

Milestone 3 Progress Evaluation (Nov. 25)

The Music Assistant

Daniel Griessler, dgriessler2016@my.fit.edu

Dan Levy, dlevy2016@my.fit.edu

Javier Munoz, jmunoz2014@my.fit.edu

2. Faculty Sponsor: Dr. Thomas Eskridge, teskridge@fit.edu

3. Client: Professor Elizabeth Dopira, Director of Choral and String Studies at FIT

4. Progress of current Milestone (progress matrix)

Task	Completion %	Daniel Griessler	Dan Levy	Javier Muñoz	Todo
Finish Integration of AlphaTab and ML5 Pitch Detection into Main Website	100%	50%	50%	0%	Issues with Firefox in production. This TODO is in our backlog.
Re-write Microphone Stream Code	100%	0%	100%	0%	None.
Optimize P5 Drawing	100%	90%	10%	0%	None.
Code Home Page Template (no real data)	0%	0%	0%	0%	Decided to hold off on this task until we have more data.
Code Practice Selection Page	0%	0%	0%	0%	Decided to hold off on this task until we have more data.
Design Real-Time Feedback UI	100%	33%	33%	33%	None
Improve P5 Note Drawing	45%	30%	15%	0%	Add options for parts to sing along with and adjust for scaling and save performance
Improve AlphaTab rendering and add options	21%	7%	7%	7%	Design has been discussed with client and faculty advisor.
Exercise Generation - Basics	20%	5%	0%	15%	Design has been discussed with client, actual implementation remains to be done
Post Analysis Program-Basics	90%	10%	0%	80%	Dynamic note detection needs to be done along with further testing

5. Discussion (at least a few sentences, ie a paragraph) of each accomplished task (and obstacles) for the current Milestone:

- Finish Integration of AlphaTab and ML5 Pitch Detection into Main Website
 - We had originally planned 1-2 hours for this task, but it took about 25 hours to complete. We had existing code in a test website, but the code was not optimized. Most of it needed to either be re-written or refactored. Once the code was ready to go, we began the integration process. While the integration went smoothly, we soon realized that there were significant performance problems that needed to be fixed. This required a re-write of the microphone stream code (see task below) and optimizations to the P5 drawing algorithms.
- Re-write Microphone Stream Code
 - As part of our performance optimizations, we coded the microphone stream from scratch. Originally, we were using P5's audio library, but that proved to be a road block because the library had multiple reported problems and Chrome and Safari had instituted new privacy policies that were difficult to adhere to with P5's audio API. Using the native Web Audio API provided by most modern browsers, we were able to adhere to privacy policies. Optimizations were made to the pitch detection code to only listen for audio when the music is playing.
- Code Home Page Template (no real data)
 - This task was pushed back until January because there was no good reason to complete it at this time.
- Code Practice Selection Page
 - This task was pushed back until January because we still need to encode other pieces of sheet music in AlphaTex.
- Design Real-Time Feedback UI
 - We finished designing the real time feedback. A line will persist as the singer continues through the song showing prior performance with a green/yellow/red coloring as an indicator of the student's pitch and duration ability. For dynamics, the expected dynamic will be drawn below the part which the singer is singing along with a green/yellow/red coloring as an indicator of the student's dynamics ability.
- Improve P5 Note Drawing
 - We had planned to add options for selecting which part to sing along with, reacting to scaling changes in the music, and saving performances for analysis. These options still need to be added. We added the trailing line behind the note showing the performance with color indicators for real time feedback.
- Improve AlphaTab rendering and add options
 - The integration of the working AlphaTab, ML5, and P5 into our main website took longer than expected. Luckily, this tasking only requires adding one liner API calls. We had a chance this milestone to sit down with our client and iron out how best to present these settings. AlphaTab continues to have stutter issues with the bar cursor which is now a priority to fix for next milestone.

