Harmonious:

Pitch Training for Musicians

Scott McNulty

Neumont University

Summer 2013

**Define Project Success**

Music is an art form with mathematics and physics at its core. As sound waves travel from the throat of a singer or an instrument, their analysis reveals many different qualities including pitch, volume, and timbre. Musicians train for years, often lifetimes, to hone their skills, especially skill in recognizing and producing notes that are on pitch. The human ear can detect variations in pitch of less than 1 Hz, so performance quality musical expression must be very precise.

This system will be able to detect and analyze musical notes sung, played, or synthesized. As a training tool, it will be able to analyze these notes in real time to show a user if their voice or instrument is off pitch and help them be in tune. The functionality of the system will start with single notes, extend to short musical phrases, and end with whole songs.

The target hardware platform is Windows 7/8 on a laptop with possible porting to Windows phone and tablet devices, all in C# .NET. In this environment, there are no APIs for Fourier Transforms, a mathematical way to convert time to frequency and vice versa. There are many subtypes of this algorithm; each is interesting and challenging to implement. The main sampling and analyzing parts of the system will be built from scratch using this family of algorithms.

**Target Users**

The target users for this product are music instructors and their students. The product will be a valuable training tool for one of the critical objectives of music training- accurate pitch determination and expression vocally or instrumentally.

**Project Backlog**

\*All stories apply to vocal or instrumental music unless otherwise noted.

1.) A vocal student or musician would like to determine how accurate their ear is during warm-up exercises. They will sing or play a pitch and the system will give feedback on its accuracy based on its closest frequency.

2.) A vocal student is training to have perfect pitch, meaning they would like to envision a note in their mind and sing it accurately without any auditory prompting. The student enters how many random notes to try and the range of those notes. The system presents the name of a note to the student; the student sings the note and receives accuracy feedback based on the correct frequency.

3.) A student would like to see a visual representation of their performance as it is in progress. As the student varies the pitch of their note, the system provides a visual representation of pitch performance.

4.) A music instructor would like to evaluate a student’s pitch performance on a musical phrase or song. The system captures the student’s performance; it analyzes the performed music and displays accuracy feedback for the instructor and student based on closest frequency comparisons. The instructor saves the session for further review.

**2-Week Plan**

**Week 1**

|  |  |
| --- | --- |
| Monday | * Start implementation of Fourier Transform core. |
| Tuesday | * Research and decide format(s) to use for the audio files and the performance information files. |
| Wednesday | * **Create three sample performance information files.** |
| Thursday | * **Create three sample sound input data files.** |
| Friday | * Procure sample musical scores for solo vocal performance. * Procure sample musical scores for solo instrumental performance. |
| Saturday |  |
| Sunday |  |

**TBD – by appt. this week**

* Interview a vocal coach SME to determine necessary feedback parameters.
* Interview a music instructor SME to determine necessary feedback parameters.

**Week 2**

|  |  |
| --- | --- |
| Monday |  |
| Tuesday | * **Implementation of basic Fourier Transform core complete.** |
| Wednesday | * Start implementation of warm up pitch utility (this will be a good starting utility to test the effectiveness of the FT core). |
| Thursday | * **Complete basic GUI for warm up pitch utility.** |
| Friday |  |
| Saturday |  |
| Sunday | * **Pitch utility compares input note to averaged interval and calculates the flat/sharp difference.** |