Training Absolute Pitch
Gloria Ma
Yale University, Spring 2014

Abstract
People often struggle with the mastery of tedious, mechanical skills, such as memorizing foreign language vocabulary or drilling physical movements into muscle memory. We focused in this project on the question of how to devise effective training software that explicitly encoded ideas of efficient learning, applied to the specific goal of developing absolute pitch in adults. The challenge of this project, and motivation for it, lies in the fact that there is no scientific evidence that this is even possible, and in fact essentially all of the literature agrees that it is not. However, we noted a lack both of compelling biological explanations for why this should be the case, as well as of any experimental evidence that actually put adults through a well-designed ear training regimen and then tested their progress. Our approach thus consisted of two main stages of design, to address both efficacy as the primary concern, and then efficiency. First, we considered various criteria to design a legitimately useful exercise, with the particular goal of subverting the average non-possessor of absolute pitch's immediate tendency to think in terms of relative pitch (whether consciously or subconsciously). Upon arriving at an exercise design, we then considered efficient learning strategies to incorporate into the design of the software, hoping to enhance its usefulness by performing analysis on a user’s responses to both dynamically modify his training experience and present him with information on his tendencies. Both of these stages were guided by significant personal experimentation with actual ear training throughout the course of the project to confirm their effectiveness. Our final product was a piece of training software for users to develop their absolute pitch recall.

Introduction
In learning any complex skill, one must start with the basics. For example, in learning an instrument, a student usually starts with how to produce a single note with a nice sound, and then moves on to scales and arpeggios, and then onto simple pieces and so forth. In learning a sport like tennis, one first learns how to hold the racket, and learns the different motions for a serve, a forehand, a backhand, and so on. Each complex skill is in fact a combination of many smaller skills, each of which can be broken down further into even smaller skills: with the violin, for example, even the skill of “producing a single note with a nice sound” can be further broken down into mastering a pleasant vibrato in the left arm and developing sufficient bow technique in the right arm, the latter of which can be further split into developing a proper bow hold, as well as muscle memory of the motion of the arm, and so on. At some point, a serious student will necessarily get down to some atomic unit, a
fundamental building block of the skill she is trying to learn, that she must then learn via rote practice: simple, deliberate repetition.

The development of all complex skills thus relies heavily on this fundamental level of skill-building. Yet, it is not a topic that is necessarily given a great deal of focus or time by most instructors, because by nature it consists of precisely the most tedious tasks for any learner. The average budding young soccer player wants to play a game, and not practice dribbling the ball alone around a set of cones for an hour - and then, even when she at last commits to doing these drills out of the knowledge that it is important to develop her foundation, she is left with only very vague guidance on how exactly to go about it to learn it well and efficiently. “Do that twenty times,” or “keep repeating it slowly and gradually go faster” are often about as detailed as directions get for students. There is little emphasis on giving concrete strategies or methods to ensure that the student’s practice is efficient: spending the right amount of time on the right specific aspects of their game or their art, instead of a bored and unfocused going-through of motions that rarely if ever actually produce solid results. There is an adage that “practice doesn’t make perfect; perfect practice makes perfect” - and yet very often, students are not given any hints as to how to “practice perfectly.”

The failure to develop effective strategies for rote learning can have far-reaching consequences. Certainly, in many cases, it makes it impossible for someone to really progress in a skill he wants to learn, and he may become frustrated as he finds himself bogged down with rote learning issues and unable to move onto the more enjoyable stages of learning. Over a long period of time, this will undoubtedly lead to decreased self-esteem as the subject attributes his lack of success to inherent ability rather than inefficient practice methods. This can be particularly damaging because rote skill-building is by nature a very generalizable concept: the process of coming up with an effective regimen to drill any mechanical task deals with the questions of determining what to focus on, what feedback to get from each attempt, the proper rate and amount of practice, and so on. These same principles and criteria apply to any such task, even as the context varies widely. Thus, the person who never falls into an effective method or rhythm of practice in one area will likely have difficulties in other areas. However, on the flip-side, discovering a way to practice one rote skill effectively will likely dramatically improve efficiency of practice in many other skills. Our goal with this project was thus to develop software to help people learn a specific rote skill, in the hopes that it may inspire them to think critically about how they are practicing other skills and perhaps see ways in which that process can be improved.

