Practica: A Music Education Application for Learning Jazz Improvisation

by

Julia Fiksinski

B.S. Computer Science and Engineering
Massachusetts Institute of Technology (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

Author . . . .
Department of Electrical Engineering and Computer Science
August 13, 2021

Certified by . . . .
Eran Egozy
Professor of the Practice, Music Technology
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Practica: A Music Education Application for Learning Jazz Improvisation

by

Julia Fiksinski

Submitted to the Department of Electrical Engineering and Computer Science
on August 13, 2021, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

We present Practica, a music education application which combines the user experience of a play-along practice application, the functionality of a melody transcription program, and the soloing tips and harmonic analysis that a music student might receive from an instructor. The user can read soloing tips and introductions to harmonic analysis, record themselves soloing over a backing track, and view transcriptions of their solos with different analysis modes applied in the form of color-coded notes. We develop an audio processing and transcription pipeline to generate sheet music for solo recordings. We examine how to present the subjective teaching and evaluation of improvisation in a programmatic manner. Two user studies suggest that Practica successfully presents students with an educational platform that empowers them to improve their musical abilities and explore jazz improvisation in a interactive and beginner-friendly format.

Thesis Supervisor: Eran Egozy
Title: Professor of the Practice, Music Technology

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Eran Egozy. Thank you very much for agreeing to advise my thesis. Your guidance throughout the research process has been invaluable. I greatly appreciate all of your advice and encouragement.

Thank you so much to my family for their unconditional support. It is no small comfort to know that you are always in my corner, pushing me to be my best. I would also like to thank my friends for their encouragement and company. I feel very lucky to know such fantastic people.

Finally, thank you to the MIT Festival Jazz Ensemble for five incredible years of music together. In particular, thank you to Dr. Fred Harris for your guidance in music and beyond. I credit much of the inspiration for this thesis to your impact as a jazz educator.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Improvisation is a key component of jazz. The spontaneous composition of a solo over a set of chord changes requires not only an understanding of harmony and rhythm, but also an ability to intentionally convey emotion with these structures. As with learning to read music or playing an instrument, learning to improvise takes dedicated work. Numerous systems for teaching music currently exist but only focus on a single aspect of music education, such as sight-singing or ear training. Evaluating the quality of a solo is inherently subjective and significantly more difficult to implement in a system. Helping users analyze harmony and rhythm in compositions to build stronger understanding is a more feasible goal that some systems have explored.

Jazz educators often recommend transcribing recorded solos as a method of practicing improvisation because it covers a wide range of learning opportunities: it offers practice of harmonic and rhythmic analysis and ear training; it increases the musical vocabulary that one can utilize while soloing; and it deepens one's understanding of how skilled musicians craft solos. However, transcription can be time-consuming, especially for beginners. Without an understanding of how the phrases played relate to the underlying chord progression, a student may not learn as much as possible from transcribing a solo. Building that level of understanding requires exposure to harmonic analysis. Though plenty of resources exist in print and online to address these challenges, using those resources is incomparable to learning and receiving performance feedback directly from an instructor, which may not be an available opportunity for some students.

While students can practice improvisation through performing or playing in jam sessions with their peers, learning how to improvise as a beginner can be intimidating when playing with more experience improvisers. Currently available systems such as Impro-Visor or Mapping Tonal Harmony Pro already support transcription or harmonic analysis and may be used for educational purposes, but existing applications generally do not integrate both of these activities into a single system geared towards education [1, 2].

Practica is a web application which assists music students with learning how to improvise. Though more advanced students may be comfortable with transcription or harmonic analysis, this system does not require users to have any familiarity with these activities. Practica simply requires students to record themselves improvising over a backing track. Each backing track has an accompanying help section that provides students with soloing tips and information about the chord progressions. Students can explore transcriptions of their recordings and view introductory harmonic analysis of their solos in the form of color-coded notes. Practica offers multiple types of color-coding analysis to teach students about chord tones, tensions, and scale modes. Students can also edit their transcriptions and save their solo recordings and sheet music for future reference.

We evaluate Practica through two rounds of user testing. We seek to investigate (1) whether a software system can assist musicians in transcribing and analyzing solos, and (2) how much of the experience of learning improvisation from an instructor or other musicians can be replicated with a computer program. Through the development and evaluation of Practica, we seek to empower students to explore jazz in a hands-on format and improve their musical abilities using an accessible and beginner-friendly platform.

Chapter 2

Background and Related Work

2.1 Jazz Improvisation

Throughout this research, we describe improvisation as the spontaneous melodic composition of a solo over a set of chords ("chord changes"). However, one could instead improvise over a single chord or without a particular chord as a tonal anchor (i.e. atonal). Jazz compositions can include some guidance for improvisation through musical features such as time signature, song form, and commonly used chord progressions.

To practice improvisation over a particular composition, a musician first learns the chord changes of the piece. Subsequently, the musician may practice referencing musical lines from the piece (e.g. playing a variation on a set of notes from the original melody) or playing the chord tones (the main notes in a chord), tensions (notes that cause dissonance–a clash in harmony), and extensions (notes that extend the chord higher in pitch past the usual four notes in the chord; i.e. the "upper structure"). Many jazz musicians recommend transcription as another activity for practicing improvisation. Transcribing the solos of other musicians not only provides examples of improvisation over certain musical compositions, but also provides students with licks, or short melodic lines, that they can incorporate into their own solos.

2.2 Preliminary Interview

Preliminary research for this thesis included an interview with Dr. Frederick Harris, Jr., the Director of Wind and Jazz Ensembles at MIT, regarding jazz education and teaching improvisation. Dr. Harris made the following suggestions for how to practice improvisation:

- Learning a piece by sections. Jazz compositions often have multiple labeled sections that repeat. When learning to solo over a new piece, focusing on one section at a time instead of immediately trying to solo over the entire piece can help students learn each section's chord changes better.

  For example, a student may focus on learning how to solo over "So What" by Miles Davis using a modal approach. The student may first practice soloing using the D Dorian scale over the first sixteen bars of the piece, then practice soloing using the E♭ Dorian scale over the next eight bars of the piece. Finally, the student can practice soloing over the entire piece using these two scales, as the last eight bars of the piece return to D Dorian. The benefit of using a modal jazz standard like "So What" for teaching improvisation is that it reduces the number of scales in the chord progression for students to learn (e.g. only one scale mode per section) and lets students start practicing improvising over the entire solo form earlier.

- Focusing exclusively on rhythm. A musician's rhythmic choices can impact their solo as much as their melodic choices. Playing interesting rhythms but only a few different pitches can still produce a pleasing or engaging solo, whereas playing a more creative or complex melody without any rhythmic intention or variation might not sound as compelling.

  To improve their rhythmic comprehension, a student could first start by practicing a particular rhythm or learning how to play with syncopation without thinking about chords or melodies. The student could then practice playing that rhythm over the chord progressions in a piece.

- Discussing notes from the perspective of tension and resolution instead of labeling notes as “right” or “wrong.” Incorporating tension makes a melody more complex. A musician can create tension by playing a dissonant note outside the chords (a “wrong” note) and then resolve that tension by playing a note that complements the surrounding chords. Simply calling a note “wrong” fails to teach this concept and may discourage students from experimenting with tension. Instead of coloring dissonant notes in red and describing them as “wrong,” color-coding notes by chord tones or tensions could help introduce this concept to students.

While we could not implement all of these suggestions in the current iteration of Practica, we focused on the concepts of (1) learning in steps and (2) teaching dissonance as tension instead of as “wrong” notes. The educational components of Practica present soloing tips in increasing difficulty and describe notes in terms of chord tones and tensions.

2.3 Related Work

Much of the current research at the intersection of jazz and computer science focuses on computer-generated jazz improvisation using machine learning. Numerous projects aim to generate melodies through the use of melodic similarity patterns or probabilistic grammars to capture differences in musical style across musicians. A common approach, as seen in projects like BebopNet or Deepjazz, is to create an LSTM (long short-term memory network) trained on solo transcriptions or MIDI files to generate new improvisations [1, 2].

A related research challenge is pattern similarity search for input melodies. A leading resource in this field is the Weimar Jazz Database, a database of solo transcriptions that comprises part of the larger Jazzomat Research Project [3]. Jazzomat applies music analysis and cognitive psychology to analyze the creative and educational aspects of improvisation. The project has produced music analysis tools, search engines for pattern similarities, and applications for exploring feature and pattern history of solos [3].

In the consumer space, many applications for music theory and analysis educate users in ear training. These applications range in difficulty from basic intervals and triads to inversions and chord extensions. Some applications focus on creating more structured learning experiences, such as EarMaster, which offers courses for users to explore sight-singing or jazz chords [4]. Others, such as the mobile application Harmony Cloud, generate custom progressions from user-selected input chords and inversions so users can play along [5]. Despite their differences in approach, many of these applications share a similar design, which includes a learning mode that lets users listen to examples and a challenge mode that provides users with quizzes to evaluate their improvement over time.

As for melody transcription applications, many exist either for educational (e.g. challenging users to transcribe phrases) or productive (transcribing users' recordings for future use) purposes, but few market themselves for both. Most of these applications offer automatic monophonic transcription, which as of now is sufficiently established and utilized in numerous consumer- and research-geared products [6, 7]. Accurate automated transcription of polyphonic music presents a greater challenge. Detecting simultaneous pitches and octaves as distinct notes is more difficult and often results in reporting extraneous notes [8, 9].

Impro-Visor is a musical notation program in which users compose or transcribe solos in a score editor and view suggestions for licks or harmonic notes. Users may also listen to solos generated by Impro-Visor (based on a grammar learned from transcriptions) or play along to a generated accompaniment [10]. Impro-Visor covers a wide variety of use cases and promotes learning of improvisation by encouraging users to compose new solos instead of transcribing existing recordings.

iReal Pro is a play-along application that offers users hundreds of jazz standards for practice. The application allows users to input their own progressions and customize the instruments used in the backing tracks. Musicians may also use iReal Pro to practice more rigorously by challenging themselves with automatic transpositions or tempo increases with each play-through of a piece [11].

Mapping Tonal Harmony is another play-along application that allows users to select backing tracks. While iReal Pro focuses more on letting musicians customize their practice sessions, Mapping Tonal Harmony places more emphasis on chord progressions and lets users explore an expansive tonal harmony map for each piece [12, 11]. As the user plays through a backing track, the program updates the current state in the harmony map to show how the chord progression evolves throughout the piece. The program allows users to input chord progressions and examine chord tones, tensions, and notes to avoid.

Chapter 3

Design

With Practica, we present a novel music education tool by combining the user experience of a play-along practice application, the functionality of a melody transcription program, and the soloing tips and harmonic analysis that a music student might receive from an instructor. As is standard for play-along applications, Practica displays the sheet music for a selected piece and plays the backing track. The application also offers a help section for each piece with basic information about its chord progressions and relevant soloing tips. A user can practice improvising over the backing track and record a solo. Like other transcription applications, Practica processes the recorded solo audio to generate a transcription and then displays the returned sheet music. In addition to transcribing the user's solo, Practica also inserts annotations into the transcription by using color-coded notes to convey relevant harmonic information, such as chord tones or tensions.

3.1 Example Use Case

Figure 3-1 depicts an example scenario in which a student practices improvisation with Practica. A student opens the program and selects one of the available compositions from the system database. The student wears headphones to hear the backing track so that when they record a take, the recording contains only the solo audio. The student can listen to the backing track using the audio playback component and follow along

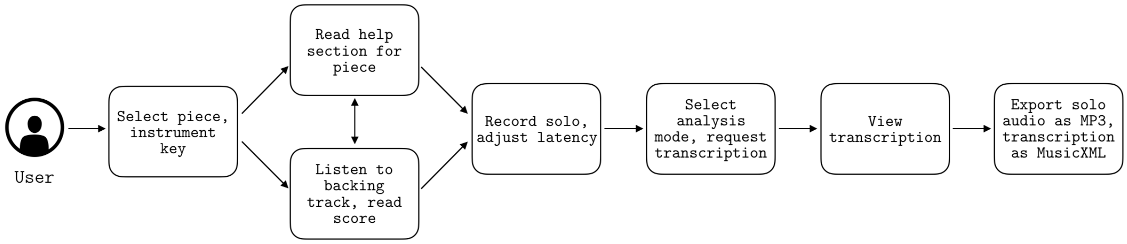


Figure 3-1: Example use case of a practice routine using Practica.

in the sheet music displayed in the score component. If interested, the student may also open the help section for the currently selected piece to read relevant chord information and soloing tips. The student records themselves playing a monophonic solo over the track after a specified count-off. Once the student finishes playing, they can listen to their solo along with the backing track and request a transcription of their solo or start another recording.