- Exercise Generation - Basics
 - We outlined the method for generating exercises according to the client's specifications. The student will perform the song and each measure will be scored on note duration and note pitches. The worst-performing measure will be used to generate exercises.
 - The first priority will be beat exercises if the student performed poorly there. These exercises will be similar to the ta-ta-tiki-ta-ta exercises common in practice (i.e. hold some note for the appropriate duration, pitch doesn't matter)
 - The second priority is pitch exercises which will simply be performing the measure again, if necessary broken into steps (do first note, do first two notes, etc). We may also implement a slowdown of the measure to see if that helps students improve.
- Post Analysis Program-Basics
 - Given an audio stream, aubio can produce a series of pitches. By analyzing the variance of the pitches, a transition between notes can be found. However, more testing is necessary to see what variance is appropriate and if a dynamic cutoff can be achieved rather than the current limit of a standard deviation of 2 Hz.

6. Discussion (at least a few sentences, ie a paragraph) of contribution of each team member to the current Milestone:

- Daniel Griessler: I spent most of my time assisting Daniel Levy integrating AlphaTab, ML5, and P5 into the main website. I augmented the P5 drawing by adding a trailing line indicating prior performance during the piece and green/yellow/red coloring for indicating how well the student's performance is progressing in terms of pitch and duration. I also assisted Javier with the basics of exercise generation.
- Dan Levy: I spent the entire milestone integrating AlphaTab, ML5, and P5 into our main website. Daniel Griessler helped with pieces of this because he had written the original code. After the integration was complete, I re-wrote the microphone stream code from scratch because of problems with the P5 audio library. This included getting microphone input and streaming it to the ML5's pitch detection library. I also gave suggestions to Daniel Griessler on how to optimize the P5 drawing code to get better performance. He was able to implement those changes.
- Javier Muñoz: I worked on outlining the method of generating exercises, though I have not yet implemented that in code. In order to be able to generate exercises, I need to be able to compare student performance to the sheet music. In order to do that, I have been working on analyzing the student performance with the aid of Aubio. With the help of Daniel Griessler, I was able to determine that the variance in a sample of pitches could be used to determine transitions to other notes. From there, it's mainly been implementing the algorithm to analyze the pitches to determine which notes were sung for how long.

7. Plan for the next Milestone (task matrix)

Task	Daniel Griessler	Dan Levy	Javier Muñoz
Design format for comparing sheet music to student performance	25%	0%	75%
Test ml5 and aubio against a generated audio file	50%	0%	50%
Refine aubio analysis to dynamically determine the variance of note transitions	0%	0%	100%
Generate exercises for a given piece of sheet music	0%	0%	100%
Design Authentication Flows (Sign-up and Sign-in)	0%	100%	0%
Set up Authentication	0%	100%	0%
Code Practice Selection Page	0%	100%	0%
Set up SQL or Document Database	50%	50%	0%
Set up Realtime Database	0%	100%	0%
Design and Code Chat Tool (Tentative)	0%	100%	0%
Improve AlphaTab performance and introduce all customization options	75%	25%	0%
Setup Node server to respond to requests to the databases	100%	0%	0%

8. Discussion (at least a few sentences, ie a paragraph) of each planned task for the next Milestone or "Lessons Learned" if this is for Milestone 6

- Design format for comparing sheet music to student performance: In order to generate exercises, we need to be able to compare student performance to what the sheet music requires. This will likely be something simple (i.e. Start Time, Midi Value, Duration (in s or in half/quarter/eighth note). However, we will need to convert both the performance and the sheet music to this format.
- Test ml5 and aubio against a generated audio file: ML5 is used to display the current pitch to the performer, but aubio is used to determine the pitch on the back end. By testing both against a generated file, we can test how accurate they are, and for aubio, get more data for how to detect note transitions
- Refine aubio analysis to dynamically determine the variance of note transitions: The current implementation says there's a note transition when the standard deviation passes 2 Hz. Two important changes will happen. First, we will use midi-value standard deviation instead of

pitch because the logarithms in the conversion means that 2 Hz at the low end is much more distinctive than 2 Hz at the high end. Second, we will perform tests with data from other singers because a singer's vibrato will also affect the variance so testing is needed to see how to dynamically determine those transitions.

