Mozart’s Ear

ECE1778 Final Report

Andrea Stuart (Apper)
Jason Deng (Programmer)
Luis Vázquez del Mercado (Programmer)

Final Report Word Count: 1957
Apper Context Word Count: 499

Table of Contents

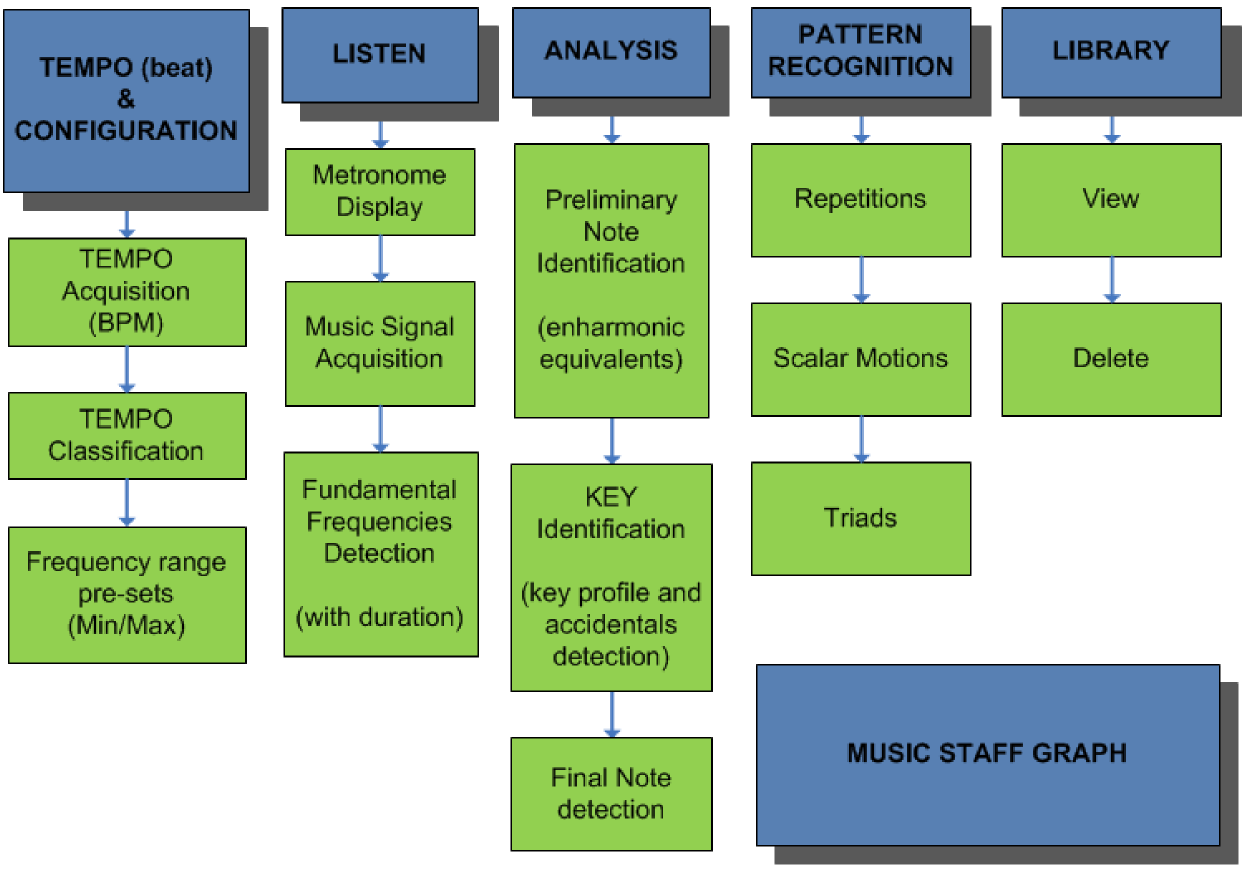
1. Introduction

Mozart's Ear is a music transcriber. Analogous to speech-to-text software, it does music-to-notation of unaccompanied melodies. Music transcription is very useful for musicians but difficult to learn and laborious to execute. Once a melody is transcribed, the app is able to do simple melodic analysis, identifying features such as the key of the selection, motives, moments of exact repetition, triads and scalar motion. Identification of these melodic features is useful for people trying to engage intelligently with the music either when memorising a passage (for performance), trying to understand its construction (for analysis by a composer), or naming features to discuss their effect or significance (for the musicologist).

