Project Plan

Project Title: Play Piano

An AI Piano Tutor App

Supervisor: Dr. C. Wu

Group Members

Liu Yujie (3035329754) | Anant Garg(3035365978)

Project Background

Many parents send their children to learn a piano at school or with a private tutor. These beginners may also be learning the piano from the related videos available online. The music instructors lay emphasis on the daily practice of piano along with what has been taught in the class. This daily practice is usually conducted in the absence of an expert. However, for the amateurs, it is really difficult to judge their performance in such an environment. Hence, they can't be firm about the accuracy of their play. At the same time, due to the high labour costs, it is impossible for the parents to hire a tutor to supervise the children during practice. It can be inferred that due to these unfavourable conditions, the amateurs may lose interest in the learning or there are chances for them to take an ample amount of time to learn piano than needed. To tackle this situation, we want to build an application that can support the piano players through virtual supervision.

This application will first record the student's play and then compare ("Determining Signal Similarities", n.d.) it with the instructor's play. After the comparison is done, the application will give a visualized report to the user. Through this report, the user will get to know his/her mistakes and will improve the play inevitably.

There are some technologies available in the market which supports the amateur piano players by improving their plays. Then why there is a need for

Play Piano? It's because other technologies have failed to meet the requirements what *Play Piano* wishes to achieve. Below is the summary of such technologies stating their major functions and their limitations.

Available Related Technologies

Name	Major Functions	Limitations
Piano Tutor	Staff identification exercise, note recognition, piano pitch practice, etc.	It does not offer any feature that can supervise the student's play. Moreover, it can only be played through the keys available on the user interface (virtual piano). But it cannot process/analyse the recording of a physical piano.
Stave Master	Music theory learning, a symbol dictionary, monophonic practice, etc.	It cannot record the (physical) piano's play.
Flowkey	Piano video tutorials, piano fingering practice	Lacks user interaction, cannot record the play and compare it with the instructor's play.
Piano Tutoring Class	Staff identification exercise, music practice	Have to access the midi keyboard to recognize the music

The above technologies are helpful to teach the staff reading, finger placement on piano, piano theory, etc. But they don't provide any features to give satisfactory feedback on the piano play. Moreover, the user has to use the virtual piano or has to access the midi keyboard to draw conclusions. Hence, we believe that there should be an application that can gather useful conclusions from physical piano play.

Project Objective

The objective of this project is to aid the piano players, especially the beginners, to improve their play. The piano seems easy for amateurs because they usually remain under the illusion that they are playing it correctly. As there is no tutor available all the time, it is difficult to remain on the right track. This application aims to give the platform so that the amateurs can recognize their mistakes and take their learning to the next level. A comparison between the student's play and the instructor's play will be made. Based on the comparison, a visualized report will be given to the user. As the application will be easy to use, we believe that it will increase the number of amateurs playing piano in the coming years.

Project Methodology

This application will implement, mainly, three methodologies, namely, front-end, back-end and audio signal processing.

The front-end will focus on the graphical user interface (GUI) so that the users can interact with the application using visual icons. Options like record the play,

compare the play, etc. will be provided to the user. We will be using softwares like Android Studio to meet the requirements.
Codes written for back-end will be executed on the server-side. We will be using Node.js platform to meet the back-end requirements. Additionally, codes written in Express, MongoDB and Matlab will also be executed. This will be done to maintain the data from the user and from the application.

To process the audio signals, we will be designing our own algorithms, along with the support of existing base codes. Primarily, there will be two algorithms. First, an algorithm that can convert the audio signal into the digital signal. This algorithm will be used on the recorded audio, which will be received from the user. This algorithm will also be used to convert the instructor's play's audio into digital signals. Second, an algorithm will be implemented that can compare two digital signals and provide necessary results for the report. This algorithm will be used to compare the user's digital signals (this can be procured from the first algorithm) and the instructor's digital signals (this can be procured from the first algorithm). Once the comparison is done, the algorithm will give the desired results which will be useful for the visualized report. To build this program, we will be using various libraries from Python ("Audio in Python", n.d.; "Audio Toolbox", n.d.), Matlab ("Audio and Digital Signal Processing(DSP) in Python", n.d.), etc.

Procedure

The procedure consists of four parts. Firstly, the application will record the student’s play. This will be stored in the form of audio signals. Secondly, an algorithm will be used to convert the audio signal to the digital signal. The same algorithm will be used to convert the audio signal to the digital signal of the instructor’s play. Thirdly, an algorithm will be implemented to compare these two digital signals. Finally, a report will be generated after analysing (“pyAudioAnalysis”, n.d.) the digital signals.

Project Schedule and Milestones

September

1. Meet with the supervisor, Dr. C. Wu
2. First Deliverable - Website and Detailed Project Plan

October

1. Blueprint of the mobile application
2. Research on related softwares like Android Studio, Matlab, etc.

November

1. Research on user requirements
2. Environment setup

December

1. Identify and process audio signals
2. GUI designs

January

1. Interim report
2. Demo application

March

1. Adding services for the user

April

1. Final implementation
2. Final report

References

Audio and Digital Signal Processing(DSP) in Python. (n.d.). Retrieved from https://www.pythonforengineers.com/audio-and-digital-signal-processingdsp-in-python/.

Audio in Python. (n.d.). Retrieved from https://wiki.python.org/moin/Audio/.

Audio Toolbox. (n.d.). Retrieved from https://www.mathworks.com/products/audio.html.

Determining Signal Similarities - Video. (n.d.). Retrieved from https://www.mathworks.com/videos/determining-signal-similarities-97063.html.

pyAudioAnalysis. (n.d.). Retrieved from https://pypi.org/project/pyAudioAnalysis/0.1.3/.