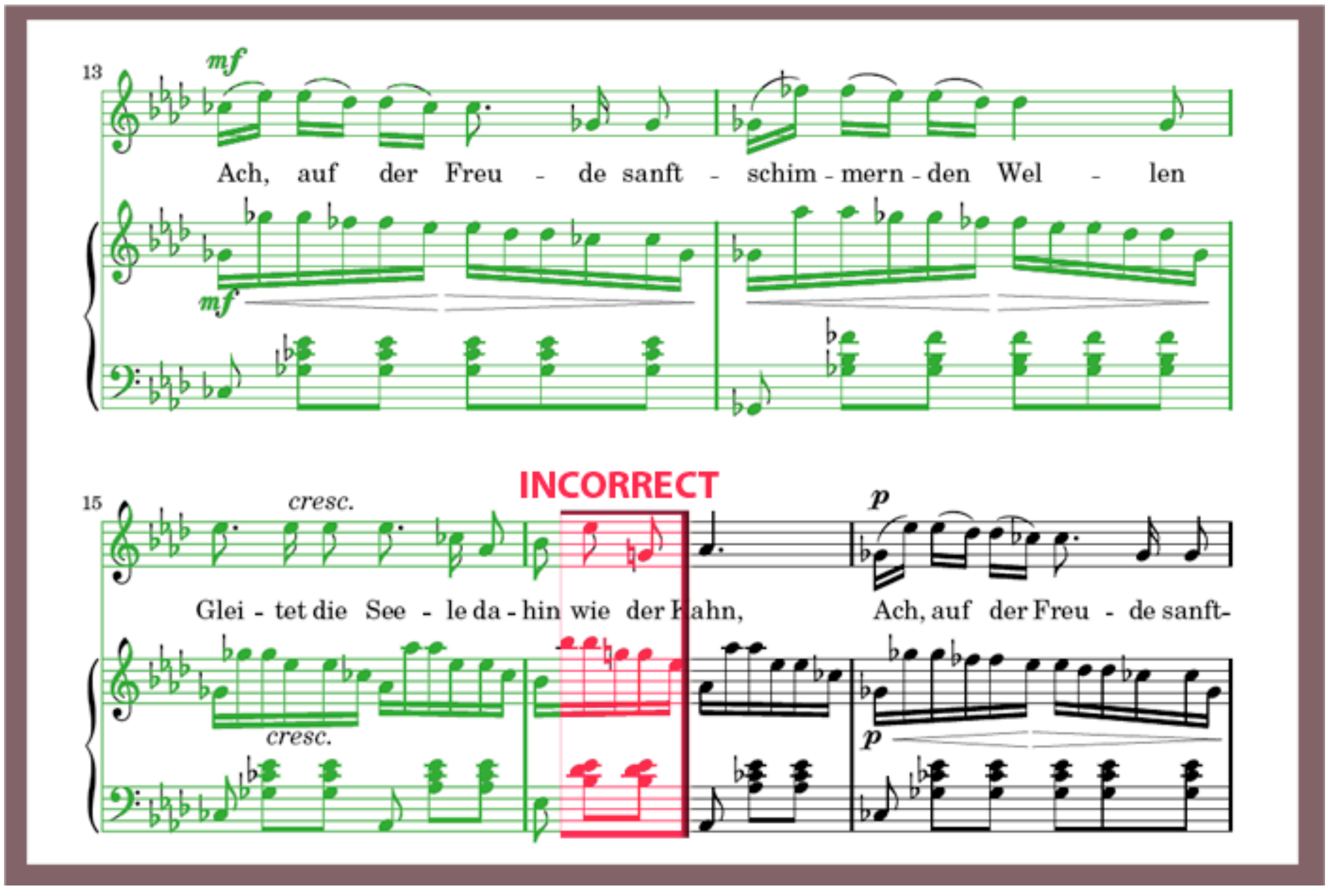
EMIT ELECTRONIC MUSICAL INSTRUMENT TRAINER

About

EMIT is a piece of software that follows along with digital sheet music as an individual practices his instrument. The program currently only has one mode. The program analyzes the number of mistakes and prints the statistics of errors after the completion of the musical piece. Currently we manually adjust the sensitivity in the code, however our ideal UI would allow the user to simply adjust it before he begins playing a song.

