ECE 5725 Final Project Progress Report 2
Piano Game
Guanqun Wu gw284, Zhaopeng Xu zx273
Thursday Lab
Submission Date: 26 Nov 2019

Current Progress

Since we're in Thursday Lab, our last lab report was on Friday 22 Nov and we didn't have much time (Friday - Mon) between then and this report, but we've made some progress on the software side.

Score File

```
Twinkle Twinkle Little Stars
60
4 4

T C4 1
B C3 1
T C4 1
B C3 1
T G4 1
B G3 1
T G4 1
B G3 1
```

We create a score file to represent a musical score. The initial metadata on the score file includes the name of the piece (Twinkle Twinkle Little Stars), followed by the initial beats per minute (60) and the timing (4/4).

We then have a newline before and after every bar (set of notes summing up to 4 beats in this case, number of notes vary by timing). Each note is on its own line and has 3 parameters. T/B represents the treble or bass clef, C4 represents the note itself (- would represent a pause) and its frequency can be directly mapped to the frequency of the waveform generated through a lookup table. The 1 at the end means that this note is of a single beat. This number could be 0.25, 0.5, 1, 2 or 4 (depending on the timing). We make sure that both the treble and bass clefs contain the correct number of notes per bar.

We permit changing the pace through a CHANGE PACE <new_pace> line and changing the timing through CHANGE TIMING <new_timing_1> <new_timing_2>.

We provide a file music.py that contains classes (Score and Bar) to model the score and the bars so that we are able to easily draw reason about the music and are able to easily display the music on screen.

We also provide a file components.py that contains Btn, Line, Text, Image and Stage classes that allow us to easily reason about and draw stage elements that we then use for the training mode and the practice modes.

Work for the coming week

We plan to find a grand piano at Lincoln hall and record the piano sounds in a wav file. We will then isolate all the notes that we're interested in and make them into their own wav files. We will then use pydub to play the sounds onto a USB speaker that we acquired. Trying to get the correct waveform onto the speaker is using up too much time and we think we should spend that time on the UI and the game itself then come back to the speaker in the future.

We've also thought of a way to wire up the buttons in such a way that it'd be easy for a human player to press the buttons while also allowing us to connect all the buttons to the port expander. We'll be trying to interface with the port expander to make sure that everything works.

Progress with respect to initial plan

We're somewhat behind with respect to our initial plan, but we plan to make up for that by working hard over the Thanksgiving break :(