Simply, Mozart's Ear puts into the palm of users the ability to hear like Mozart without spending hours in a practice room.

2. Overall Design

Mozart's Ear is implemented in different phases as shown below:



2.1 Configuration

Users perform the following configuration prior beginning playing music:

- User taps the tempo to be used. Tempo is acquired in BPM and classified.[1]
- Specific frequency range can be entered, narrowing range improves notation accuracy.

2.2 Listen

Along with the recording screen (where a metronome and a visualizer are shown), the following occurs:

- Music is recorded, saved in a temporary file.
- After recording, the spectral analysis for note frequency and duration begins. A sliding FFT[2] is used on the audio data that is modified by a Hanning window to generate a short time Fourier transform (STFT).
- Pitch detection uses a threshold gate (in decibels) to filter out noise and two mechanisms to identify valid frequencies. The smaller frequency of the two results is taken.
    - Find the first peak
    - Find the median of the frequency distances between all the overtone peaks
- Results outside of the configured frequency range pre-sets are eliminated.
- Note duration is identified by maintaining a count of contiguous frequency peaks in the STFT.
    - All identified durations are an integer multiple of the distance between FFT sliding windows.

2.3 Analysis

In this phase:

- A preliminary note list is generated by interpolation. This may contain enharmonic equivalents (same frequency with two different possible notes, i.e. black notes on piano). The list includes rests (no music played).

---

[1] Percy Scholes, et al., "tempo," *The Oxford Companion to Music*. *Oxford Music Online*, Oxford University Press, http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/opr/t114/e6699 (Accessed 8 April 2013).

[2] Piotr Wendykier, "JTransforms," http://sites.google.com/site/piotrwendykier/software/jtransforms (Accessed 8 April 2013).

- Preliminary list is analyzed and all continuous notes are merged as a single note. Notes with duration less than 1/16th of a beat are considered noise and eliminated from the list.

- A key detection algorithm is performed, based on key profiling from criteria provided by Andrea and by detecting pitch classes as present or not (pitch weight) as a good approach for small music passages (combination of the Krumhansl-Schmuckler and the Longuet-Higgins/Steedman approaches)[3] as follows:
    - Weights of natural notes are calculated
    - Weights of enharmonic equivalents are calculated
    - Calculate key profile weight based on natural notes and enharmonic equivalents possible for each major key
    - Select key with greatest profile weight (e.g. if no enharmonic equivalents were detected, key is equal to C)
    - If no key is detected, accidental identification (# or b) is performed and key profile weights are updated based on accidental proximity criteria with natural notes present
    - If key remains undetected, a deduction based on likelihood (circle of fifths)[4] is performed

- If key detection is impossible, user is notified and key is forced to be C Major (accidentals forced to #)

- Note list is updated to remove enharmonic equivalents with accidental determined. Note duration is updated to be shown in terms of beat duration (e.g. 4 beats, 1 beat, ¼ beat, etc.)

2.4 Pattern Recognition

The identification of music characteristics from the music score translates to a pattern recognition problem. We identify the following characteristics:

[3] David Temperley, "What's Key for Key, The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered," *Music Perception: An Interdisciplinary Journal* 17 (1999): 65-100.

[4] Arnold Whittall, "circle of fifths," *The Oxford Companion to Music*, Oxford University Press, <http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/opr/t114/e1421> (Accessed 8 April 2013).

