

Piano Training Simulator

Final Report Summary

Group 31 - Abel Abraham, Beyza Soylu, Edward Plesa, Ryan Jasiak

Overview

This document is meant to summarize the efforts our group made in the development of Group 19's Piano Training Simulator from Fall 2018 as described by their development report. To summarize what the project is supposed to accomplish as per the previous group:

This project has two main parts. The first part is a physical keyboard that plugs into a computer. It will simulate the look and feel of actual piano keys. The input from the keyboard will be interpreted by the second part of the project: a software application installed on a user's computer. This software application will virtualize sound that mimics an actual piano and outputs it to the headphones or speakers. It will also analyze the user's input compared to what song they are supposed to be playing in real time. It will give live feedback of mistakes and give overall results to help the user improve and practice.

Project Deliverables

Up to this point, there have been two major demo/ releases of our implementation of the Piano Training Simulator. To quickly note the main functionalities that the project should have: The vision the previous group had for the GUI necessitated having sheet music with a slider to show at which point the song is currently playing. They also had additional elements regarding sections for instructions about exercises, overall performance, and level of completion. The document also mentioned having subsystems for managing user data, music sheet data, and training data (performance analysis).

Our first release provided a lot of the core functionality regarding the algorithms for the sliding notes towards their respective keys, and playing a couple of hard coded songs through our software. The tracks for each note were not perfectly aligned, as well as having no sound produced when pressing any note on the connected MIDI keyboard.

In the second release, we added some quality of life features including the framework for user login, and a way to upload and pick songs from a list dynamically. The tracks for notes were still being worked on, but we managed to get the sound working for each key press.

The final prototype involved some work done on the GUI styling for a cleaner and more polished experience. The track and note alignment was greatly improved. An arbitrary scoring system was also introduced for this. When compared to the overall finished design of the project, we still had not implemented a database for user profiles, a scoring and analysis system for user performances, sheet music with a slider, and on screen instruction for exercises.

Testing

Testing was done thoroughly and was stressed to ensure that we created a quality product adhering to the principles that Fall '18 Group 440 detailed in their documentation. We decided to test four main sections of the project: 1) MIDI Controller Connection, 2) Note Sliding Algorithm, 3) .midi File Parsing Algorithm, and 4) Song Scoring Algorithm.

For the MIDI Controller Connection, tested the reliability and usability of the connection to a MIDI Controller. For the Note Sliding Algorithm, we tested the reliability and usability of the algorithm that sends the animated notes based on .midi data down a path towards the piano

GUI in the application. For the .midi File Parsing Algorithm, we tested the reliability of the algorithm used to parse the .midi file data. For the Song Scoring Algorithm, we tested the reliability and usability of the algorithm that scores the users' piano playing.

Inspection

To ensure a consistent, familiar style of code between all contributors, a set of guidelines were created to ensure that future or retrospective work is easily readable and rational. These guidelines include correct use of camel casing, descriptive variable and method naming, ample use of comments for non-descript sections of code, and follows consistent spacing and indentation. Each member of the team followed a set of procedures in order to inspect another contributor's work to the project code base. These are as follows: each member will pick any random candidate for inspection, the candidate will notify the inspector of the code they have contributed to the project, and the inspector will go over the items to be inspected and alert the candidate of any breaches to the guidelines and mark them down. The results of following these guidelines and procedures led to having an easily maintainable project between each contributing member.

Project Issues

The Piano Training Simulator project was one in which the group reached a common consensus in that it proved to be challenging and frustrating at times, but ultimately was a rewarding experience for the entire team and we each had the remarkable duty of using our strengths to make up for others' weaknesses.

We had an urge to create a well thought out and complete implementation of Fall '18 Group 440's (Srinivas Lingutla, Shawn Cody, Tapus Patel, Ji Xiao) documentation, and once we got started we faced many roadblocks, but ultimately it was through our teamwork that we were able to succeed. I think some of the biggest challenges we faced were being able to meet up with each other (in-person and online) and coordinating times to communicate and work with each other in which members were mutually free. Beyond that, our group was able to work well.