**Background**

Absolute pitch, or perfect pitch, is the ability to recall and recognize pitches by name (e.g. A, A#, and so on) without any context. It is often discussed in contrast to relative pitch, in which a listener
is able to name notes in relation to another note, by determining the interval distance between notes. While relative pitch is quite common in trained musicians, absolute pitch is considered a rare skill, with some papers citing incidence as low as 1 in 10,000. (Sacks, 1995) It is, however, noted to be more common among speakers of tonal languages such as Mandarin Chinese, as well as among people who began studying music at an early age, particularly under age 5, suggesting that the skill is at least partially a function of external exposure and practice rather than being simply inborn. (Deutsch et. al, 1999; Adams, 2006) However, research has shown that this critical window appears to close by age 9, and absolute pitch is widely considered impossible to develop after adolescence. (Adams, 2006; Gardner, 2009)

However, the studies give little justification for this claim, either scientifically or empirically - no systematic studies have been found that actually aimed to teach perfect pitch to adults and then measure the results. Furthermore, there is ample evidence suggesting that perfect pitch is not an all-or-nothing skill and that the average person actually has some rudimentary form of it. The most important indicator of this is that essentially all hearing people can recognize that two pitches sound different when they are played immediately one after the other - indeed, the fact that people have any sense of relative pitch or ability to recognize a tune means that they are able to distinguish pitches. This is very much not analogous to the scenario of a color-blind person with colors, who literally cannot tell the difference between certain colors even as they are placed side-by-side. Thus, the brain does have the basic capacity to perceive the difference between notes, and developing absolute pitch is simply a matter of finding a way to encode this distinction into long-term memory. This idea is easily illustrated by a simple scenario: one can imagine sitting in a quiet room, being given an A on the piano, and being asked to wait silently and then sing back the note aloud ten seconds later. The vast majority of people are able to do this (or, if they are not comfortable singers, at least indicate their memory of the note in some manner, for example by poking around on the piano and locating the pitch, or by responding whether a second played note was the same or different as the first). Asked to repeat the exercise but wait a full minute before singing back the note, most can still do it; asked to wait ten minutes later, it might require significant mental fortitude, and the success rate among the general population might begin to dip. However, what is evident is that there is clearly some point at which one forgets a heard pitch. A person can be on either of the two extremes: if that point falls somewhere after the span of a human lifetime, then she has perfect pitch; or if that point happens immediately after hearing the note, then she could probably be considered legitimately tone-deaf. However, the rest of the population falls in between these bookends and thus on a spectrum of pitch memory permanence. Once one accepts this idea, it becomes reasonable to believe that pitch memory can become gradually retained better and better with deliberate practice, like any other set of facts one might try to commit to long-term memory.

We briefly present two more indicators of a rudimentary basis for perfect pitch. As a second
observation, a significant portion of the population can accurately sing back a heard pitch, even as they have no conscious idea what note name it is. The ability to do so suggests that the brain recognizes that pitch frequency, and immediately knows what tension to generate in the vocal cords to produce it. Finally, most people have long-term memory of at least “general” highness or lowness of pitch - asked to try to memorize a note played very high on the piano and sing it back a day later, the average person might very well fail to sing it back accurately, but he will certainly at least guess something in the correct register of the piano. In fact, a more recent study on pitch recall demonstrated that 81% of adults - far greater than the incidence of people with absolute pitch - were able to accurately sing the starting pitch of familiar recordings within two semi-tones on either side on at least one of two trials. (Levitin, 1994) This suggests people have at least a crude memory of pitch that simply needs to be fine-tuned. Collectively, these three observations show basic mental faculties present in a huge part of the population that suggest the potential for improvement.

On the topic of efficient learning strategies, we introduce some memorization algorithms thus far applied primarily to learning foreign language vocabulary. In particular, there exist two commonly used algorithms for determining the order and frequency to show flashcards in a deck of facts one would like to commit to memory. Firstly, there is the spaced-repetition system proposed by H. F. Spitzer, which now has gained traction in smart flashcard programs like SuperMemo and Anki. (Spitzer, 1939) This type of system shows cards to a user and, depending on his response of whether or not they answered correctly and how easy it was to answer, calculates the right time to show the card again, using the fact that our memory decays exponentially (with a less steep decay after subsequent correct recalls, and additional calculations included to account for how easily the card was answered). This is best suited for developing and maintaining long-term memory. Secondly, there is the Leitner system, named after its creator and first published in his highly successful book on the psychology of learning. (Leitner, 1972) It is related to the spaced-repetition system in the concept of increasing intervals for better-known cards, but has a different approach. Instead of calculating the timings individually for cards, it ranks them in relative difficulty to other cards in the deck. The model separates cards into different “boxes,” each of which corresponds to a different level of mastery of the cards within it. The cards in lower-ranked boxes are then shown with greater frequency in comparison to higher-ranked boxes - for example, the cards known the least well are shown in each run-through of the deck, while cards in the second box might be shown only once every two rounds, those in the third box once every three rounds, and so on. Roughly speaking, each correct answer for a card upgrades it to a higher box for subsequent rounds, while an incorrect answer downgrades a card immediately to the lowest box. The Leitner method is best for effectively prioritizing and covering all cards in a deck in a short amount of time.
Motivation