- *Triads*: Ascending or descending sequence of 3 notes that skips every other line or space.[5]
- *Scalar motion*: Ascending or descending sequence of at least 4 notes with a note placement pattern of space, line, space, line.[6]
- *Repetition*: An exact repetition of at least 3 notes.
- *Motives*: A repeating pattern of 4 to 10 notes that may be translated on the staff.[7] First differences between pitches don't have to be exact.
    - An autocorrelation is used to identify how well the pattern matches

2.5 Library

The user can save the current notation on a CSV file for future reference. Only key, tempo, notes and duration (as a fraction of a beat) are saved. Pattern recognition algorithm is re-executed every time the user views the notation, reducing file size.

User can also create a CSV file and open the file with Mozart's Ear, viewing the notation which can then be run through pattern recognition.

3. Statement of Functionality and GUI Screenshots

3.1 Application Functionality

The following statements are applicable to Mozart's Ear functionality:

- Pitch detection is reasonable when a clear sound with a clear fundamental tone is provided. We used synthesized tones; cello required too much calibration
- Rhythm detection is somewhat wonky. Partly because the metronome feature does not give a hyper specific beat in time (equivalent analogy of flabby conductor) making it easy to human error to give imprecise rhythmic performance. An audible beep would have helped but was not pursued due to time constraints.

---

[5] "Triad," *Grove Music Online*, ed. Deane Root. http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/grove/music/28347 (Accessed 8 April 2013).

[6] William Drabkin, "Scale," *Grove Music Online*, ed. Deane Root. http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/grove/music/24691 (Accessed 8 April 2013).

[7] William Drabkin, "Motif," *Grove Music Online*, ed. Deane Root. http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/grove/music/19221 (Accessed 8 April 2013).

- Key detection has limitations, but working within those limitations it is accurate:
    - does not identify minor keys
    - does not cope well with ambiguous keys or key modulations
- Melodic analysis is capable of identifying the desired traits: scalar motion, triads, repetition, and motivic development. More traits could have been added with more time, but we are satisfied the concept has been proven.

3.2 Graphical User Interface

The following exhibits the GUI in the order that they would typically appear to the user.

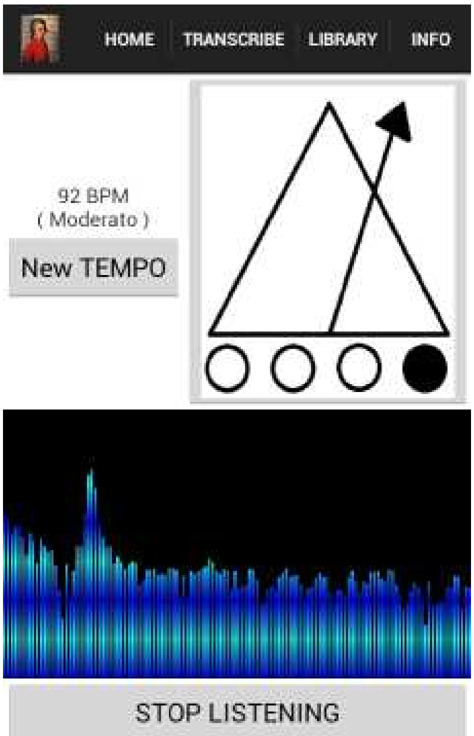
Home Screen

Mozart greets the user on the home screen. From here the design emphasis was on minimizing the number of taps required from users to access the music transcriber (main feature).

To aid consistent and easy app navigation, a constant bar menu is displayed at the top of all screens, giving access to: the home screen, transcriber, library, and general application info.