After requesting and receiving the solo transcription from the server, the application loads the sheet music for the transcription and displays it in the score component. The student can scroll through the transcription score to view color-coded harmonic analysis of how their melody relates to the underlying chord progression. The student may also use the audio playback component to listen to their solo while the score component automatically updates a matching cursor in the transcription score. The score component includes an editor, with which the student may modify any pitches and rhythms in the transcription. Before closing the system, the student saves and exports both the audio and the sheet music for their transcribed solo.

3.2 Architecture

Practica's architecture uses the client-server model with a React frontend and a Python Flask backend. The server controls only the generation and analysis of transcriptions of user-submitted audio; the client controls other interactions (such as recording and downloading audio). The architecture also integrates Flat, a third-party music notation web software [13]. Practica uses Flat to display the scores for backing tracks and solo transcriptions. The application uses Flat instead of a custom-made component because creating a new component to support rendering and editing music notation in the browser would require a significant amount of work, and Flat already provides the rendering and editing features that Practica requires.

3.3 Client Interface

The client interface consists of a single page with five components: (1) help section, (2) backing track and instrument key selection, (3) audio recording and playback controls, (4) transcription and analysis controls, and (5) score editor.

(1) Help section

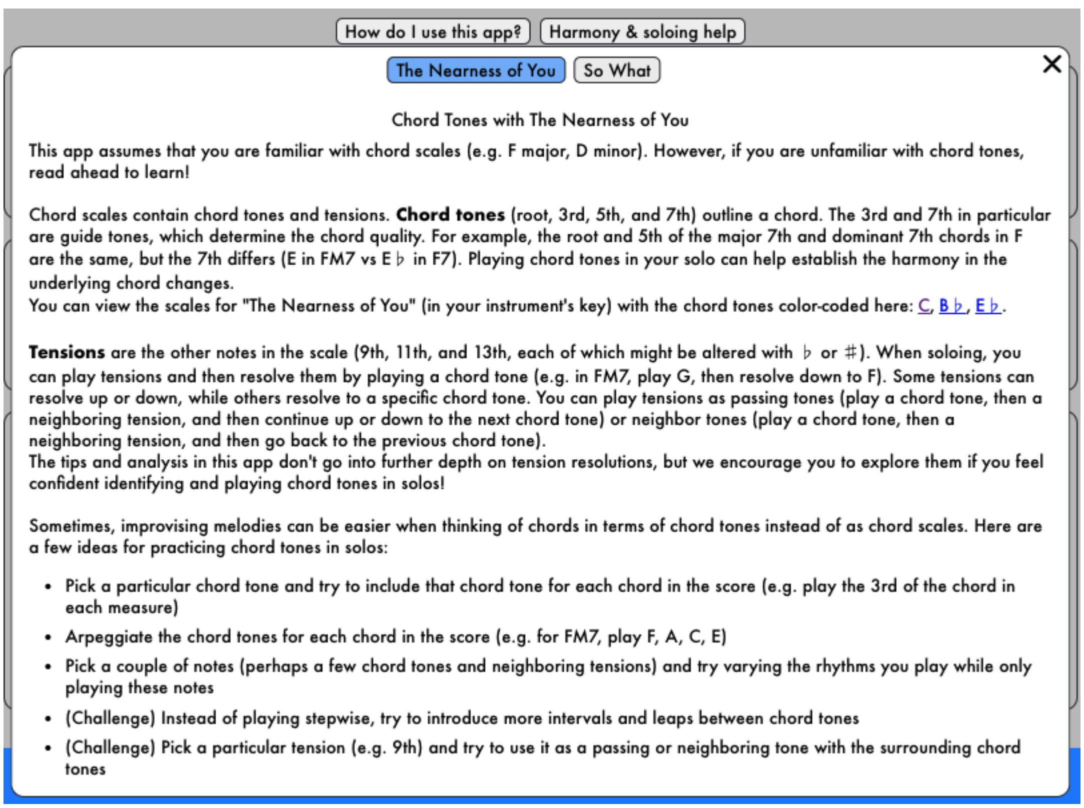


Figure 3-2: Help section, currently displaying the harmony and soloing help for "The Nearness Of You."

The help section consists of an application walk-through for first-time users ("How to use Practica") and a set of soloing tips for each backing track ("Harmony & soloing help”). The soloing help for each backing track includes the scales for the relevant chords and some musical ideas that users can incorporate into their solos. Figure 3-2 shows the harmony and soloing help section for “The Nearness Of You.” A more detailed description of the design of the help section can be found in Section 3.5, Educational Approach.

(2) Backing track and instrument key selection

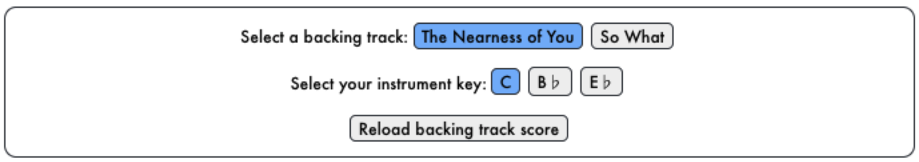


Figure 3-3: Current backing track selection is “The Nearness Of You” in concert (C) pitch.

This component allows the user to select one of two jazz standards as backing tracks: "The Nearness of You" and "So What" [11] [14] [15] [16]. This application could support a larger library of backing tracks, but we chose to support only two backing tracks due to time constraints. Implementing fewer backing tracks also allowed us to develop more detailed help sections for each piece and collect user feedback directly comparing the two. These two pieces were selected because both are well-known jazz standards. “The Nearness Of You” is a slow tempo ballad with two distinct sections. Offering a slower piece gives students more time between each chord change to practice playing the current chord tones and tensions. “So What” is a good entry point into modal jazz because it only uses D Dorian and E♭ Dorian. Requiring students to learn fewer scales in a new mode lets them start practicing improvisation sooner.

The user can then select the key of their instrument (C, B♭, or E♭) to view the score for their selected standard in either concert or transposed pitch. Supporting transposition for different instruments is necessary because not all instruments are in the key of C.[1] Figure 3-3 shows an example selection in which “The Nearness Of You”

[1]Some instruments (such as saxophones or clarinets) with different pitch ranges play in different keys so that each notation, when transposed correctly for each instrument, has the same fingering.

is selected for a C instrument. If the current score displayed in the score editor is a transcription of one of the user's solos, the user may also choose to reload the lead sheet for their selected backing track using this component.

(3) Audio recording and playback controls

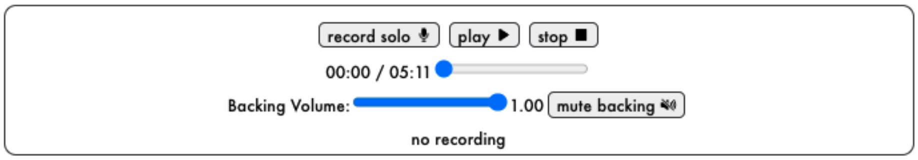


Figure 3-4: Audio recording and playback controls (without a recording made).

This component contains the controls for recording audio and listening to the recordings and backing tracks (Figure 3-4). The user can navigate to a particular playback time using the seek control, change the playback volume, and reset playback to the audio start time. The user can also start and stop recording using this component. Upon first loading, Practica requests user microphone permissions, which are necessary for recording.

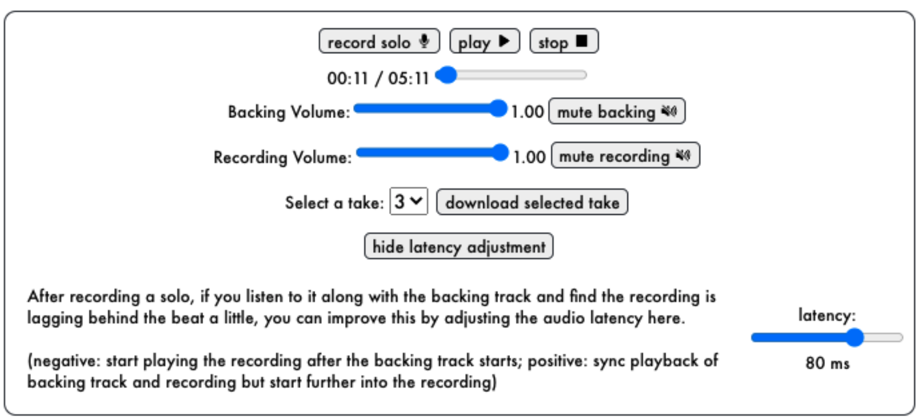


Figure 3-5: Audio recording and playback controls (with recordings made).

Additional playback controls appear in the interface once the user has completed their first recording (Figure 3-5). Practica links the recording and backing track playback so that the user can listen to both simultaneously. The user may modify or mute either track's volume to focus on one particular audio source. The user might notice that their recording audio lags behind the backing track. This problem is a result of latencies inherent in the software and hardware of the user's computer system. There is often a time delay from when audio processing begins to when the audio data is available for recording or playback. This time delay can take tens to hundreds of milliseconds [17] [18]. The resulting audio recordings have additional silence at the start, so the latency adjustment causes playback to start further into the recording to compensate. Adjusting the latency using the slider shown in Figure 3-5 can compensate for delays of up to 200 milliseconds. After recording a solo, the user should adjust the recording playback latency using the slider to ensure that the backing track is perfectly lined up with the solo they just recorded. The user may repeat this process of adjusting the slider and listening to the synced recordings until they reach a satisfactory latency adjustment.

Finally, if the user records more than one take, they can select a take in this component and download it. The client manages audio downloading and saves solo recordings as MP3 files. Currently, the user can only download an MP3 file of their solo recording and not a mix of their solo with the backing track. To create a mix, users would need to request the MP3 files for the backing tracks or record the backing track playback separately and then join the backing audio with with their solo files. Supporting the downloading of an audio mix of the solo and backing tracks is an important update for a future version of Practica.

(4) Transcription and analysis controls

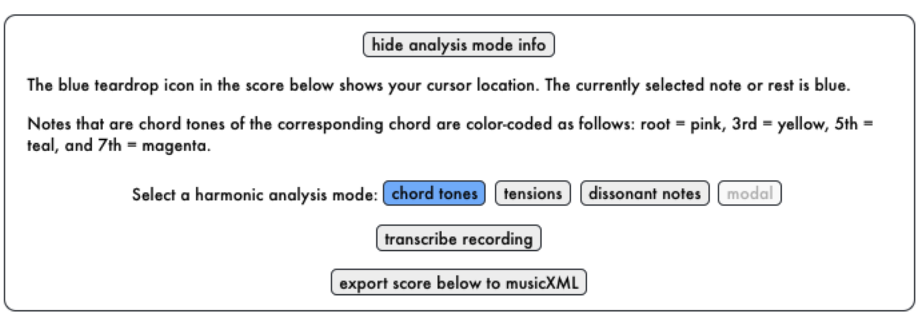


Figure 3-6: Transcription and audio controls

This component contains the controls for requesting solo transcriptions, exporting transcription scores, and selecting a type of harmonic analysis to apply to solo transcriptions (Figure 3-6). This component also contains a legend explaining the meaning of each note coloring in the selected analysis mode. Once the user has recorded a solo and adjusted the audio latency, they can request a solo transcription. Clicking on a different analysis mode updates the client state, but the user must click on the "transcribe recording" button to trigger a new server request and apply the new analysis coloring to their solo transcription.

(5) Score editor

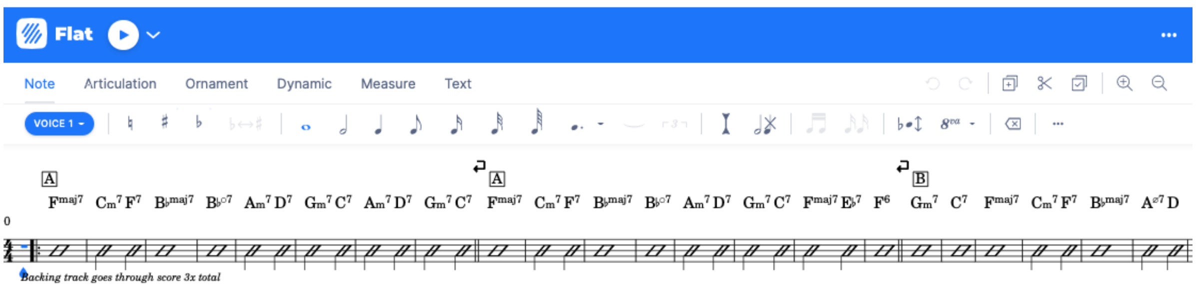


Figure 3-7: Score editor displaying sheet music for "The Nearness Of You" in concert (C) pitch.