Our motivation for this project was to combat a relative lack of logically, effectively designed tools for learning rote skills, as well as the lack of their applied use even in the areas where they exist. In personal experience with violin, piano, and ear training instruction, the drilling of basic building blocks of skills (physical or mental), skills that could be easily encoded into a dedicated program, has never been treated systematically, nor have students or teachers turned toward software to help automate and drastically improve efficiency of practice. Even in the realm of vocabulary memorization, manual or otherwise ‘dumb’ flashcard systems are still in widespread use.

The focus on rote skills (as opposed to perhaps more creative or dynamic skills like drawing or creative writing or playing a team sport) here stems from the simple fact that drills for these kinds of skills are the most easily encoded into software and thus their mastery should be a much more trivial task than it remains in this age of fast, cheap computing power. It would of course be ideal to eventually have software to help people develop all manner of skills, but as of the present time it is the most glaringly egregious that we have not started to systematize at least these most basic skills into objectively effective practice systems, to render a previously daunting task of “just learn all of these vocabulary words” to “use this software for 10 minutes a day for five days and if you are like a statistically proven 99% of all people, you’ll have it down.”

My selection of absolute pitch as the token skill for this project was at least in part personally motivated by a desire to enhance my experience of music. The fact that people with absolute pitch can perceive differences in individual notes, and the keys of different pieces, necessarily means they are perceiving an extra dimension to the sounds they hear compared to people who do not have it. Inherently, there is some quality to music, and even sounds in general, that they appreciate that is entirely lost on the rest of the population, and so in the interest of trying to gain that appreciation, I have had a long-term interest in the idea of absolute pitch and its potential development. However, more importantly, its selection was deliberate because it appears on the surface like any other rote memorization issue: humans have the basic capacity to distinguish between these twelve notes in the short-term, but cannot remember those distinctions in the long-term, so the solution must be simply to practice. Yet it is not at all treated as such, and it is instead formally considered by researchers to be impossible to develop in adults. We were thus especially motivated to direct our efforts toward training this skill, to try to break this preconception and provide particularly compelling encouragement toward the design and consistent use of software for developing other rote skills that are already considered very much possible, and merely tedious or time-consuming.

System Design

Scope
We began with the premise that any user of this software would be able to hear a note and accurately sing it back. This certainly does not include everyone, and the project could have attained greater inclusion by encoding the task of pitch-matching inside the software, rather than relying on the user to supply that skill. However, we chose not to go in this direction, as we felt it would dilute the efforts of the project: namely, if the user was unable to even hear if they were singing the correct note or not, it would not be clear if his practice with the software would be training his pitch recognition, or merely recognition of vocal tension as he produced various notes and then was told by a program whether he were correct or not. Training the skill of matching pitch could well merit an entirely separate project, and in the interest of maintaining our focus, we did not want to address it here.

**Efficacy**

We aimed to devise ear training exercises that would avoid the usual pitfalls regarding perfect pitch. The first major challenge was to subvert the tendency toward *reliance on relative pitch*. This single factor is arguably the largest barrier to developing absolute pitch, as the societal (and, at this point, likely evolutionary) pressure has always guided the ear toward recognizing pitches in relation to each other, in the form of tunes, rather than the distinct quality of individual pitches, and now the adult ear almost cannot help but perceive the interval between any two consecutive notes. The second criterion was to avoid any premature association with *note names*. This may be counter-intuitive, given that absolute pitch is essentially defined as the ability to match pitches with note names. However, one needs to recognize that the specific note names are merely a useful convention to *demonstrate* the actual skill itself, which in its pure form is the ability to recognize different notes as different, regardless of what they are called. Introducing note names too early on in the learning practice is actually heavily counter-productive, as it diverts attention from the auditory experience toward the rather arbitrary abstraction of note names, and, very importantly, implicitly toward interval distances as it invites subconscious calculations of how many steps away one note is from the next.