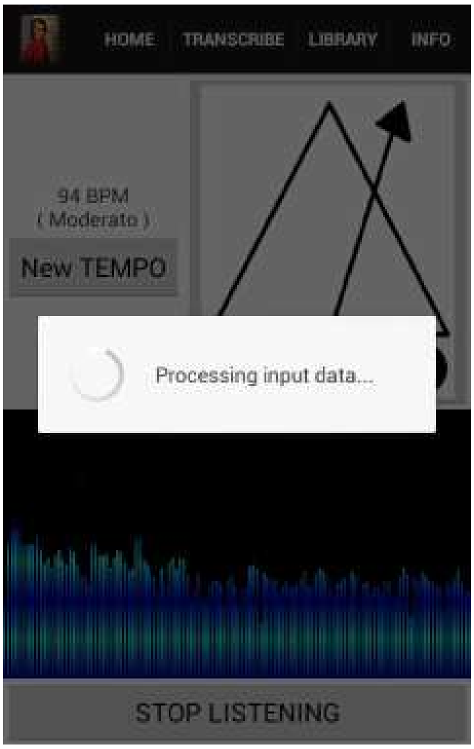
Configuration

The user taps tempo and specifies the frequency range to filter.



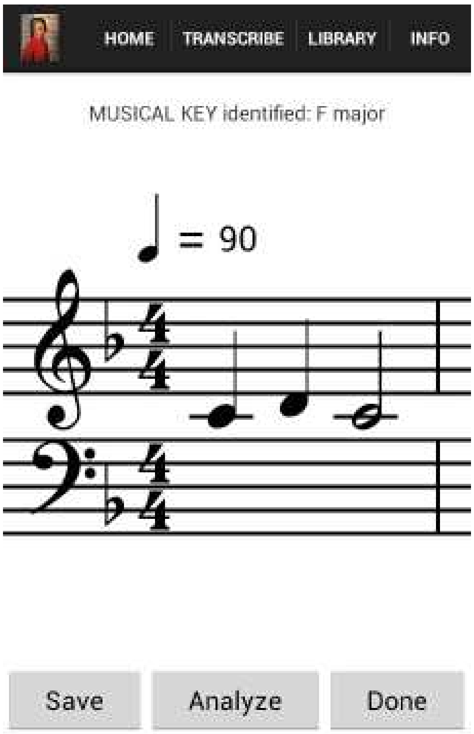
Music Recording

The metronome and the spectral visualizer are displayed while the user records the performance. The horizontal axis is in hertz and the vertical axis is in decibels.



Processing Recorded Data

Processing time is linear with the recording length.



Scrollable Music Score

The music key and notes are identified and displayed in the score according to correct musical notation practices.

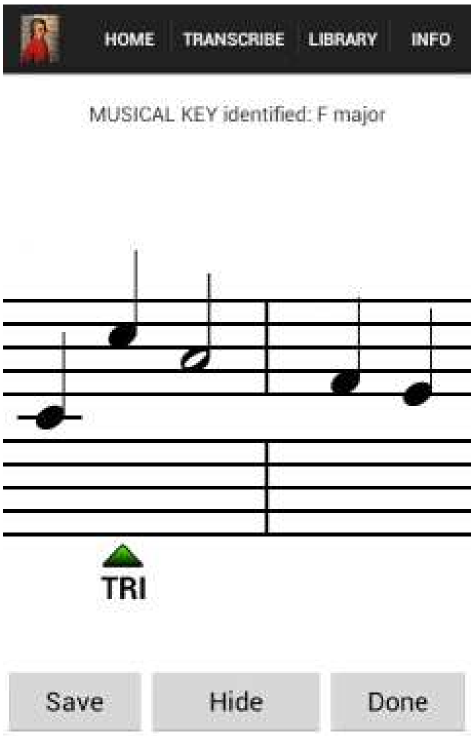
Music Characteristics

Characteristics are identified by the tag consisting of an arrow and an abbreviation.