This component primarily uses a third-party music engraving embed from Flat, a music notation web software package, to support the rendering and editing of sheet music [13]. All responses to Flat API calls are first handled by this component. Loading a score into the Flat embed requires three pieces of information: (1) the score ID (if the score is hosted by Flat) or MusicXML file, (2) the duration of the accompanying audio file, and 3) a list of synchronization points. The duration and synchronization points define the pace at which the cursor moves throughout the score. Each synchronization point is represented using a measure number and the time (in seconds) at which the cursor should reach that measure.

Figure 3-7 shows the Flat score embed used in Practica. The score editor and audio playback controls are linked so that pressing the play, pause, or stop buttons on either component triggers the same behavior for the other. These controls were not originally linked in the first version of Practica, as we thought that users would primarily start or pause playback from the audio recording and playback controls. After evaluating user feedback, we linked these controls in the second version of Practica to better integrate these components. The user can export whatever score is presently visible (backing track or transcription) in the editor as a MusicXML file. The user may also use built-in editing functions provided by the score embed to change the notes, rests, articulations, ornaments, dynamics, measures, or text of the score displayed.

3.4 Server Design

The application architecture includes a server solely to support audio processing and transcription in Python. More audio processing packages are available in Python, and implementing the processing and transcription generation code in JavaScript would be more challenging. The server also responds to pings from the client regarding transcription status or retrieving MusicXML files. The server stores files necessary for transcribing solos for either backing track: mappings of times to beats, mappings of beats to chords, and chord scale information. Section 4.2 describes the creation and usage of these mappings in greater detail.

3.4.1 Audio Processing Pipeline

Figure 3-8 outlines each audio processing step. Upon receiving a transcription request, the server processes the solo audio using the solo MP3 file, the name of the backing track used, the specified instrument key, and the adjusted latency value. The audio processing script returns a list of the estimated notes played in the solo. Each note is defined by its actual onset time (the time when the note starts), quantized onset time (the beat nearest to the actual onset, in case the note is slightly off-beat), pitch, and duration. The audio processing pipeline is as follows:

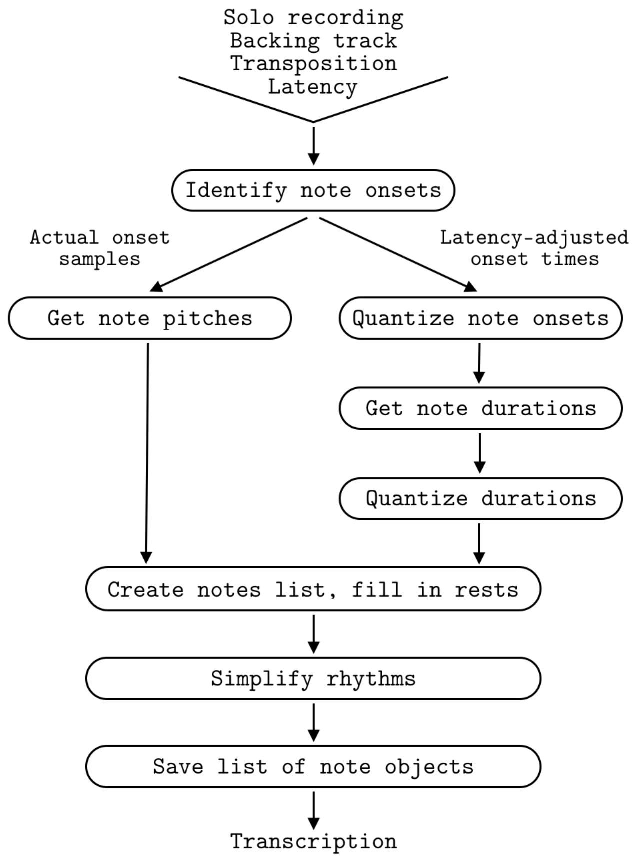


Figure 3-8: Flowchart depicting audio process pipeline.

1. Identify the note onsets in the solo recording. Save these onsets in two forms: (1) a list of samples at which the onsets occur, and (2) a list of times (in seconds) with the input latency value subtracted. The original list of onset samples is necessary for correctly indexing into the audio object for pitch detection. We adjust the onset times using the specified latency value to quantize notes more accurately.

2. Each backing track has a corresponding mapping of each sixteenth note in the audio to the time when it occurs (a beat map). For each latency-adjusted onset time, find the closest time in the beat map and return the associated beat as the quantized note onset.

3. Each set of audio samples between onsets is a potential note. For each note segment, estimate the pitch and octave of the note played. If the input instrument

key is not C, transpose the pitch as necessary.

4. For each note segment, estimate the end time of the note to calculate the note's duration.

5. For each note, quantize its duration to an integer number of sixteenth notes. This quantized value should be the greatest duration that does not exceed the original estimate or surpass the start of the next note (so as to avoid overlapping and shifting other notes later).

6. Map the time of each sixteenth note in the backing track to either a note or a rest. This step produces a consolidated list of all note objects in the solo.

7. Because the transcription supports sixteenth note durations as the smallest denomination and errs on the shorter side when estimating note durations, some of the generated rhythms are overly precise in their number of sixteenth notes compared to what a person would estimate in a handwritten transcription. To amend this, iterate through the list of notes and rests and simplify the rhythms of neighboring objects.

8. Save this simplified version of the notes list for use in transcription.

3.4.2 Transcription Process

After processing the solo audio to create a list of the notes played, the server then generates a MusicXML file for the transcription using the list of notes, the requested harmonic analysis mode, the instrument key, and the name of the backing track used. Each backing track has a corresponding mapping of each beat in the audio to the name, quality, and scale of the chord at that beat. This chord map is used to add chord symbols to each measure to match the backing track. The chord map is also used to compare each note's pitch to the specified pitches in the chord scale for harmonic color-coding. After entering each note and rest into the MusicXML file, the transcription script fills the end of the last measure with rests if it is incomplete and then saves the completed transcription. The server then sends the transcription file to the client for display in the score component.

3.5 Educational Approach

Practica presents educational content through the help sections users peruse upon opening the app and the harmonic analysis visible in solo transcriptions. Each backing track has a corresponding harmony and soloing help section that provides relevant chord scales and soloing tips. Depending on the current backing track, users have a selection of different harmonic analysis modes to choose from and may apply one to their solo transcription.

3.5.1 User Interview

The development stage for the educational content and harmonic analysis included an interview with one of the users (Tony Terrasa) from the first round of testing. Tony offered extensive feedback during testing and expressed interest in discussing possibilities for the educational content of the app. Based on his prior experience in learning improvisation, Tony gave the following suggestions:

- The definitions of "right" and "wrong" notes in jazz improvisation can widely vary among musicians. When analyzing the notes a user played, adding context as to why notes may be considered dissonant is more helpful than marking notes as "in" or "out" of chords.

- Providing a cheat sheet in the app with scales for the relevant chords in each backing track could encourage users who have little to no experience with improvisation.

- Presenting notes in chords as chord tones (root, 3rd, 5th, 7th) and tensions (9th, 11th, 13th) could help build users' understanding of the chords in the backing tracks. Giving users this information could also serve as a springboard for soloing. Thinking of the key notes for a chord (i.e. the chord tones), rather than remembering all of the notes in that chord's scale, can simplify a musician's approach to soloing.

We implemented multiple ideas from this interview in version 2 of Practica. Each help section now includes the scales for the backing track chords. We also expanded the analysis modes offered to highlight chord tones and tensions. We did not update v2 to include an explanation of dissonant notes and how to use them in a solo. This information would be a valuable addition to a future version of Practica.

3.5.2 Soloing Tips

The two available soloing help sections in Practica are "Chord Tones with The Nearness Of You" and "Modal Jazz with So What." "The Nearness Of You" is a tonal jazz piece with clear tensions and resolutions in the chord progression, so teaching improvisation over this piece is more effective using chord tones and tensions. The corresponding help section explains what these tones are, provides color-coded scale sheets for each chord in the score, and suggests different approaches to incorporating more chord tones or tensions into a solo. Because "So What" is a modal jazz piece set in the Dorian mode, tips related to chord tones and tensions are less useful. Instead, the help section for this piece gives a brief explanation of modal jazz, provides the Dorian modes for the necessary keys, and encourages users to try and hear a difference between soloing using the chord tones for Dm11 and E♭m11 (in concert pitch) and using the D and E♭ Dorian modes. Appendix B includes both help sections for reference.

3.5.3 Analysis Modes

Practica offers four modes of harmonic analysis: chord tones, tensions, dissonant notes, and modal. The system uses the selected mode as a rule for coloring specific notes when generating solo transcriptions. The chord tones method highlights the root, 3rd, 5th, and 7th of each chord, while the tensions mode highlights the 9th, 11th, and 13th. Appendix A lists the color assignments for each analysis mode.

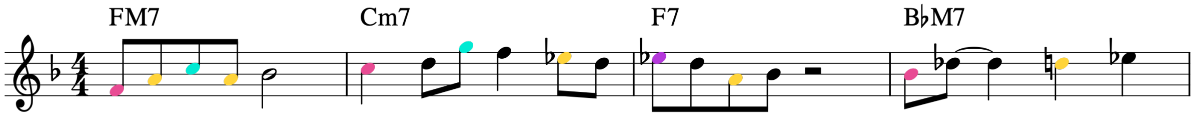


Figure 3-9: Analysis mode highlighting chord tones.

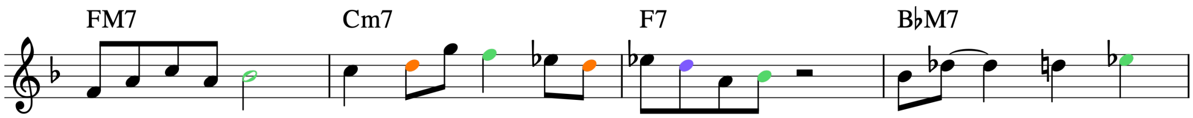


Figure 3-10: Analysis mode highlighting tensions.

Each tone and tension has a distinct color assignment, which allows users to more clearly see how chord tones change throughout a chord progression. Ideally, this coloring scheme also reinforces to users how the notes they play while soloing fit into the underlying chords. Figures 3-9 and 3-10 show examples of a solo over "The Nearness Of You" with these color-coding modes applied.

The dissonant notes mode highlights notes in a measure if they are not part of the relevant chord scale. See Figure 3-11 for an example of a solo over "So What" using this analysis mode.

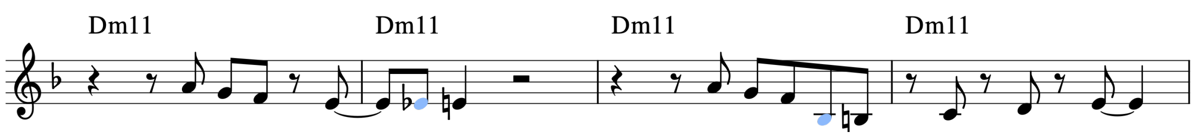


Figure 3-11: Analysis mode highlighting dissonant notes.

The modal analysis highlights notes in a measure that are part of the Dorian mode for the relevant chord. Figure 3-12 shows an example of a solo over "So What" using this highlighting.

Certain analysis modes are backing track-specific in alignment with the corresponding soloing tips in the help section. The tensions and dissonant notes modes

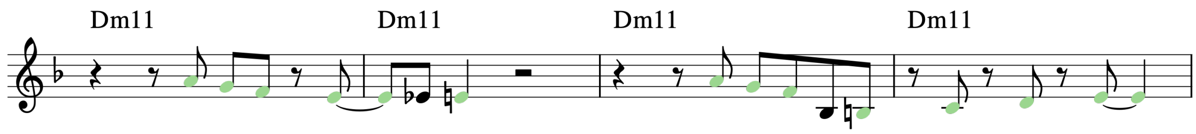


Figure 3-12: Analysis mode highlighting notes in the D Dorian mode.

are only available for "The Nearness Of You," while modal analysis is only available for "So What." The chord tones mode is available for both backing tracks. The chord tones mode is available for "So What" because, although the help section clarifies that "So What" is a modal jazz piece, one of the soloing tips encourages users to try soloing over the piece as if they were following the chord tone approach from "The Nearness Of You" and then compare that solo to one recorded using the Dorian mode. Appendix B lists all soloing tips included in the help sections.