We settled upon two different types of exercises, both of which avoided these two issues. The first, which appeared in the original proposal for this project, was developing the user’s *short term memory of arbitrary notes*, by having him listen to a string of random notes and then sing back the first pitch. The details of how this exercise was conceptualized was written about in detail in the original proposal for the project and thus will not be fully expounded again in this report, but it was inspired by the aforementioned idea of absolute pitch being simply a matter of slowly increasing our short-term memory of notes until at some point it crossed over into long-term memory. In regards to the two criteria discussed above, the randomness of the notes helped break the tendency to use relative pitch, as it made it unlikely that the notes would be arranged in an easy-to-follow tonal melody. Secondly, there was a deliberate avoidance of using note names: one would simply sing back the first note and
then (by listening to the string, or simply the first note, again) check if it was correct, rather than try to *name* the first note.

Our second idea was the *long term memory of specific notes*, with “specific” here referring to literal, specific notes in selected music recordings familiar to each unique user, and in particular the starting pitches of these recordings (or at least a note appearing very early in the recording, i.e. anywhere within the opening phrase). In the interest of clarity, we will refer to these recordings as “songs” for the remainder of this report, although these recordings certainly need not contain lyrics. This exercise capitalizes on a different approach mentioned above, which is fine-tuning the already existing long-term memory of how high or low pitches are. As stated, the average person can recall roughly how high or low any note is, and this recall is likely to be even more precise with particularly beloved songs she has likely listened to a large number of times, even as her guess before hearing the recording may still remain a bit off in either direction. We aimed to simply systematize practice of guessing these starting pitches, and came up with the idea to prompt the user with a song title, have her guess the starting pitch, and then listen to the song to corroborate. The program would simply repeat this as a drill with a select, relatively small library of user-specified songs, gradually increasing the accuracy and precision of her initially rough memory of these pitches.

Again in regards to the criteria of effective absolute pitch exercises, we purposely chose song starting pitches, as opposed to any arbitrary pitch in the middle of the song, as a way to avoid thinking about relative pitch. In particular, while listening to a song, and especially one already very familiar to the user, the tendency to follow along with the tune is nearly impossible to fight. However, the gap *between* different songs is a specific moment in which relative pitch actually temporarily becomes entirely irrelevant: people naturally do not pay attention to or care about the interval distance between the final note of one song and the first note of the next because they know those are two different songs and thus that interval is meaningless. With the natural tendency to follow a tune momentarily suspended, the ear can actually perceive the next song’s starting note with absolute pitch rather than with relative pitch. Our exercise was thus devised to consist only of these gaps between songs, and minimizing the time spent within one song. After having the user guess the starting pitch of a song and then check her answer, we would immediately prompt her on the next song, to disrupt the dominance of relative pitch that would occur if she continued listening to the rest of the song. Secondly, this exercise again entirely avoids association with note names: the user is asked only to think about the specific song and all her memories of having listened to it. Only if or when a point of reasonable consistency has been reached with a particular song, can the user go to the now trivial task of checking what the memorized note is actually called. She thus goes through the brunt of the memorizing process while avoiding the inadvertent mental shift to relative pitch that might occur if she were aware of and thinking about the note names from the beginning.

Via a significant amount of personal experimentation (see Experiments below) it was
determined that the second of these exercises was much more effective, and thus the final software implements this latter exercise.

Efficiency

Having come up with an exercise we believed would be effective, and which showed promising results in our preliminary practice (see Experiments below), our next design concerns dealt with how to maximize the efficiency of the exercise. This boiled down to three considerations: how to select the order and frequency of songs prompts to the user, what kind of information to collect from the user during practice, and what way to present that information (e.g. in the form of graphics, statistics, or hints to the user). We were here inspired by both of the aforementioned flashcard system algorithms, SRS and the Leitner method (see Background above). Between the two algorithms, we felt the latter was a better fit for determining the selection of song prompts, because while pitch memory is being developed, the memory of any single song is very unlikely to be truly reliable. If a user happened to get it right four or five times in a row, a SRS would consider him quite consistent with that particular prompt and therefore not show it again for several days, by which time the exact starting pitch of that song would almost certainly have been forgotten entirely. The Leitner model, on the other hand, would note that the song was consistently recalled in the short term and show it fewer times in the current session, but then make sure it was covered again in the next session, which might occur later the same day or even just a few minutes later, depending on how frequently the user wanted to use the software.