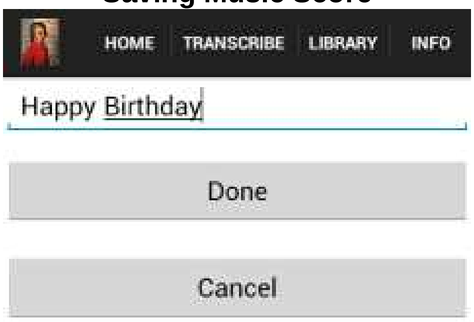
TRI - triads
SCA - scalar motion
REP - repetition
MOT - motives

The number below the abbreviation represents the first note order at which the instance of the characteristic first appeared (in this example, the first note of the passage).

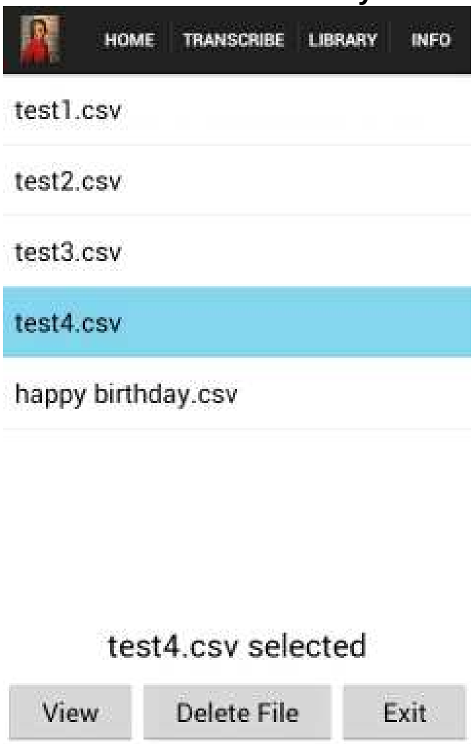
Music Characteristics



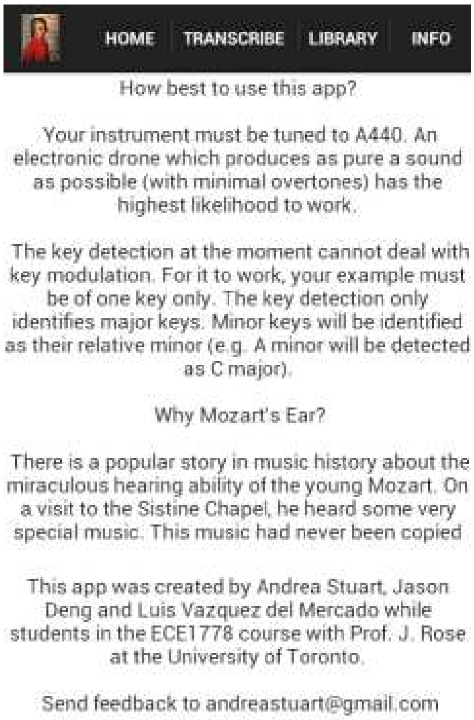
Saving Music Score



Music Score Library



General Information



4. Learning Experience

In this section, we explore the lessons learned working together and other compromises that can be made in design as to not require programmer or apper to dive fully into the other's field of study.

4.1 Communication and Perspective

The same words are interpreted differently according to the listeners' technical background; the thought process is organized differently depending on context and experience. We have experienced firsthand, the necessity to reiterate ideas in the simplest terms possible without obscuring contents. We would like to restate the obvious that *a picture is worth a thousand words*. In addition, *frequent and short discussions* allow each member to better focus and organize their thoughts on a foreign concept.

4.2 Development of Detection Algorithms

Key detection is at best an estimate since it is indeterminate for any algorithm and the human ear, for example, if the music contains an equal number of every note. Mozart's Ear would have benefitted from additional comparisons to existing works such as the MIRtoolbox.[8]