Chapter 4

Implementation

Practica is a single-page web application built with React.js and a Flask Python server. The source code used for the version of Practica evaluated in this thesis is available at: `https://github.com/fiksin/practica`

4.1 Frontend with React

The components that contain the audio recording and playback controls, the transcription and analysis controls, and the score embed are each implemented using class components in React (described in Section 3.3). The help section and the backing track and instrument selection section are both implemented in the parent App controller for the application.

The implementation contains separate class components for backing track playback and recording playback, but the user interface displays a single set of audio controls (with the exception of track volume) that links playback for the two components. The client uses `react-howler`, a React wrapper for the `howler.js` audio library, to create HTML5 audio objects in each audio playback component [19]. Each combination of backing track and instrument key is associated with the ID of the related score hosted on Flat. Clicking on a different backing track or instrument key triggers a Flat API call to load the requested track using its Flat score ID (Figure 4-1).

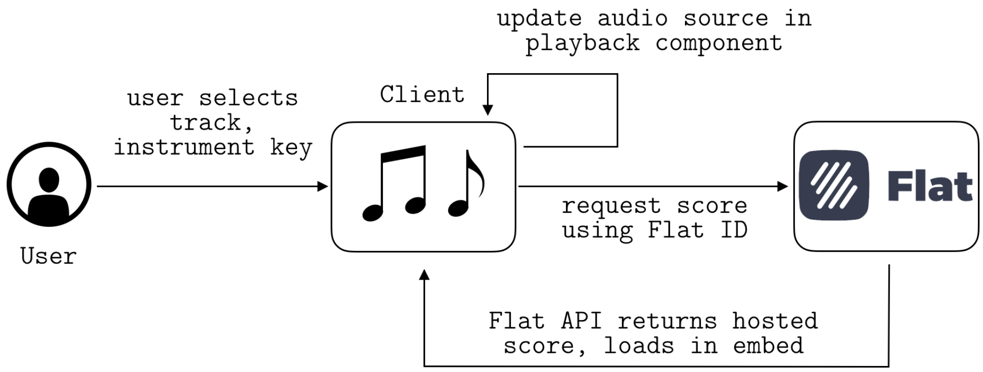


Figure 4-1: Process of updating backing track score in client.

The client does not instantiate a recording playback component until the user has recorded their first solo. Each time a user stops recording a solo, the system adds the new solo to the list of audio sources for the recording ReactHowler object. The recording implementation uses the MediaStream Recording API to record audio [20]. The downloading implementation uses the `lamejs` MP3 encoder with a sample rate of 48 kHz [21].

The user may adjust the audio latency in the playback controls. As mentioned in Section 3.3, the client supports latency adjustments of up to 200 milliseconds. If the recording playback is behind the backing track playback, the latency adjustment will be a positive value. To apply a latency adjustment of l milliseconds,the client first checks the playback time of the backing track t_b and sets the playback time of the recording, t_r, to $t_r = t_b + l$. The parent App controller in the client manages these latency adjustments before pausing or starting audio in the playback component.

4.2 Backend with Flask

The primary role of the Flask server is to transcribe solo recordings. The server also handles client requests for transcription status updates. Figure 4-2 outlines the client's and server's roles and interactions in the transcription process.

Solo audio processing and transcription generation are implemented as separate Python scripts. Transcription requires the following contextual information: backing

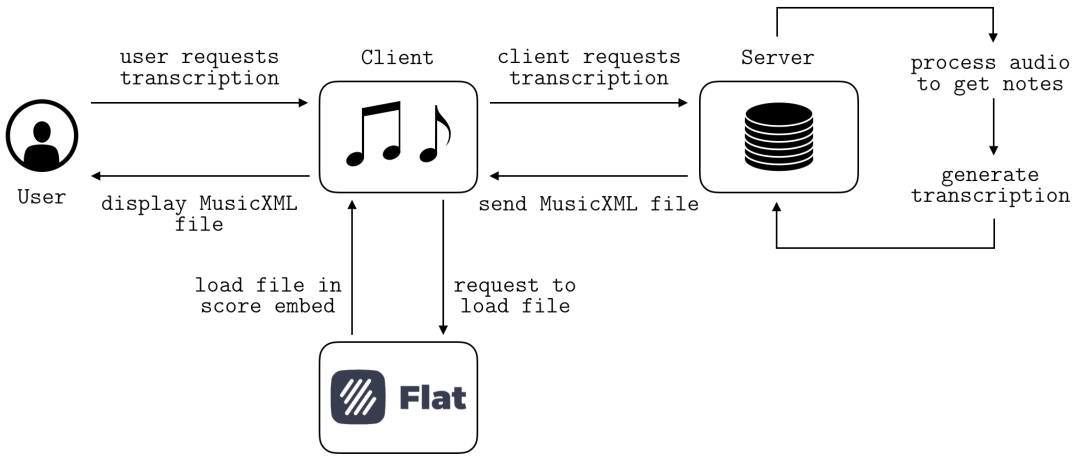


Figure 4-2: Diagram of client-server interactions when handling transcription requests.

track name, instrument key, analysis mode, and latency. The server uses the solo audio file and the contextual information to generate notes and rests (Section 4.2.1) and then to create a MusicXML file for the transcription (Section 4.2.2).

4.2.1 Audio Processing

Figure 3-8 presents a high-level depiction of the audio processing pipeline. Figure 4-3 shows this pipeline in greater detail by introducing the method used in each step and the resulting outputs. Audio processing in this implementation can be broken into the following phases: (1) onset detection and quantization, (2) note creation (pitch detection and duration quantization), and (3) rhythm simplification.

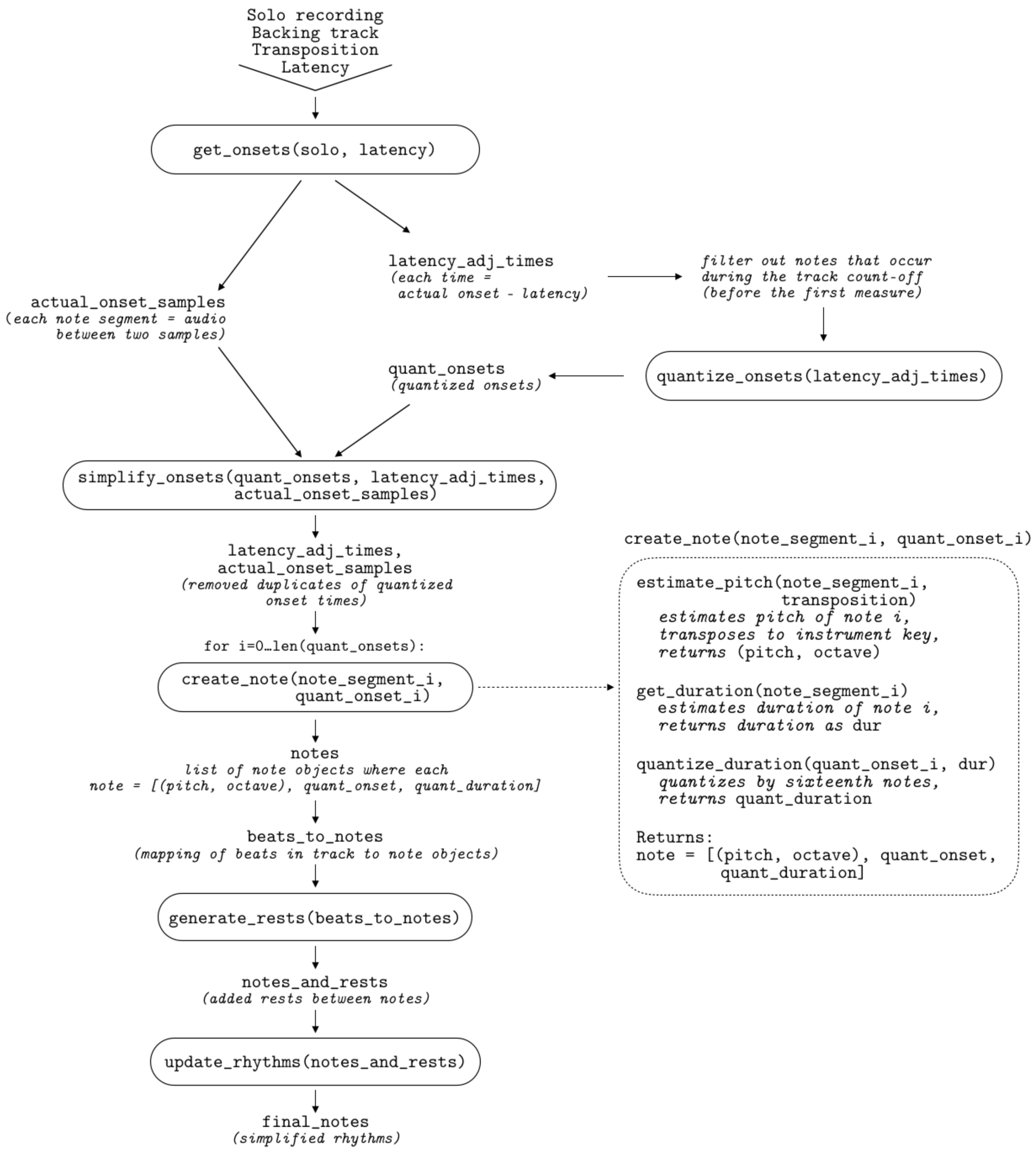


Figure 4-3: Audio processing pipeline. For each step in the pipeline, the diagram includes the relevant method and outputs.

Reference: Audio Processing Methods

As noted in Figure 4-3, the audio processing implementation uses the following methods. We reference these throughout the implementation description.

- `get_onsets(solo, latency)`: identifies note onsets in solo
- `quantize_onsets(latency_adj_times)`: quantizes note onset times (in seconds)
- `simplify_onsets(quant_onsets, latency_adj_times)`: removes duplicates of quantized onsets
- `create_note(note_segment_i, quant_onset_i)`: creates note objects using `estimate_pitch`, `get_duration()`, and `quantize_duration()`
- `estimate_pitch(note_segment_i, transposition)`: estimates pitch and octave of note in audio segment, transposes if necessary
- `get_duration(note_segment_i)`: estimates duration of note in audio segment (in seconds)
- `quantize_duration(quant_onset_i, dur)`: quantizes estimated duration (in sixteenth notes)
- `generate_rests(beats_to_notes)`: fills spaces between notes with rests
- `update_rhythms(notes_and_rests)`: simplifies note rhythms

Reference: Computed Audio Processing Data

We also reference the following outputs throughout the description of the implementation:

- `actual_onset_samples`: time of each note onset (in samples)

- `latency_adj_times`: time of each note onset (in seconds), adjusted by specified latency
- `quant_onsets`: quantized time of each note onset (in seconds)
- `note_segment_i`: represents the i^{th} note segment
- `quant_onset_i`: represents the i^{th} quantized note onset (in seconds)
- `dur`: estimated duration of a note (in seconds)
- `quant_duration`: quantized duration of a note (in sixteenth notes)
- `notes`: list of notes estimated
- `beats_to_notes`: mapping of sixteenth notes in track to notes
- `notes_and_rests`: mapping of sixteenth notes in track to notes and rests
- `final_notes`: final list of notes and rests

Note Format

The audio processing pipeline outputs a list of note objects (notes and rests) to be used in generating a transcription file. Notes and rests are formatted as follows:

```
note = [(pitch, octave), quantized onset time,
        quantized duration (sixteenth notes)]

rest = ["rest", quantized onset time,
        quantized duration (sixteenth notes)]
```

Supporting a sixteenth note as the shortest note duration is necessary to accurately transcribe users' solos. Preliminary transcription experiments and user study results showed that shorter notes (e.g. 32nd notes) were played very infrequently in solos for our selected backing tracks, so we chose not to implement a shorter possible duration. As mentioned in Section 3.4.1, we found that choosing a sixteenth note as the shortest duration resulted in our transcription implementation generating overly specific rhythms. To manage this, our quantization methods still quantize note onsets and durations to sixteenth notes, but we treat an eighth note as the base unit for duration and a sixteenth note as a fraction of the unit duration. The subsection "Rhythm Simplification" in this section further explains this decision.