- Generate exercises for a given piece of sheet music: We currently have an outline for how to generate exercises, we just need to actually implement this in code so that once we have actual testers, we can start providing them exercises to work on.
- Design Authentication Flows (Sign-up and Sign-in): Create a UI mockup for the authentication flow. This will include a sign-up flow and sign-in flow. Choir members will have to be approved by the choir director(s) in order to be added to the choir (this is for copyright purposes).
- Set up Authentication: Code the UI design for the authentication flows and hook up Firebase Authentication to the web app. Firebase Authentication is a free tool that ensures that we have secure authentication and can track down users if music copyright violations occur. We still have to decide which technologies we are going to use to store user information beyond a username and password. The likely candidates are Firebase Storage (user's profile picture) and Firebase Firestore/MongoDB (user data such as name).
- Design and Code Practice Selection Page: Code the UI design for the practice selection page (we created the UI mockup in milestone 2). The current design includes a screenshot of the piece of sheet music, but this may prove to be too difficult to scale, so the design will likely have to be updated before coding commences.
- Set up SQL or Document Database: We need to decide which database technology is best for our needs (SQL or NoSQL). Once this decision is made, we can spin up a database and connect it to the web app. This database will store information such as pieces of sheet music, user statistics, and choir groups.
- Set up Realtime Database: In order to build our chat tool, we need a realtime database. The likely candidate for this task is the Firebase Realtime Database. This database is NoSQL and is just a large JSON tree. The advantage is that it has built-in real-time functionality and is very lightweight (if data is stored properly). This database will need to be set up and connected to our web app.
- Design and Code Chat Tool: Once the realtime database is set up and connected, we need to create the UI mockup for the chat tool and code it. This is a tentative task and is very low on our features list. If we have more pressing issues that come up, this task will be dropped.
- Improve AlphaTab performance and introduce all customization options: We need to fix AlphaTab related playback issues including fixing the beat cursor moving with the music. In addition, we need to design and implement all related settings to AlphaTab's customization options including soloing, muting, and selecting tracks.
- Setup Node server to respond to requests to the databases: We will setup a Node server to act as the middleman between the document database and the website. This includes serving up sheet music from the database to the website. This also includes saving performance data from the website and analyzing it server side before sending a set of exercises in response.

9. Date(s) of meeting(s) with Client during the current milestone:

11/21/2019 - Showed the current progress on the integrated website and real-time feedback UI.

10. Client feedback on the current milestone

- Client stressed needing to fix the blue tracer bar during playback which is unreliable and will make the website unusable.
- Client suggested separate settings window for full customization options and have the most popular options like changing volume and hiding parts on the main practice page.
- Client suggested including music glossary for all music terms/icons/definitions/symbols used in sheet music and that we display.
- For the green/yellow/red line coloring, the client would like green to be shown if within 1 half step of the correct note, yellow to be shown if within 2 half steps, and red if more than 2 half steps off from the music.
- For dynamics, client suggested a similar green/yellow/red coloring but affecting text which moves with the music indicating what dynamic is expected from the piece.

11. Date(s) of meeting(s) with Faculty Sponsor during the current milestone:

11/21/2019 - Showed the current progress of the integrated website and discussed integration issues

12. Faculty Sponsor feedback on each task for the current Milestone

1. Finish Integration of AlphaTab and ML5 Pitch Detection into Main Website
 - a. Looks good. The faculty advisor wants us to be able to scale the music. Possibly draw music closer together or draw further apart for more analytics
2. Re-write Microphone Stream Code
 - a. Seems fine other than the stutters. Fix those.
 - b. Look into adding a filter to the microphone stream for background noise
3. Optimize P5 Drawing
 - a. Looks good
4. Design Real-Time Feedback UI
 - a. Dynamics shouldn't be represented with line thickness because that is unfeasible. Faculty advisor suggested instead to annotate underneath the music.
5. Improve P5 Note Drawing
 - a. Split design for settings window. Recommendation having settings on the main page instead of a separate window.
6. Improve AlphaTab rendering and add options
 - a. Separate the pitch detection from the drawing in the function calls.
7. Exercise Generation - Basics
 - a. Process sounds good. Come up with a scoring and rank the exercises generated.
8. Post Analysis Program-Basics
 - a. Adapt current system so it accepts a WAV file input. Then, compare the results of Aubio's pitch analysis and our main website's line drawing analysis to make sure they align.

Faculty Sponsor Signature: _____ Date: _____