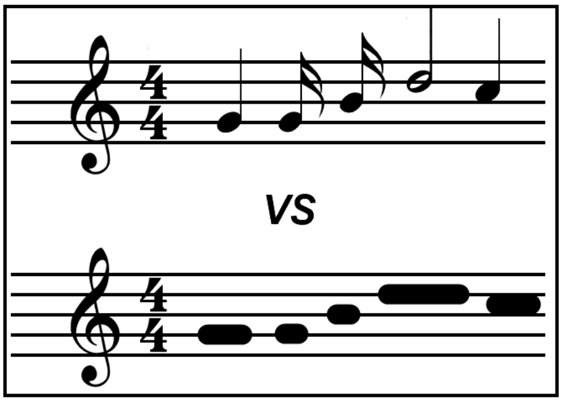
Pitch and rhythm detection is difficult. It turns out customizing the algorithms to specific instruments would be the best approach for the following reasons:

- support for all the notes in the low and mid-range is excessive and error prone
- different instruments have varying amounts of overtones, undertones, and strength in frequencies

The spectral analysis algorithms could be improve via a two-pass analysis. The first pass analyzes valid frequencies to identify pitch and interpolate missing frequencies. The second pass would identify all rests and refine the interpolated frequencies. We did not have enough time to implement the second pass, resulting in poor rhythm detection.

4.3 Music Score Notation

To reduce the need to follow all music notation requirements, the score could be abstracted as illustrated below. This would reduce the need to teach music theory between group members, allowing more focus on feature improvement.



[8] University of Jyväskylä, *MIRtoolbox,* https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox (Accessed 8 April 2013).

5. Group Contribution

Andrea	GUI ● Developed mock-up design ● Exhaustive testing to detect bugs Algorithms and analysis ● Algorithm for key detection and setting criteria for determining difference between ambiguous notes with Luis ● Developed test examples for algorithms (composed music) ● Algorithm for motivic recognition with Jason ● Discussed app with, and solicited feedback from colleagues (1 musicologist, 1 ethnomusicologist, 1 music theorist, 4 performers, 2 composers, 2 music educators, 1 music psychology/cognition specialist) Communications ● Provided programmers recommendations with initial sources to learn basic music, aided with necessary music concept learning ● Provided Jason with information about notation and visual images
Jason	Hardware ● Microphone input recording GUI ● Spectral visualizer ● Music score API ○ display of music notes and notation ○ display of music score characteristics Algorithm and analysis ● Spectral analysis for frequency and time identification ● Music characteristic analysis Communications ● Component integration discussion and planning with Luis ● Music theory and features discussions with Andrea as well as bug/improvement feedback

Luis	Communications • Component integration discussion and planning with Jason • Music theory discussions with Andrea GUI • Welcome screen • Action bar menu • Tempo acquisition (user tapping) and classification • Metronome display • Library • Info screen Algorithm and analysis • Preliminary note identification (enharmonic equivalent removal) • Note merging functionality • Key detection algorithm • Accidental identification algorithm • Final note identification with duration as fraction of tempo beats

6. Apper Context

The ability to transcribe music and analyse its components is useful for western art music practitioners. It takes a long time to learn this skill. In Canada, university music programs require a minimum ability in ear training (of which transcription is a fundamental component) upon entering bachelor programs and typically require even further study during the degree. Undergraduate course requirements in ear training usually involve, during the first two or three years of study, classes which meet for several hours throughout the week (plus variable amounts of individual study depending on students' abilities). Just as calculus is a fundamental course for physics or engineering students, so too is ear training for music.

The dream is that a music transcriber app like Mozart's Ear could do for music what calculators do for math. Though mathematical calculations and music transcription can both be done the old fashioned way with pen and paper, a handheld device means a lowered entry requirement of time and labour from the practitioner, thus freeing up people for more complicated endeavors.

When is transcription used? Cross-cultural music exchanges involve teaching one style of music to musicians trained in a different practice. An example of this would be the symphonic violinist learning Celtic fiddling. In this situation transcription concretizes for the western art musician non-western oral and aural traditions. Transcription also allows people to remember music. Composers will often improvise at their instrument before writing down their ideas. Ethnomusicologists often do field work to study the music of other cultures. On these trips they take hours of field recordings which are then transcribed later so that a sense of the music being studied can be communicated in academic writing. In all three of these instances, transcription is used by the musician, the composer, and the ethnomusicologist.