To simplify rhythm transcription, our quantization implementation assumes that solos do not contain triplets or other tuplets. Quantizing onsets for triplets proves more difficult because these onsets do not occur exactly on-beat. Due to time constraints, we were unable to implement support for triplets. The no-triplet assumption was ultimately a more common source of error in some of the transcription results collected during user studies.

Beat Maps

Each backing track has a related CSV file that maps each sixteenth note of the backing track to a time value. We use these beat maps for quantizing note onsets and durations. The first entry is the time of the first beat of the music in the backing track. We generated these beat maps using Sonic Visualizer by tapping along to the eighth note beats of each backing track and exporting the list of recorded time instants [22]. Tapping sixteenth notes by hand did not produce sufficiently accurate times, so we inserted sixteenth note times into the beat maps at the midpoints between each pair of eighth notes.

Onset Detection and Quantization

The audio processing script first loads the beat map for the backing track used with the solo recording. Next, the script calls `get_onsets()`, which uses the `librosa` onset detection function to identify the note onsets in the solo recording [23]. The `librosa` onset detection function finds peaks in the audio's spectral novelty functions and then backtracks from each peak to more accurately identify onsets as audio segmentation points. `get_onsets()` converts the onset times to samples for indexing into the audio during pitch detection. Each audio segment between samples is considered a potential note. `get_onsets()` then subtracts the latency value reported by the client from the detected onset times for more accurate note quantization results later.

The processing script then identifies onset times that occur before the first measure of the backing track starts (i.e. before or during the count-off clicks) with a sixteenth-note margin of error. These reported onsets are likely due to background noise rather than soloing before the first measure. The script removes these early onsets from `latency_adj_times` and `actual_onset_samples`. We then use `quantize_onsets()` to snap each remaining onset to the nearest sixteenth note time in the beat map. The system sometimes reports multiple onsets that occur less than a sixteenth note apart and quantizes them to the same time. To fix this, the last step of onset detection removes duplicates using `simplify_onsets()` by only keeping onsets closer to the quantized value.

Note Creation: Pitch Detection and Duration Quantization

Each set of audio samples between onsets is considered a potential note. The audio processing script runs `create_note()` for each onset and its corresponding note segment to generate a note object. `create_note()` gets a pitch estimate for the segment using the method `estimate_pitch()`. `create_note()` also estimates the note end time using `get_duration()`. The estimated note duration is the difference between the note onset and the note end estimate. `create_note()` then calls `quantize_duration()` to quantize the note length to an integer number of sixteenth notes. The note object returned contains the pitch estimate, quantized onset, and quantized duration converted to number of sixteenth notes (the unit of duration used in the transcription-generating script). Figure 4-3 summarizes each step of note creation under the `create_note()` description on the right side of the figure.

`estimate_pitch()` in version 2 first estimates the pitch for chunks of samples in an audio segment and determines the most frequent estimate. The first version of Practica originally implemented pitch detection using an autocorrelation method from `librosa` [23]. We changed our implementation in v2 to use `aubio` for pitch estimation after results from the first user study showed that the `librosa` estimates had frequent octave errors [24]. `aubio` provides an optimized implementation of the YIN pitch detection algorithm, which uses autocorrelation with some modifications to reduce estimation errors [25]. `estimate_pitch()` returns a pitch estimate (as a frequency in Hz) and a confidence value for the estimate's accuracy. If the pitch estimate is greater than 0 (implying a note exists rather than empty space) and the confidence is at least 0.6 (determined from some testing during development as a reasonable measure), the estimated frequency is converted to a concert pitch value and octave, transposed using the requested `transposition` value if necessary (solo recorded by an instrument not in concert pitch), and saved in a list of pitches. The reported confidence value for each pitch estimate is also saved in a corresponding list.

After estimating the pitch for every chunk in the note segment audio, `estimate_pitch()` selects the most frequently reported pitch as the collective estimate. In our first user study, we observed that the pitch detection algorithm erred on the higher side of reported frequencies and estimates some pitches as an octave higher. We implemented the following improvement in version 2. To account for potential octave errors, `estimate_pitch()` checks if either (1) the same pitch (one octave down) was reported and the max confidence reported was less than 0.9, or (2) the max confidence value reported in the pitch estimation was less than 0.8. During development, we found that these confidence cutoffs can be reconciled with the original use of a 0.6-confidence cutoff for saving a pitch. When the confidence value falls below 0.9, the estimated pitch itself may be accurate, but the octave is more likely to be incorrect. If either case applies, `estimate_pitch()` moves the most frequently reported pitch an octave down before returning the estimated pitch and octave.

`get_duration()` determines the end time of the note in a given audio segment. This method uses an estimation function from `librosa` to compute the root-mean-square (RMS) energy for each sample in the audio segment as an estimate of loudness. First, this method removes the silence at the start of the audio segment so as not to interpret it as a premature note end. Next, `get_duration()` determines and normalizes the RMS estimate for each frame in the audio segment. The first sample in which the normalized RMS falls below the threshold of 0.05 is considered the note end time. If no samples in the segment have a normalized RMS below the 0.05 threshold, we assume that this note has a louder volume and then double the RMS threshold to identify the note end. This threshold adjustment to accommodate louder notes was implemented in the second version of Practica, as evaluation of the first version showed that some transcription errors occurred due to `get_duration()` not being able to identify note ends for louder notes.

Finally, `quantize_duration()` quantizes the estimated note duration to an integer number of sixteenth notes, the shortest note length used in the transcription script. This method returns the greatest quantized duration that does not exceed the original duration estimated from the note onset and end times. Exceeding the original estimate could accidentally quantize the note duration to a length that extends past the onset of the next note.

After creating a note object for each onset identified in the solo, the audio processing script creates a dictionary data structure, `beats_to_notes`, that maps the time of each sixteenth note in the backing track that appears in the solo to either a note object or a rest. The script first enters all note objects into the mapping at their onset times. Then, the script uses the method `generate_rests()` to fill in any empty entries in `beats_to_notes` with sixteenth rests and then consolidates consecutive rests into a single, longer rest. `generate_rests()` returns a time-ordered list of all notes and rests in the solo (`notes_and_rests`). This method of generating rests is an improvement made in version 2 of Practica as a result of poor quantization results observed in user study 1. In version 1, we measured rest durations by calculating the time between one note's quantized end and another note's quantized onset. We then quantized each rest duration to the nearest sixteenth note. Quantizing rests like how we quantized notes produced errors when some rests were quantized to last slightly longer than the actual time between notes. In these situations, the notes following these longer rests would get pushed to later onset times so as not to overlap. We fixed this error in version 2 and improved our quantization results.

Rhythm Simplification

As mentioned in the above subsection "Note Format," although supporting the transcription of sixteenth notes is necessary for accurately notating solos, our version 1 implementation tended to produce overly specific note onsets and rhythms. For example, note onsets that were slightly off-beat were occasionally quantized to sixteenth notes in between eighth note downbeats, whereas a listener transcribing these notes by hand would correct these onsets to occur directly on the eighth note downbeats. Figure 4-4 shows an example of this difference in note quantization, where the "simplified" result is what a listener would likely transcribe compared to the "original" rhythm generated by the system.

Figure 4-4: Comparison between original transcription of off-beat onsets and simplified transcription that moves notes a sixteenth note earlier to land on downbeats.

Considering duration in terms of sixteenth notes also resulted in our implementation erring on the shorter side when quantizing durations. Using more precise sixteenth-note onsets and durations proved more accurate for solos with consecutive sixteenth notes. Solos with longer notes or more rests between notes often had note durations quantized to odd numbers of sixteenth notes and filled the spaces with sixteenth rests. As an example of such a situation, Figure 4-5 compares how our original implementation returned shorter durations than what a listener would likely report when listening to the same audio. While the actual note durations may have been closer to what the system reported, the musician's intention was likely to play slightly longer notes (and not sixteenth-note denominations of notes). A listener would likely account for this situation and round up these note durations when writing a transcription. These observations show that transcribing sixteenth notes improves accuracy as desired for short and fast notes, but not for longer or more spaced-out notes.

To handle instances in which the note quantization returned overly specific rhythms,

Figure 4-5: Comparison between original transcription of shortened notes and simplified transcription that rounds up note durations.

we added a rhythm simplification step to the end of our audio processing pipeline in version 2. While version 2 still supports sixteenth-note durations, we treat these as fractions and consider an eighth note as the base unit of duration. In line with this decision, we implemented the method `update_rhythms()` to simplify rhythms from sixteenth-note increments to eighth-note increments. `update_rhythms()` simplifies neighboring objects in `notes_and_rests` using the following two rules:

- Simplify slightly off-beat onsets. If the object is a sixteenth rest played on the downbeat of an eighth note, and if the rest is followed by a sixteenth note: replace this pair with a single eighth note that starts when the sixteenth rest originally did. See Figure 4-4 for an example of how our implementation applies this rule to simplify onsets.

- Round up note durations that have an odd number of sixteenth notes and that are followed by a hanging sixteenth rest. If the object is a note that is (1) played on the downbeat of an eighth note and (2) has the duration of an odd number of sixteenth notes, and if a rest follows that note and has a duration of an odd number of sixteenth notes: the note's duration increases by a sixteenth note and the rest's duration decreases by a sixteenth note (e.g. sixteenth note, sixteenth rest becomes eighth note). See Figure 4-5 for an example of how this rule would simplify note durations.

Figure 4-6 shows an example in which both rules are applied to simplify the original rhythm generated during transcription.

Our transcription implementation assumes that all recorded solos are in 4/4 time. We made this assumption because both of the backing tracks currently used in Practica have a 4/4 time signature. Removing this assumption would require careful

Figure 4-6: First measure: original output of note creation. Second measure: results after applying rhythm simplification.

consideration of how to map beats to measures and how to simplify rhythms in solos over backing tracks with different time signatures.

Our system cannot identify the difference between straight-ahead or swung notes. For example, when transcribing swing eighths, the system typically outputs a dotted eighth note and a sixteenth note instead of two eighth notes. Users who play straight-ahead solos will likely not be affected by this. `update_rhythms()` provides some improvement to the transcription of solos with swing note (more so for a sixteenth-note amount of swing than for swing eighths). Because Practica teaches jazz improvisation, the lack of support for swung notes is a weak point in the system. We would certainly revisit this in a future implementation of the system.

While the note creation results could certainly benefit from having more rules for rhythm simplification, implementing more rules proves to be much harder for two reasons. First, implementing these rules requires comparing more notes within longer musical phrases or across measures rather than simply using two consecutive notes. Second, adding more rules to simplify different rhythm combinations could unintentionally over-correct users' solos. For these reasons, improving the transcription accuracy through approaches other than rhythm correction may be more advisable.

4.2.2 Transcription Generation

After processing the audio of a solo recording and saving a list of all notes and rests in the solo, the server uses a transcription generation script to write these objects into a MusicXML file and color-code them based on the selected harmonic analysis mode. Documentation of how MusicXML works, including a reference for all MusicXML elements, is available through the World Wide Web Consortium.[1][2]

The transcription script requires the following inputs:

- File path of saved list of notes and rests in solo
- File path for new MusicXML file
- Key of instrument used in solo (C, B♭, or E♭)
- File path of the CSV file containing harmonic information for the backing track and instrument key combination used for the solo (i.e. a chord map)
- Selected analysis mode (chord tones, tensions, dissonant notes, or modal)

Chord Maps

For each combination of backing track and instrument key, a CSV file exists in the server to save the relevant harmonic information for each eighth note in the backing track (i.e. a chord map). Each entry in one of these CSV files contains the following information:

- beat: the eighth note in question (1 through 8)
- measure: the number of the measure in which this note occurs
- notes: the notes in the scale for the current chord (e.g. F,G,A,B♭,C,D,E for FM7); used when determining whether to color-code notes
- harmony: the XML string used to write the chord symbol in the MusicXML file

Color Map

Section 3.5.3 describes and provides examples of each analysis mode. The transcription implementation maintains a mapping of each analysis mode to a list of colors.

