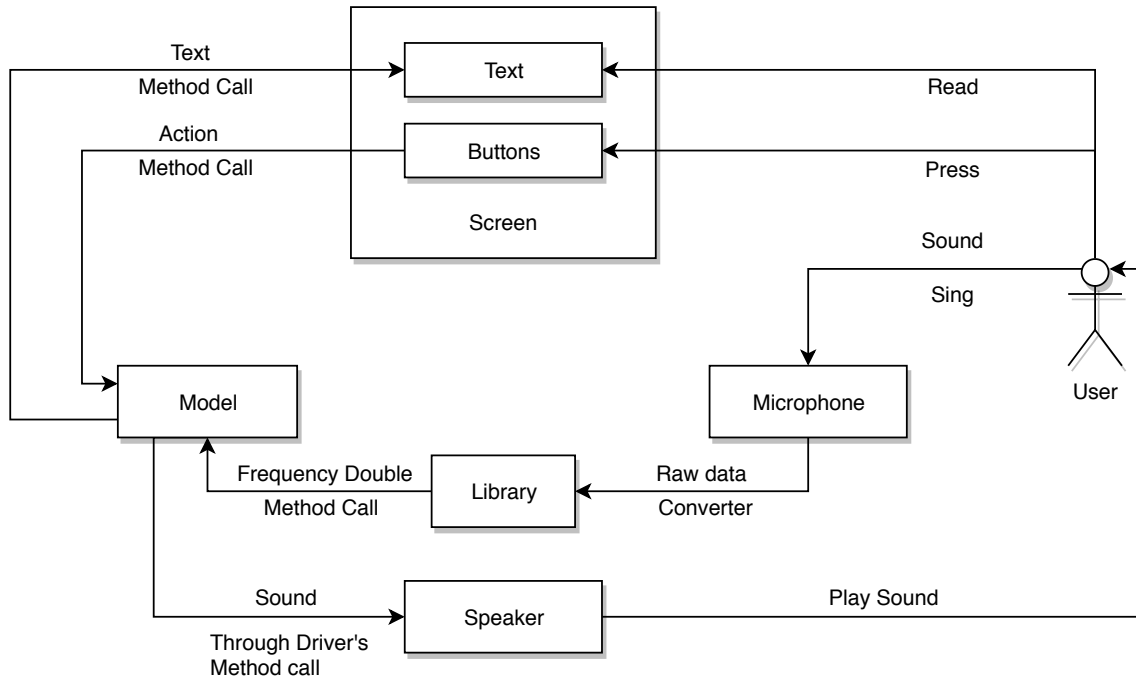
Since our focus was on backends logic design and the library understanding so far, we did not go very far into the UI development at this stage. Even though a lot of the pages and features are still under development and buttons are fake, the main functions UI has been drafted and implemented successfully into our app so that we can test and demonstrate it in our prototype.

The main challenge we face during the development so far is trying to find the right library and understand how to use it. Since detecting frequency of a sound requires a lot of complicated algorithms and understanding of the physics of sound, we spend quite amount of time researching and testing libraries. We eventually land on Tarsos because it provides the exact things we need without being too confusing, even though it still took us days to fully understand the API.

2 Demo

First we launch the app with the microphone permission turned on and it will enter Note Practice mode automatically. Then we will walk through our implemented UI, which includes: main view with a question prompting users to sign a note, an arrow that will jump corresponding to the answer user gave, and a playback button which will play the note that appears on the screen. We will also demonstrate those UI components that are still under development, such as the left menu button containing all of our modes as buttons, settings and help button. For now, most of the fake buttons will direct us to a blank page saying under construction. Then we will start to demonstrate Note Practice mode. One of our team members will mimic the user, and sing the note based on the question presented on the screen, and then adjust his voice according to the feedback through the bouncing speed of the arrow. If time allows, we would also like to invite volunteers from the class to come up and try our app.

3 Architecture Overview



Note:

1. Library component contains all the libraries we need to process the sound data, such as Tarsos, and give a double to represent that sound's frequency to the Model.
2. After Microphone got the sound, android will convert that sound to data automatically.
3. Model will pass the audio file to the system play function and android phone will convert that into analog signal through its Digital Analog Converter, and then play it through phone's speaker to the user.