What is the Project title?

Harmoni

Who is in the group?

Ishaan Madan, Sarah O'Brien, Ryan McHenry

Give a high-level description of what you want to do.

Our team will be creating a tune harmonizer. Given an input (one line MIDI tune like Twinkle Twinkle Little Star), we will output two files: a harmonized version of the melody (with added chords) and a file containing only the harmony (MIDIs). This will be done through a web interface where input uploads and output downloads are easy.

Explain why this is this an interesting/useful/cool thing to do.

We are hoping to help facilitate music creation for young/non-musically oriented people. Furthermore, this could be used as an efficiency tool for musicians who regularly compose or harmonize large portions of music, and need to eliminate tedious work from their routine.

What prior art is there? Give citations.

There is no commercial product available that accomplishes what we are setting out to do with this project. There are synthesizers such as Native Instruments MASCHINE that will build triads and extended chords from a given note, but none that will automatically develop full accompaniment when given only a melody line.

Paper 1 attempts to build a tool to accomplish a goal similar to ours: given a melody as sheet music, it determines appropriate harmonizations and outputs multiple solutions in the form of chord names as annotations on the sheet music. Paper 2 attempts to use the emotional properties of a recording to automatically build a chord progression to a melody. Paper 3 utilizes a 'measure of dissonance' to guide the decision making algorithm of a project that also aims to automatically generate harmony. Although Paper 4 does not attempt to create a tool extremely similar to ours, the goals of the product described in the paper are parallel to our own goals. We drew inspiration from this paper in the form of our goals and our survey.

1. Ramirez, Rafael, and Julio Peralta. "A Constraint-Based Melo Dy Harmonizer." *Department of Mathematics*, Vrije Universiteit Amsterdam, 0AD, 1. https://www.math.vu.nl/~eliens/design/multimedia/design/media/@archive/ecai98/ramirez.ps.

2. Soysa, Amani Indunil, and Kulari Lokuge. "ChordATune - An Emotion Based Melody Harmonizer for Piano Music." *IEEE Journals & Magazine*, Wiley-IEEE Press, 16 Aug. 2010, ieeexplore.ieee.org/document/5550101.

3. Zatserkovny, Andrew. “Melody Harmonizer.” *LinkedIn SlideShare*, 28 Jan. 2017, www.slideshare.net/AndrewZatserkovny/melody-harmonizer-71497063.

4. Donahue, Chris, et al. “Piano Genie.” *ARXIV*, Cornell University, 11 Oct. 2018, https://arxiv.org/pdf/1810.05246.pdf.

What is the EXACT TASK that your system will do?

We will take a MIDI file as input uploaded onto our interactive web application. The front-end will be written in react.JS, and the server hosting will be done using EC2 (AWS). We assume that the file contains a one-line melody playing one note at a time.

Our software will first perform key detection by using the notes encoded in the MIDI and a mapping (which we will create) between sets of notes and keys. Next, we will use the MIDI to create a set of chords that are appropriate as harmonies for each note in the MIDI, referencing our mapping. Using these chord sets, we will apply our mapping to decide which chord is most appropriate to follow the previous chord (assuming we are picking a random chord in the set for the first note in the MIDI). On the web interface, two MIDI files will be available for download and for play. One will include the melody and harmony **combined** and one will provide the “Karaoke track”, which will be the MIDI of just the chords we generated. We will also keep track of all the chords we generate and print them to the interface, for testing purposes.

What measure will you will use to evaluate performance of what you build?

We will use the Acoustic Music Archive as a basis for testing our tool. Our program will output a list of chords that are included in the harmonization. We will manually generate MIDIs of nursery rhyme melodies (found here), and compare our program’s output from these input MIDIs to the chords listed on the AMA. We currently plan on using a scoring method like the following:

Our Output: C C C F F G G C

Ground Truth Chords from AMA: C F G C

Accuracy: 100%

Output: C C C F F A A C

Ground Truth Chords from AMA: C F G C

Accuracy: 75% (¾ chords in the correct order)

We are aiming for around 80% accuracy using this method. Also, it is important to note that we plan to measure the program’s performance on these new melodies we create by asking a group of listeners to rate the combined melody-harmony audio on a scale of 1-3, 1 meaning "This sounds right," and 3 meaning "This sounds wrong."

Is there a data set that your system can be tried on? If so, what is it (give links). If not, explain why not? .

We have not found a dataset that we can use for sample inputs (simple one-line melody, one-note-at-a-time MIDIs), but we plan to generate our own MIDI files using MIDI controllers. We will create both melodies to be used for ground-truth comparisons and new melodies we can use to test the actual purpose of the program.

What is the baseline approach you will compare your system to?

Our baseline approach for comparison will be a music theory student with the basic ability to harmonize a given melody. Using a standard library of folk tunes available at www.acousticmusicarchive.com, we will score the accuracy of the student the same way that we score the accuracy of our algorithm, and judge the success of the algorithm based on the difference of average accuracy scores. We will get an independent third party (student) to objectively harmonize the melodies provided and see how our system compares.

Describe any existing software packages you will use.

We will be using the Java MIDI package for our backend processing and usage of MIDI files. The web app design will be done with react.js, and hosting will be done on AWS. We will be using existing packages to read and write MIDI files, and later convert audio files to MIDI, and vice versa. Digital audio workstations may be used for testing as well (such as Ableton). We have begun exploring the Java MIDI package.

Describe any software will you need to write.

For the back-end, we will write the algorithm to compose a basic accompaniment for any given MIDI melody and output it. For the front-end, we will write JavaScript for a web application that will ask for file uploads and provide output interactive audio and download files.

What are potential obstacles to success?

We will need to learn how to construct MIDI files using Java frameworks, and how to parse an existing MIDI file. We will need to design a storage solution for managing input MIDI files and output MIDI files.

How will you QUICKLY determine if these obstacles will stop you?

We are beginning our project with a decidedly iterative approach. Our first step will be testing if we can load a single note (with MIDI) and returning the appropriate chord for it. If we struggle with the first step, we know we will need to pivot in some way. By then continuing on to larger MIDI files with more notes, we can increase the level of difficulty without the risk of a major roadblock upending the project as a whole.

What other tasks (besides coding) will you need to do?

We will need to generate MIDI melodies for songs, including new material and test data. We will play the 17 nursery rhyme tunes on AMA (to begin with) on a MIDI controller and export them for our test data collection. We will also need to conduct a user survey to test the validity of new melody and harmony generation.

Milestones

Date	Milestones	Responsible Party
2/27 (Meeting 1)	1. Clean front-end interface where file upload can happen 2. EC2 instance set up 3. Review and understand JAVA MIDI package 4. Record ground-truth tunes using MIDI	1- 2 - Ryan 3 - Sarah, Ishaan 4 - Ryan, Ishaan
3/6 (Meeting 2)	1. Working code that takes note and key as input and gives set of possible chords as output 2. Coded Java chord-to-chord mapping rules in a code-accessible structure 3. Very simple one-note end-to-end demo	1, 2 - Sarah, Ishaan 3 - Ryan
3/13 (Meeting 3)	1. Working end-to-end code that we have	1- Sarah, Ishaan 2 - Ryan

	tested using ground-truth lullabies, other ground-truth examples 2. New melody input MIDIs with demos	
3/22 (Final Project)	1. Poster made 2. Survey data acquired for new melodies we created 3. Multiple demos, hopefully with examples of really good outputs (and some not-so-good outputs with reasonings for why they didn't go as well) 4. Presentation written	1, 2 - Sarah, Ryan 3, 4 - All