[1] MusicXML documentation: `https://www.w3.org/2021/06/musicxml40/`

[2] Element reference: `https://www.w3.org/2021/06/musicxml40/musicxml-reference/element-tree/`

Each list contains seven color entries, one for each scale degree. If a note should be color-coded based on the criteria of the selected analysis mode, the implementation gets the expected color by indexing into the color list using the note's degree in the scale. Appendix A provides the mapping of each analysis mode to its list of colors.

Transcription Process

Algorithm 1 shows pseudocode for the transcription implementation. Algorithm 2 shows pseudocode for the method used in transcription to write MusicXML for notes and rests. Our explanation goes into greater detail for algorithm 2 in particular because it contains the logic for color-coding notes for harmonic analysis.

Section 4.2.1 introduces and explains the assumption that the time signature of each solo is always 4/4. Similarly, section 4.2.1 also describes why an eighth note is the unit of note duration in the audio processing implementation. The transcription implementation follows the same assumptions. The system generates the score for a solo one measure at a time and maintains three counters: (1) the number of note objects (notes and rests) transcribed so far, (2) the current measure number, and (3) the number of eighth notes left in the current measure.

Before adding each note object to the transcription, we identify the underlying chord in the measure using the chord map. If the current chord is different from the previous one, we add a MusicXML string to write a new chord symbol. We then determine whether the note object extends into the following measure. If it does, we need to account for a tied note when creating the object's MusicXML element.

When evaluating the first version of Practica, one of the errors found when evaluating transcriptions was that our implementation failed to handle notes tied across measures. Even if a note's duration extended into a second measure, the implementation would add the full note's duration to a single measure. This mistake also impacted the placing of other notes, as the next note to add to the transcription would then be incorrectly added at the start of a new measure. To resolve this in version 2, we check whether the duration of the new note object exceeds the number of eighth notes left in the current measure. If so, we split the note object into two

Algorithm 1 Transcription generation

Input

notes	list of notes returned from audio processing
chordmap	map of measures and beats to chords and scales
mode	requested harmony analysis mode

Output

score	list of MusicXML lines to write to transcription file

```
n ← 0                                               ▷ n = number of notes read
m ← 0                                               ▷ m = current measure
b ← 8                                               ▷ b = beats left in current measure
score ← []                                          ▷ score = output list of lines to write
while n < |notes| do
    m ← m + 1
    add MusicXML new measure line to score

    while n < |notes| and b > 0 do
        current_beat ← 8 − b
        chord ← chordmap[m][current_beat]
        add MusicXML line for chord to score

        obj ← notes[n]                              ▷ next note or rest to add to score

        if obj[duration] > b then
            first_obj ← write_note_obj(obj, b, chord, mode)
            add first_obj to score

            m ← m + 1
            add MusicXML new measure line to score

            second_obj ← write_note_obj(obj, obj[duration]−b, chord, mode)
            add second_obj to score
            b ← 8−(obj[duration]−b)

        else
            object_line ← write_note_obj(obj, obj[duration], chord, mode)
            add object_line to score
            b ← b−obj[duration]
        end if

        n ← n + 1
    end while
end while
if b > 0 then         ▷ after reading all note objects, if last measure is incomplete
    add MusicXML line for rest with duration b to score
end if
return score
```

Algorithm 2 Writing MusicXML lines for notes and rests

Input

obj	note object with type rest or (pitch, octave)
d	duration of note object
chord	chord map entry, maps "scale" to chord scale
mode	requested harmony analysis mode

Output

line	MusicXML line representing note or rest

Uses global `color_map`, which maps each analysis mode to a list of colors where index represents scale degree.

```
if obj[type] == rest then
    line ← MusicXML line for rest of length d
else
    (pitch, octave) ← obj[type]
    scale ← chord["scale"]
    colors ← color_map[mode]                  ▷ use to get color attribute of note
    color ← black                ▷ default note without color-coding is black
    if mode == dissonant then
        if pitch ∉ scale then color = blue
                              ▷ all dissonant notes are marked blue in this mode
        end if
    else if pitch ∈ scale then
        scale_degree ← scale.index(pitch)
        color ← colors[scale_degree]
    end if

    line ← MusicXML line for note of length d with pitch, octave, and color
end if
return line
```

shorter-duration objects, one to fill the current measure and another to start the next measure with the remainder of the object's duration. Figure 4-7 shows an example of how the version 1 transcription would overfill a measure instead of creating a tied note. Figure 4-8 shows an example of how the version 2 transcription would correctly divide the note and tie it across two measures.

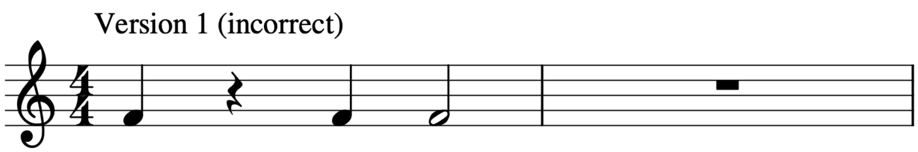


Figure 4-7: Example of failure to handle tied notes in version 1.

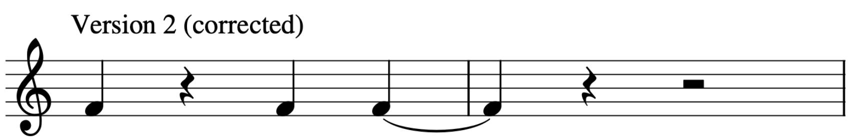


Figure 4-8: Example of correction in version 2 to handle tied notes.

After determining whether we need a tied note, we write a MusicXML string for the object. Algorithm 2 provides pseudocode to explain how our implementation generates MusicXML strings for rests and pitched notes based on their attributes (duration and either "rest" or a tuple of pitch and octave) and the chord in the progression at the current measure and beat. Rests require little examination because they only include duration information. Adding pitched notes requires more care in order to implement harmonic analysis through color-coding notes.

We apply color to a note by including a `color` attribute in the note's MusicXML string. For example, a green note would have the attribute `<note color="#00FF00">`. Applying the color-coding for the selected harmonic analysis mode ("chord tones," "tensions," "dissonant notes," or "modal") requires comparing the pitch of a note to the pitches specified in the relevant chord scale. The entry in the chord map for the current chord contains a list of pitches that comprise the chord scale. Each analysis mode has different criteria for whether to apply color-coding. If the "dissonant notes" analysis option is selected, the script colors a note if the pitch is not found in the relevant chord scale. If the "modal" option is selected, the script method uses the same color for any pitch found in the relevant Dorian mode. The "chord tones" option identifies if the pitch is the root, 3rd, 5th, or 7th note of the scale. The "tensions" mode identifies if the pitch is the 9th, 11th, or 13th note of the scale. For the "chord tones" and "tensions" modes, each scale degree has a distinct color assignment.

The pitch detection method in the audio processing script represents any altered notes using flats instead of sharps, as jazz scores are more frequently written with flats to accommodate transposition for horns. The audio processing script uses the same mapping of frequencies to pitches for every note regardless of whether the underlying chord scale uses sharps or flats. However, some scales for the chords in the backing tracks include sharps, so if the current note's pitch is not part of the retrieved chord scale, the note-writing algorithm must check if any enharmonic pitches are in the chord scale. For example, if a user played an F♯ over a D7 chord, the audio processing script would have identified that note as a G♭. To ensure that this note is colored correctly, the transcription implementation would check if the chord scale contained either G♭ (the input pitch) or F♯ (an enharmonic pitch). Without this enharmonic check, the implementation would either accidentally label G♭ as a dissonant note (in the "dissonant notes" analysis mode) or would fail to identify the note as a chord tone (in the "chord tones" analysis mode).

After adding all of the note objects returned from audio processing to the transcription, the implementation checks if the final measure contains fewer than the required number of sixteenth notes (i.e. whether the measure is broken). The last note object of a solo does not necessarily occur at the end of a measure, so in some cases, a trailing amount of sixteenth notes remains in the last measure of the transcription. If this situation occurs, the script adds rests to resolve this trailing amount.

4.3 Deployment with Heroku

Practica is deployed to Heroku at the url `https://practica-music.herokuapp.com/` using Gunicorn to serve the Flask application concurrently. By including a copy of the production build of the React application in the Flask server, the Flask application can also serve the React application. The application does not currently use a database or cloud storage resource for storing users' audio files or transcriptions after they finish using Practica, as these files are not intended to be shared with other application users or saved for future use in the application. However, implementing a storage system for users' audio files would be necessary in a future revision of the application.

Chapter 5

Testing and Evaluation

Evaluating the system consisted of two user studies at different stages of project development. After designing the system, we implemented version 1 of Practica and evaluated it in the first user study. We then improved the system based on participant feedback to create Practica version 2, which we evaluated in the second user study. Both studies evaluated usability of the system, though the survey used during the first study approached usability from a technical perspective, while the survey for the second study focused more on system usability as a tool for music education.

The participant pool included seventeen total users, six of whom participated in both studies. Participants were recruited through personal connections and emails sent to MIT. The participants demonstrated a wide range of musical experience levels. Seven participants had played an instrument for eight or more years. Almost one-quarter of participants had played an instrument for at most two years. Figure 5-1 shows the distribution of years of musical experience across all participants.

Although some participants had extensive experience playing a musical instrument, many participants reported that their musical education focused primarily on playing technique with relatively less music theory. Four participants specifically listed improvisation as part of their musical education. Roughly two-thirds of the participants had little to no experience with music transcription, while the remaining third reported an intermediate experience level with transcription. Five users reported at least a beginner-to-intermediate level of experience with musical improvisation, and the remaining users all reported little to no improvisational experience. Figures 5-2 and 5-3 summarize the participants' self-reported experience in transcribing, arranging/composing, and improvising music.

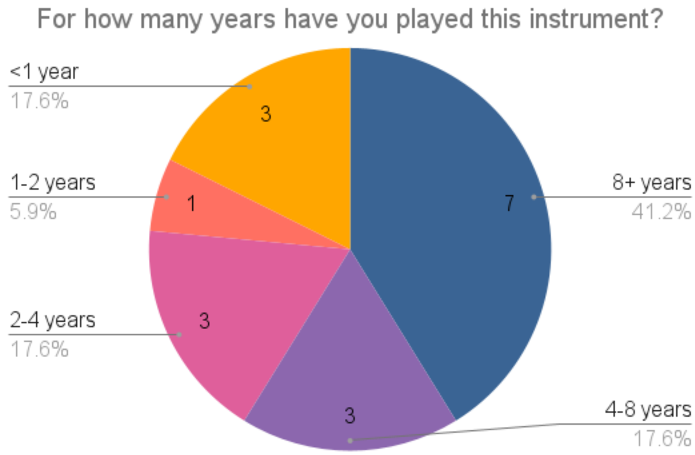


Figure 5-1: Distribution of years of instrument experience reported.

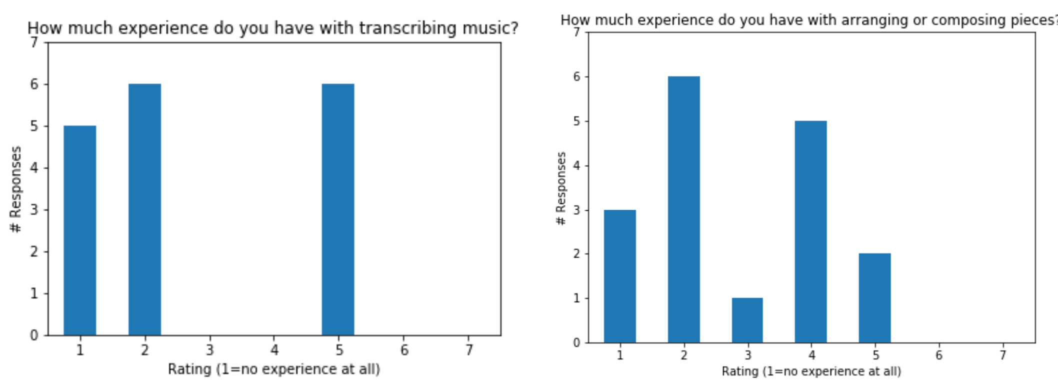


Figure 5-2: Participants' self-reported experience with transcribing (left) and arranging or composing music (right), on a scale of 1 (no experience) to 7 (very experienced).