EMIT was initially going to be implemented for the trumpet, however the pitch recognition algorithms work well enough with voice and piano. We plan on continuing this project and potentially expanding EMIT for use with every instrument. For more on possible add-ons see our "Expansions" section


13
mf
Ach, auf der Freu - de sanft - schim - mern - den Wel - len
15
cresc.
INCORRECT
p
Glei - tet die See - le da - hin wie der Kahn,
Ach, auf der Freu - de sanft-

The figure above is an example of the graphical interface of EMIT.

Schnella

Eric Schlossberg '10 Computer Science and Jeff Carbonella '10 Computer Science joined forces for this project as the group Schnella. Our focuses shifted shortly after beginning work and each of us basically worked on what we originally thought the other person would be working on.

Jeff Carbonella's main focus in this project became ensuring that the essential areas of musical notation are integrated into EMIT and the digital sheet music data type

Eric Schlossberg's mainly worked with the Fast Fourier Transform and other digital signal processing nuances for detecting pitch.

Motivation

Learning a new musical instrument can sometimes be a daunting task. This task can be made even more difficult when an individual is using sheet music for a song he does not know. This software would help the individual know that he is making a mistake so that certain mistakes do not become habitual. It would be most useful for instruments where the difference between playing different notes is minimal, such as a piano and guitar (as opposed to instruments where different notes are specific seemingly unrelated combinations of keys i.e. trumpet and flute). It would also be extraordinarily useful in instruments with continuous frequency choices (i.e. the trombone) and instruments in which the notes are determined by thought (i.e. voice and sight singing).

Testing/Equipment

Equipment:

Piano (Woolworth Center)

Voice (Eric's)

[Note: We initially thought that we would use this software for the trumpet but the pitch detection worked better with voice and the piano.]

Software:

We decided to creat our own musical notation sheet music data type rather than using an open-source one such as MusicXML because of the simplicity desired for later implementations with FFT. We didn't have time to produce software to convert to and from MusicXML for portability purposes, which would also allow the use of Music Optical Character Recognition from open-source software such as Audiveris. With this OCR software, individuals can scan music into the MusicXML format which can then be converted to our custom made data type. We hope to continue working on this project in the future and ultimately add this functionality.

We wrote the majority of the code in Java. However, due to the ease of the Chuck fft function compared with the fft.java program we originally wrote, our fft program was written in Chuck. Because of the ease of running multiple Chuck programs at once, we were able to time synchronize the fft program so that there is always an fft program hovering over different parts of the fft data searching for peaks.

To run in Mac osx:
1) After downloading Emit, open two terminal windows
2) In one window execute "chuck fft/fft* >& data".
3) In the second window execute "Java EMIT"

Songs are written by typing in the standard BPM for that song and then alternating note name (i.e. A4, E3) with number of beats for that note. See MaryHadALittleLamb.txt.

Click here for all of our source code

Testing:

The bulk of the testing has been done by team Schnella. We thoroughly tested the pitch detection and tweaked it to ensure that the software works according to the original functional plans. The software was tested with a tuned piano and voices with various error sensitivities.

Unfortunately, we haven't gotten EMIT working at the level necessary for commercial use (still need unix calls to fire up sheet music and pitch detection) and thus it would be pointless to ask users how they like the software itself. However, the quality of the underlying application (i.e. how effective Emit's pitch detection and error checking?) was tested by Eric Schlossberg.

Expansions

Expansions/Add-ons:

- Short-Term Improvements
  1. Allowing a broader range of musical notation
  2. Adding dynamics as an additional testing feature
- Long-Term Improvements
  1. Detecting multiple notes at once
  2. Allowing changes in tempo in the music
  3. Allowing musician's tempo change based on expression

References

Previous Project Proposals

- Team Schnella - First Project Proposal
- Team Schnella - Second Project Proposal

Additional References

Notes on FFT with many references to papers listed at the bottom - http://mathworld.wolfram.com/FastFourierTransform.html

More notes on FFT and psuedo-code - http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/dft/

Audiveris, open-source Music OCR - https://audiveris.dev.java.net/

MusicXML, open-source music notation file format - http://www.musicxml.org/xml.html