Regarding what information to collect from the user throughout the exercise, we chose to allow a range of responses, rather than simply “correct” and “incorrect.” We chose to incorporate this SRS idea because, among the correct answers, we felt it was an important distinction between a user’s reasonably confident guess and a blind one that happened to be correct, and among the incorrect answers, there was a natural thought to allow the user to record whether they had guessed too low or too high, so that the software could start to collect trend information of this type. Our final decision for what responses to take from the user thus incorporated these four situations - correct, correct but difficult/unsure, too low, and too high - and a fifth result, which was simply “no idea,” to encompass the cases where the user’s guess was either it was so off as to not be meaningful categorized into too low or high (since if it is roughly half an octave away, it could be equally considered in either direction, especially given that a user has a limited vocal range and could have already had to transpose their guess by an octave), or the user simply had no recollection whatsoever and could not form even a slightly reasoned guess. We again kept the original design concerns in mind, and thus chose not to include finer-grained user response buttons (e.g. having the user differentiate between “a little too low” and “quite a bit too low”), to avoid having the user focus too much on estimating interval distances.
In combining the correct/incorrect binary system of the Leitner model and the additional detail in user responses of SRS, we were faced with the question of how we wanted to treat the different responses - for example, perhaps a “no idea” response should be considered a greater failure than simply a “too low” or “too high” response; or a “correct” response should upgrade a card more than a “correct but difficult/unsure” response; and the upgrading and downgrading of individual songs should thus reflect that. In the end, we decided to keep the Leitner method relatively simple, with only a solid correct answer upgrading a card and all other responses considered a downgrade. The reasoning for this was largely to keep the code transparent and simple, yet still functional in terms of making sure solidly remembered (at least within the session) songs were shown less, while all others were treated as equally deserving of study.

Finally, having collected these responses from the user, we considered what form this information should be presented to the user to be the most useful or elucidating. From personal experience (see Experiments below), we found it was very desirable to have a history of how the user had done on a particular song, so we formed a visual graph of how the user had performed, available for each song, here including the full range of different user responses (as opposed to merely correct or incorrect). Secondly, we assigned each song a success rating, scaled from 0.00 ~ 1.00, that summarized a user’s entire performance history with that song through a weighted average of their results, with more recent ones results counting for more. This allowed a user to browse through their library and immediately see which songs particularly could use work, and which ones were starting to become consistently memorized.

Experiments

The system design choices of which exercise was more effective, as well as how to then design a system for efficient testing, were made through over a hundred hours of ear training practice with other existing tools. In particular, I used Teoria’s online Note Dictation software to play strings of random notes and attempt to sing back the first note, and used a simple music player on shuffle to provide song prompts. (Rodriguez Alvia, 1997-2001) As mentioned, the latter was found to give more noticeable results for long-term memory, while the former was likely helpful for developing more attention to absolute pitches in the moment (and may indeed have contributed to the overall effort, but simply did not do so in quite as direct a way). The limitations of simply using a normal music player became clear as songs could not be prioritized over others, and I had to rely on a my own likely inaccurate impressions to decide which songs to test more frequently and to gauge how well I was doing on various songs. Attached is an actual example of the records I manually maintained to keep track of this information, taken from approximately halfway through the ear training experimentation done for this project (see Spreadsheet.pdf).The qualitative comments, color categorization, and
systematization of the correctness of my responses demonstrate the clunky inconvenience of encoding my performance into quantifiable success rates by hand, and eventually prompted the efficiency concerns described above (see System Design: Efficiency above).

It is hard to scientifically quantify “how much” absolute pitch I acquired throughout this entire process, especially because my recall comes and goes and may likely be affected by various factors, such as the length of time since I last heard a song I knew the starting pitch to, or the length of time since I last heard any musical note. However, I have definitely noticed significant improvement. At the start of this project, I consistently had absolutely no memory of pitches whatsoever after the normal memory decay of at best only a couple of minutes, and I certainly had never once correctly recalled a prompted note out of the blue or even had any sort of vague inkling. In comparison, at the close of this project after four months of practice, I can now recall about four of the twelve chromatic notes with at least 90% accuracy, and that level of accuracy has been consistent over the past month (tested by deliberately prompting myself randomly throughout the day, including first thing in the morning), with the remaining eight pitches still in various states of much more questionable recall. Notably, I have observed that I am much more able to produce a prompted note, rather than recognize the pitch of a heard note. This is unsurprising given the design of the devised exercise, and again suggests that pitch recall is very much a function of the amount (and type) of practice put into it.