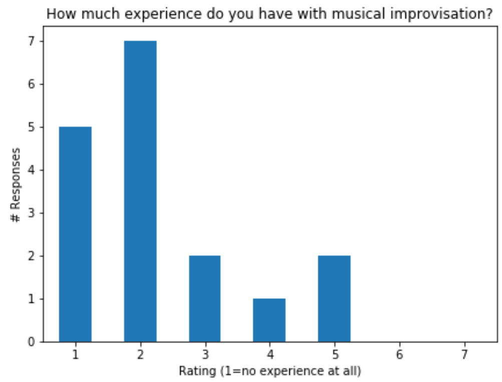


Figure 5-3: Participants' self-reported experience with improvisation on a scale of 1 (no experience) to 7 (very experienced).

5.1 First User Study

The first user study evaluated the first version of Practica (which we reference later as v1). Version 1 of Practica included the following differences from version 2 (v2):

- User study 1 only tested the transcription and analysis of solos recording using "The Nearness Of You" as a backing track. We had not testing transcribing solos recorded with "So What" before conducting the first study.
- The audio playback and score sections in the interface were barely linked (e.g. clicking "play" in one section had no impact on the other).
- Version 1 did not contain any help sections (either for soloing or how to use the app).
- The only harmonic color-coding visible in user transcriptions highlighted notes in red if they were not part of the expected chord scale (i.e. only "out" notes were flagged).
- The audio processing implementation used a pitch detection method from `librosa`. This method used autocorrelation with no modifications (instead of the YIN pitch detection algorithm used in v2).
- The audio processing implementation generated quantized rests in the same way

it quantized notes (snapping to the nearest sixteenth note) instead of generating notes first and then filling empty beats with rests.

- The audio processing implementation did not include a rhythm simplification step.

The primary objectives of the first user study were to (1) collect feedback on the functionality of each component in the system, and (2) determine which new features to add to version 2 of the system.

Participants and Protocol

Nine users participated in the first study. Five participants had at least four years of experience playing their instrument, two participants had between two to four years of experience, and two had at most two years of experience. Most users described themselves as proficient in performance, playing technique, and basic music theory. Six participants in this study had little to no experience with musical transcription or improvisation. The remaining three participants described themselves as having an intermediate level of transposition experience and a beginner-to-intermediate level of improvisation experience. These three participants had more experience in general with playing jazz music, while the other participants identified as classical musicians or did not identify their musical experiences with a particular genre.

Each participant was tasked with the following:

1. Record an eight-measure solo using "The Nearness Of You" as a backing track.
2. After recording a solo, request a system transcription of the solo.
3. Examine the generated transcription. If possible, correct any mistakes made by the system. Download and submit both transcription versions and the solo audio file for comparison.
4. Complete a post-test survey (Appendix C) about (i) personal musical education experience, (ii) evaluation of the system controls, (iii) evaluation of transcription, and (iv) potential new features to add to the system.

Evaluating Transcription Accuracy

Unfortunately, the participants agreed that the system was generally inaccurate in transcribing pitches and rhythms. On a scale of 1 to 7 (with 1 being very inaccurate), users gave the rhythmic accuracy of the system transcriptions an average rating of 2.86. To evaluate transcription accuracy, we analyzed the differences in detected onsets, pitches, quantized onsets, and durations. Evaluating the accuracy of onset detection involved identifying the time of each note onset in a solo by hand using Sonic Visualizer, generating the estimated note onsets using the system's audio processing code, and calculating the F1 score using the estimated and actual note onsets [22]. We chose to evaluate onset detection using the F1 score because this score considers both false positives (reported onsets of nonexistent notes) and false negatives (missing note onsets). To calculate the F1 score using these sets of note onsets, we used the `mir_eval` Python library for the evaluation of music information retrieval systems [26]. We calculated the F1 score for each solo using two different error thresholds: 50 ms, which is the default window for the `mir_eval` onset evaluation implementation, and 25 ms, which the author of the paper that inspired the `mir_eval` onset evaluation implementation (Sebastian Böck) recommended as a more critical window for comparing different onset detection methods [27].

To evaluate the pitches, quantized onsets, and durations of the notes in each solo, we transcribed each solo by hand and compared the expected transcription with the result of system transcription. When evaluating these three transcription components, we did not penalize the system if the transcription was missing any expected notes, as the F1 score for the transcription's onset detection would reflect the missing notes as false negatives. Pitch correctness was considered a binary value without a threshold. Note onset quantization (which we refer to simply as quantization) and duration (which is technically also quantized) were both evaluated using an error threshold of a sixteenth note. When evaluating transcriptions, we also compared the system outputs with transcriptions produced by other music production or analysis software applications such as AnthemScore and Logic Pro X [7] [28]. In these alternative transcriptions, various outputs showed sixteenth- or eighth-note errors, so we felt that using a sixteenth-note error threshold would reasonably evaluate the transcription implementation in Practica.

We found only four of the nine audio files collected usable for evaluation due to either audio quality issues that made transcription by hand difficult or serious transcription errors that made system- and human-generated transcriptions incomparable. Because the first study produced much fewer usable samples than the second, we sought to create a closer comparison of the results from each study. In the second study, we used a sample set of 22 solos to evaluate v2. To increase our test set for study 1, we took these 22 solos from study 2 and generated transcriptions for them using the v1 audio processing and transcription implementation. The v1 implementation crashed when processing one of these recordings, so we removed that solo from the test set. The final test set of solo recordings for study 1 includes the original four usable solos from study 1 and 21 solos from the study 2 test set. Analyzing transcriptions of solos from study 2 made with v1 code also allowed us to more directly compare the results of the v1 and v2 transcription implementations.

Table 5.1: Mean F1, precision, and recall for study 1, sample $N = 25$ (4 from study 1, 21 from study 2)

Window	F1	precision	recall
50 ms	0.772	0.696	0.887
25 ms	0.645	0.581	0.739

Table 5.2: Mean transcription accuracy for study 1, sample $N = 25$ (4 from study 1, 21 from study 2)

pitch	quantization	duration
0.758	0.637	0.777

Table 5.1 shows the results of F1 evaluation for the test set of 25 solos. Table 5.2 shows the mean pitch, quantization, and duration accuracy for this test set. As expected, calculating F1 scores with a smaller window of 25 ms produced worse results. The recall of our onset detection method was better than its precision. Comparing system- and hand-generated transcriptions supports this observation, as the system tended to add extraneous notes (usually at points in recordings where background noise was louder and was registered as onsets) rather than leave out notes. All three metrics in Table 5.2 showed poor transcription accuracy. Our pitch detection method frequently erred on the higher side of reported frequencies and returned pitches as an octave higher. The reported quantization accuracy was particularly low. Upon closer examination of our quantization implementation, we believe this problem could be a result of quantizing rest durations in the same way we quantized note durations. As Section 4.2.1 describes, this method returned some overly long rests and pushed some quantized note onsets back when compiling the full list of notes and rests in the solo. It is likely that the quantization accuracy also suffered because our transcription implementation does not support triplets or swung notes. As we discussed in Section 4.2.1, because these notes do not occur on a sixteenth-note grid, our implementation has difficulty with accurately transcribing their onsets and durations. Rhythm simplification was not implemented yet in version 1, so the transcription implementation was unable to fix some of these errors before returning notes.

To further examine these transcription results, we created one grouping of recordings by backing track and another grouping by instrument. Grouping recordings by backing track showed little variation in the calculated precision, recall, or F1 scores, as seen in Table 5.3 and Table 5.4. The average quantization accuracy for recordings that used “So What” as a backing track was lower by 0.232 than for recordings that used “The Nearness Of You.” We did not change the beat map used for transcribing solos over “So What” between versions 1 and 2, nor did we add modifications to version 2 to transcribe “So What” solos more accurately. Our transcription results in study 2 (Section 5.2) did not show the same difference in quantization accuracy between backing tracks used. For these reasons, we believe that the rest quantization error and lack of rhythm simplification mentioned earlier were likely responsible for the especially low quantization accuracy for “So What” solos in study 1.

Table 5.3: Mean F1, precision, and recall for study 1, grouped by backing track. Window = 50 ms. Sample $N = 25$ (4 from study 1, 21 from study 2)

backing track	F1	precision	recall
nearness	0.779	0.706	0.883
so what	0.762	0.680	0.894

Table 5.4: Mean transcription accuracy for study 1, grouped by backing track. Sample $N = 25$ (4 from study 1, 21 from study 2)

backing track	pitch	quantization	duration
nearness	0.761	0.714	0.762
so what	0.754	0.482	0.808

Tables 5.5 and 5.6 show that grouping recordings by instrument revealed slightly greater variation in precision, recall, and F1 scores, with the average scores for guitar recordings falling at most 0.1 lower than those for piano recordings. The average pitch, quantization, and duration accuracy for piano recordings fell between 0.06 and 0.08 lower than the averages reported for guitar recordings. We found that when transcribing guitar recordings, our system tended to add extra onsets for slide notes or various percussive noises made by guitar strings. This overestimation of onsets could explain why the reported F1 scores were lower for guitar recordings.

Table 5.5: Mean F1, precision, and recall for study 1, grouped by instrument. Window = 50 ms. Sample $N = 25$ (4 from study 1, 21 from study 2)

instrument	F1	precision	recall
guitar	0.730	0.639	0.867
piano	0.808	0.736	0.918

User Feedback

Feedback regarding the recording and playback controls was generally positive. Six participants described the system recording and playback controls as very easy to use,

Table 5.6: Mean transcription accuracy for study 1, grouped by instrument. Sample $N = 25$ (4 from study 1, 21 from study 2)

instrument	pitch	quantization	duration
guitar	0.806	0.676	0.833
piano	0.731	0.612	0.753

and the other three participants said the playback controls were of middling difficulty. Users liked the clean and simple design of the playback system and said that the functionality was obvious. Some text labels on the control buttons (e.g. "master play" or "latency") confused users, and a few participants mentioned that the system interface would benefit from using more icons and less text to convey information. Some participants found it unclear if they should use the playback controls in the main recording and playback component or in the score component.

The participants' responses were split evenly on the usability of the score controls. Four participants said the score controls were fairly easy to use. Three of the other five users said the score controls were fairly difficult to use, and the remaining two users said the score controls were of middling difficulty. Participants liked the appearance of the score interface and found it easy to change note pitches. However, some users reported that fixing note rhythms and durations was more difficult. The primary frustration with the score component was that the score controls were not synchronized with the playback controls. The cursor tracking playback progress (i.e. the "now bar") in the score component did not align with the actual playback of the backing track or solos. Additionally, clicking the play or pause buttons in the score component (automatically included with the Flat embed object) did not trigger changes in the system's audio playback component. Finally, users were unable to drag the now bar to different positions in the score (a common feature in some music recording and production software applications). Unfortunately, this last concern regarding the now bar is not something that could be implemented in a new iteration of the system. The Flat embed does not support dragging the score cursor and requires users to click to different points in the score to update the cursor position.

Part of the user test involved asking participants for their opinions on two proposed features that we were considering implementing in version 2: (1) the system could evaluate users' transcriptions of their own solos and provide feedback on the accuracy of users' transcriptions; and (2) the system could show users annotations about how the notes played in their solos fit into the underlying chord progressions in the backing track. Participants said that the first feature could be very useful, but only if the accuracy of the system transcription improved. Every participant said that they would find the second feature very helpful. In the post-test survey, multiple participants mentioned that the first feature would be useful for practicing transcription, but that the second feature would be more valuable for learning how to improvise over the backing track and for developing a stronger understanding of music theory.

When asked whether the system should display harmonic annotations in the score itself using color-coding or in a separate written panel, most users were ambivalent towards the mode of presenting information. Participants were also asked what they would like to see from a practice tool that would help them learn how to improvise. Some participants expressed interest in seeing tips on what to play in their solo, perhaps in a tutorial format. Almost every participant wrote that they would be interested in learning more about how the notes they played fit into the chords in the backing track. Seeing the transcription color-code "out" notes that were not part of the underlying chords in red was not as useful because (1) this function did not convey as clearly which notes would sound better in solos, and (2) teaching pitches as "right" or "wrong" for different chords failed to teach how dissonant pitches could be used appropriately in solos.

The participant feedback on the two proposed features suggested focusing on implementing harmonic analysis of what users played in relation to chord progression rather than on teaching users how to transcribe music. The first feature (evaluation of user-written transcriptions) was ultimately not considered for development of version 2 due to both the lack of user interest in the feature and the difficulty of implementation.

As we described in Section 3.5.1, one of the participants (Tony Terrasa) gave us extensive feedback for these proposed features. After the first user study, we spoke to Tony about other possible improvements to the educational content of the application. The suggestions Tony provided also influenced us to implement the second proposed feature and to focus our educational content on teaching introductory harmony in greater detail.