Even more generally, transcriptions are just easier to deal with than audio recordings. Remember when you used to have to study from textbooks? Imagine if that textbook had been an audio book and you had to rewind and fast-forward whenever looking for a particular passage. Glancing at a page, titles of different sections orient the reader, one need not remember how music unfolds linearly in time before situating themself. Without notation, music cannot be experienced non-linearly nor independent of the time domain —an hour long symphony takes an hour to hear. When you are hearing a motif, unless you have been paying close attention and you already know the music well, you cannot always tell how many times you've heard this motif before this moment nor how many more times you will hear it in the music yet to happen.

Melodic analysis, is easy for the human ear but not so for software. As stated in the introduction above, identification of melodic features is useful when trying to engage music intelligently. Analysis is an entire field within music and not restricted to the features this app identifies. This app does show, however, the potential for this work to be done by computers.

7. Future Work

Mozart's Ear is presented as a prototype for automatic transcription of an unaccompanied melody of a given instrument. Therefore its feature list is insufficient for general widespread use. Following is a list of possible extensions and improvements:

1. Metronome ticking sound
    - the sound must not interfere with note detection; or could be sounded only when earbuds
2. Customize spectral analysis algorithms for specific instruments
3. Ability to export the music score and audio recording
    - via common formats (musicXML, flac)
4. Improved music key detection
    - minor key support
    - key modulation (changing music key dynamically)
5. Music score editor
    - allowing modifications of pitch and rhythm
    - with audio recording playback
6. Automatic flexible beat detection for tempo
    - tempo is the only piece of music information manually entered; it is a good target for automation; flexible beat detection would allow the performer to naturally speed up and slow down
7. Support for harmonies and polyphonic music
    - this is difficult and, currently, would require extensive research

8. Marketing and Entrepreneurship

The Mozart’sEar team is not interested in being involved with developing this app further. We are happy to allow a business class to use it for a project. We request the codebase be made open source to all (no time restriction).

References

Drabkin, William. "Motif." *Grove Music Online*. *Oxford Music Online*. Oxford University Press, http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/grove/music/19221 (Accessed 8 April 2013).

----. "Scale." *Grove Music Online*. *Oxford Music Online*. Oxford University Press, http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/grove/music/24691 (Accessed 8 April 2013).

Scholes, Percy, et al. "tempo." *The Oxford Companion to Music*. *Oxford Music Online*. Oxford University Press, http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/opr/t114/e6699 (Accessed 8 April 2013).

Temperley, David. "What's Key for Key, The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered." *Music Perception: An Interdisciplinary Journal* 17 (1999): 65-100.

"Triad." *Grove Music Online*. *Oxford Music Online*. Oxford University Press, http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/grove/music/28347 (Accessed 8 April 2013).

University of Jyväskylä. *MIRtoolbox* https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox (Accessed 8 April 2013).

Wendykier, Piotr. "JTransforms." http://sites.google.com/site/piotrwendykier/software/jtransforms (Accessed 8 April 2013).

Whittall, Arnold. "circle of fifths." *The Oxford Companion to Music*. *Oxford Music Online*. Oxford University Press, http://www.oxfordmusiconline.com.myaccess.library.utoronto.ca/subscriber/article/opr/t114/e1421 (Accessed 8 April 2013).

Useful development links

The following table shows resources that helped develop Mozart's Ear:

Resource	Mozart's Ear functionality usage
An audio recorder and editor: http://audacity.sourceforge.net/	Used to view the spectrum of instrument notes
JTransforms: Java FFT library https://sites.google.com/site/piotrwendykier/software/jtransforms	Used in the listening activity for pitch detection
OpenCSV: Library to read/write files in CSV format http://sourceforge.net/projects/opencsv/	Used in the music score library