Performance

This project consisted mainly of designing and implementing software that would potentially be useful for development of absolute pitch. However, the time and sampling size of test subjects needed to then test the software’s empirical efficacy would be is beyond the scope of this project, both in time and conceptual focus. However, purely anecdotally, I have certainly found that having dedicated software has greatly improved the efficiency of practicing songs and evaluating progress (see Library.png, Song.png, and Exercise.png). The software has automated the tedious portions of practice such that all focus can be given to the actual task of pitch recall.

Next Steps

In later versions of this software, we aim to add more trend-seeking functionality to further aid the user. Specifically, we would like to collect even more information about the song responses for example, whether any particular pair of consecutive song prompts helps or hinders the user’s guess on the second song (e.g. if the two songs are in a specific order in a playlist the user has listened to many times, such that he actually recalls the starting note of the second using relative pitch; or if the two songs’ starting pitches are a half-step off it may be difficult to recall the second accurately). This information could then both factor into the song prompting algorithm, for example by purposely...
avoiding “crutch” pairings that give the user an unfair advantage and purposely choosing more difficult orderings, as well as into a set of hints for the user, for example informing them of these various tendencies they have in their guesses.

Monitoring the temporal element of practice more closely would also help the software further tailor the its song prompting algorithm. For example, the program might record how well a user performs in different sessions of practice, and note trends such as length of time spent in the session to suggest to the user a certain length of practice sessions, or the amount of time a user takes to answer certain songs prompts to call to his attention whether or not second-guessing is helpful in each individual case. Time could also factor into the success rating of songs as well as the visual representation of history graphs.

There is also a lot of room to add statistics, such as analyses regarding overall performance on all the songs in a session, and graphical representations of various data, for example of correct/incorrect responses over all songs over time, so that the user can gauge overall progress in their absolute pitch memory. Additionally, we could ideate some ways to actually use information to provide hints to the user during a session - for example, the program may show an optional hint section that informs the user that they disproportionately frequently will guess too high on this particular song prompt, so the user can reconsider their guess before checking their answer.

In addition to these larger pieces, there are of course always small features of convenience that could be added to improve the user experience, such as the ability to manually temporarily deactivate certain songs, or to search through the library via certain criteria, such as by rating. Following more practice with the software itself, we would add features that were determined to be legitimately useful and worth their space in the user interface.

**Conclusion**

This project was roughly 25% ideation of exercises, 70% practice of said exercises to determine their potential, and 5% coding of software. While it was at many times unnerving to have no concrete code to show for months of working on this project, in hindsight it was very much the only approach that could have produced software that is as useful as it is. There was simply no way the program could have been effective without proper thought given to system design, and there was simply no way the system could have been designed without a significant amount of actual ear training practice to provide the data necessary to determine what pitfalls to avoid in exercise design, what practical inconveniences to combat in software design, and indeed whether or not the goal was even possible to attain. This ordering constraint thus put significant time pressure on the project, as the nature of the ear training progress was to be quite slow, but it ultimately led to a rewarding and meaningful overall process.
The marked improvement in my own pitch memory throughout doing experimentation and practice for this project is a clear indication that the approach is effective at least one adult human. It would certainly be premature to generalize from my own experience, but there are promising enough signs that my improvement will continue in proportion to more practice, particularly now with the dedicated software built to streamline that process. We hope the ideas in this project will contribute to learning gains in others as well, whether in absolute pitch or in other skills.

Acknowledgments

I would first and foremost like to thank my adviser, Professor Dana Angluin, for all her help in the vague, winding road of this project. A huge amount of allowances were made throughout the semester to deal with my erratic thoughts on the project design, torpid progress with ear training, and pervasive difficulties with programming, and this project was truly only completed by the grace of her patience and kindness.

I additionally would like to extend my gratitude to the Master Judith Krauss and Dean Hugh Flick of Silliman College for being a constant source of support and understanding; to the fellow members of this year's Silliman Mellon Forum for inspiring me to finish and for sitting through my rambling presentation on this project in its earlier stages; to the creators and songwriters of the Disney-animated movie Frozen, whose soundtrack comprised the vast majority of my ear training practice and made the process infinitely more enjoyable; and finally to my suitemates, freshmen, friends, and family for their heartfelt happiness and faith in my successful completion of this final semester of college.

Works Cited

[http://www.psychologytoday.com/articles/200607/the-mysteries-perfect-pitch]

Deutsch, D., Henthorn, T., and Dolson, M. “Tone Language Speakers Possess Absolute Pitch.”

[http://perfectpitch.ucsf.edu/images/BBCMusicarticle.pdf]


Levitin, D.J. Absolute memory for musical pitch: Evidence from the production of learned melodies.