Improvements Based on User Feedback

The combination of participant feedback regarding system usability and quantitative results for transcription accuracy helped define clear priorities for the second iteration of system development:

- Improve transcription accuracy. Both pitch detection and rhythm generation would need to improve in order to make the application a viable educational tool. Participants considered the v1 transcription implementation a limiting factor in system usability. The accuracy values observed after evaluating transcriptions support this conclusion.

- Link audio playback and score components. More closely connecting interactions between the audio playback and score components would resolve one of the primary difficulties participants reported when using the application.

- Focus the educational approach on teaching harmony rather than transcription. Providing more information about how the notes people played in their solos fit into the underlying chords would more effectively introduce people to improvisation. Adding tips on what to play in solos would help encourage beginners to improvise.

- Add help sections to the system. Providing more information about how to use each component (i.e. providing a tutorial or FAQ) would make the application more understandable for users. Creating a help section for each backing track to store relevant soloing tips and harmonic information would also be useful.

5.2 Second User Study

The second user study evaluated version 2 of Practica. This included fourteen participants, six of whom participated in the first study. The second user study placed greater focus on examining the educational impact that the system could have on music students. Version 2 included the following updates:

- Linked interactions between audio playback and score components. Any pause, play, or restart actions in one component cause the same event in the other component. The score component more accurately updates progress in the sheet music based on audio playback.

- Added three new help sections. The first is an application walk-through for first-time users. The other two help sections are customized for each backing track to offer relevant soloing tips and and scales for the relevant chords (Appendix B).

- Created four analysis modes for color-coding transcriptions (Appendix A). We redesigned the original analysis mode of highlighting "out" notes to be a "dissonant notes" mode, and we also added color-coding for chord tones, tensions, and notes in the Dorian mode.

- Switched Python packages for pitch estimation. V2 uses an `aubio` pitch estimation method that implements the YIN pitch detection algorithm. The YIN algorithm builds on autocorrelation (implemented by the method we used from `librosa` in v1) with error-reducing modifications.

- Improved octave estimates in pitch detection. The v2 pitch detection algorithm checks if multiple octaves of the same pitch were estimated and picks the lower octave depending on an estimate confidence threshold.

- Improved volume thresholds for note duration estimation. The v2 note detection algorithm increases its volume threshold for note ends depending on the minimum volume identified in a note segment.

- Redesigned rest generation method. Instead of generating notes by quantizing the duration of silence (in seconds) between notes, v2 creates a mapping of sixteenth-note beats to notes and fills any empty entries with rests. The rest generation method no longer returns quantized rests that exceed the duration between a note end and the next note onset.

- Added rhythm simplification method. V2 simplifies slightly off-beat onsets and rounds up note durations to remove hanging sixteenth rests. Though v2 does not support triplets or swing notes, rhythm simplification now modifies system-generated rhythms to more closely resemble how a person would transcribe these types of notes.

- Added support for tied notes across measures. The v1 transcription implementation would overfill a measure if a note duration exceeded the remaining space. The transcription implementation in v2 corrects this mistake and creates tied notes across measures when necessary.

Participants and Protocol

The participant pool for the second study showed a similar distribution of musical experience levels as the first study. About two-thirds of the participants had little experience with transcription or improvisation, while the remaining participants expressed some confidence or experience with these activities.

The second user study presented participants' tasks as more of an educational experience. Before starting the test, participants were asked to read the application instructions in the "How do I use this app?" help section to familiarize themselves with the system. The study involved two phases of testing (one for each backing track) and a post-test survey to reflect on users' experiences (Appendix C). Testing involved the following tasks:

Phase 1: "The Nearness Of You"

1. Record a monophonic solo through at least the first eight measures of the and adjust the latency value to line up the audio of the recording and backing tracks.

2. Have the system transcribe the solo. Test at least one of the three available modes (chord tones, tensions, and dissonant notes) and explore the color-coding results. Export this transcription and audio.

3. Click on "Harmony & soloing help" and read the help section for "The Nearness of You."

4. Using at least one of the soloing tips described in the help section, record another solo.

5. Have the system transcribe the solo. Test all three available modes (chord tones, tensions, and dissonant notes) and explore the color-coding results. Export this transcription and audio.

Phase 2: “So What”

1. Record a monophonic solo through at least the first eight measures and adjust the latency value to line up the audio of the recording and backing tracks.

2. Have the system transcribe the solo. Test the modal analysis mode and explore the color-coding results. Export this transcription and audio.

3. Click on "Harmony & soloing help" and read the help section for "So What."

4. After reading about the Dorian mode in the help section for “So What,” record another solo.

5. Have the system transcribe the solo. Compare the color-coding results for the modal and chord tones modes. Export this transcription and audio.

After completing both phases of the study, participants completed a post-test survey (Appendix C) about (1) personal musical education experience (if they did not answer these questions earlier by participating in the first user study), (2) evaluation of soloing help and harmonic analysis, (3) evaluation of transcription performance, (4) evaluation of system controls, (5) reflection on the current system, and (6) potential new features to add to the system.

Evaluating Transcription

As in the previous user study, the participants in the second study stated that the system was somewhat inaccurate at transcribing rhythms. However, the average user rating of the rhythmic accuracy of the system transcriptions increased in the second study by one point from 2.86 to 3.86 (on a scale of 1 to 7, with 1 being very inaccurate and 7 being very accurate). The user feedback regarding pitch accuracy improved in the second study as well, noting that while some errors still occurred with detecting the correct pitches, the errors they noticed more frequently in the pitch detection were missing pitches or extra notes. Based on this feedback and the results of F1 and transcription evaluation below, we think that the updates made to pitch detection and rest generation were responsible for the transcription performance improvement observed in version 2 of Practica.

We repeated the evaluation process from user study 1 to analyze the collected transcriptions. Certain transcriptions generated in this user study were affected by a program error that went unnoticed until after the study ended. The error involved generating an extraneous closing XML tag in some measures with longer rests, which affected how rhythms were displayed in the generated scores. We were able to recreate the expected outputs without the effects of the program error similarly to how we created additional transcriptions for evaluation in the first user study (mentioned in Section 5.1). For the transcriptions affected by this error, we re-ran the same transcription code from v2 that was deployed during the second user study. When this code produced the expected list of notes and rests to write into a MusicXML file, we compared the beats (for quantization), note durations, and pitches reported in each list to the expected values in the handwritten transcriptions. This method allowed us to fairly evaluate the pitch, quantization, and duration accuracy of the audio processing and transcription implementation in version 2.

Table 5.7: Mean F1, precision, and recall by study (50 ms window).
Study 1 sample $N = 25$ (4 from study 1, 21 from study 2), study 2 sample $N = 22$.

	F1	precision	recall
Study 1	0.772	0.696	0.887
Study 2	0.765	0.690	0.886

Table 5.8: Mean F1, precision, and recall by study, grouped by backing track (50 ms window).
Study 1 sample $N = 25$ (4 from study 1, 21 from study 2), study 2 sample $N = 22$.

	backing track	F1	precision	recall
Study 1	nearness	0.779	0.706	0.883
	so what	0.762	0.680	0.894
Study 2	nearness	0.775	0.705	0.879
	so what	0.753	0.671	0.895

Table 5.7 shows little change between the F1, precision, and recall scores reported in study 1 compared to study 2. The reported F1 scores for v1 and v2 are also similar when grouped by backing track or instrument, as seen in Tables 5.8 and 5.9. Because both v1 and v2 use the same onset detection method, we expected to see similar F1 scores across both studies.

On average, the second user study showed improvements in all three accuracy metrics of pitch, quantization, and duration, as shown in Table 5.10. These results confirm that changing our pitch estimation method to the `aubio` YIN implementation, adding a correction for higher-octave pitch estimates, adding a rhythm simplification method, and redesigning our rest generation method collectively improved audio processing and transcription in v2.

As seen in Tables 5.11 and 5.12, we did not observe any noticeable differences in pitch, quantization, or duration accuracy when comparing recordings grouped by backing track or instrument. In the first study, we also found that transcription accuracy was fairly similar for both guitar and piano solos (Table 5.6). However, we saw a significant increase in the quantization accuracy for “So What” solos in study 2 (Table 5.11). As mentioned in Section 5.1, in study 1, transcriptions for “So What” solos reported especially low quantization accuracy. We did not update the “So

Table 5.9: Mean F1, precision, and recall by study, grouped by instrument (50 ms window).
Study 1 sample $N = 25$ (4 from study 1, 21 from study 2), study 2 sample $N = 22$.

	instrument	F1	precision	recall
Study 1	guitar	0.730	0.639	0.867
	piano	0.808	0.736	0.918
Study 2	guitar	0.735	0.639	0.889
	piano	0.810	0.729	0.937

Table 5.10: Mean transcription accuracy by study.
Study 1 sample $N = 25$ (4 from study 1, 21 from study 2), study 2 sample $N = 22$.

	pitch	quantization	duration
Study 1	0.758	0.637	0.777
Study 2	0.930	0.964	0.934

What" beat map between versions, nor did we modify the v2 transcription algorithm to behave differently when transcribing "So What" solos. For this reason, we think that perhaps our v1 rest generation method was responsible for the low quantization accuracy observed in "So What" transcriptions in study 1.

Table 5.11: Mean transcription accuracy by study, grouped by backing track.
Study 1 sample $N = 25$ (4 from study 1, 21 from study 2), study 2 sample $N = 22$.

	backing track	pitch	quantization	duration
Study 1	nearness	0.761	0.714	0.762
	so what	0.754	0.482	0.808
Study 2	nearness	0.938	0.965	0.943
	so what	0.920	0.966	0.922

Table 5.12: Mean transcription accuracy by study, grouped by instrument.
Study 1 sample $N = 25$ (4 from study 1, 21 from study 2), study 2 sample $N = 22$.

	instrument	pitch	quantization	duration
Study 1	guitar	0.806	0.676	0.833
	piano	0.731	0.612	0.753
Study 2	guitar	0.953	0.957	0.964
	piano	0.923	0.974	0.947

User Feedback

	1	2	3	4	5	6	7	Mean	Std
How helpful was the information in the "How do I use this app?" section?	0	0	0	1	5	5	3	5.71	0.88
How helpful was the "Chord Tones with The Nearness of You" section?	0	0	0	1	2	4	7	6.21	0.94
How helpful was the "Modal Jazz with So What" section?	0	0	0	3	2	4	5	5.79	1.15

Figure 5-4: Summary statistics for users' evaluations of the help sections.

Figure 5-4 summarizes participants' ratings of the three help sections in the application. Participants provided generally positive feedback about the help sections. When asked to provide their opinions of the various help sections, users did not comment much about the section describing how to use the system. The help section "Chord Tones with The Nearness Of You" received solidly positive feedback with an average rating of 6.21 on a scale of 1 to 7, with 1 describing the section as "not helpful at all" and 7 as "very helpful." Users appreciated that this help section included a "good background to explain the theory behind certain soloing decisions." Participants also gave positive feedback regarding the soloing tips (Appendix B) because the tips "suggested simple actionables (and the user) could immediately apply them." One user with more improvisation experience noted that they "tended to play similar style solos, but reading that section helped broaden my horizons." To improve this help section, users suggested adding a summary of the soloing tips to the transcription component so that they could read the tips while recording instead of opening the help section as a modal component.

The help section "Modal Jazz with So What" received positive but slightly more mixed feedback from users with an average rating of 5.79 (Figure 5-4). Every participant mentioned that this help section was at least somewhat helpful, but that adding more specific soloing tips in this section (with either harmonic or rhythmic suggestions for greater variety) could have made it as useful as the chord tones help section. Users with less music theory experience expressed that including more detailed explanations for the concepts of modal jazz and the Dorian mode could have provided them with a better foundation for understanding scale modes.

	1	2	3	4	5	6	7	Mean	Std
How easy to use are the transcription/analysis controls?	0	0	1	1	3	7	2	5.57	1.05
How accurate do you think the system was in transcribing rhythms?	1	1	1	8	2	1	0	3.86	1.23
How helpful were the color-coded results from transcription using different analysis modes?	0	0	1	3	3	5	2	5.29	1.16

Figure 5-5: Summary statistics for users' evaluations of transcription controls and harmonic analysis.

Figure 5-5 shows a summary of users' ratings of the transcription controls and the use of color-coding to convey harmonic information in their transcriptions. Users generally provided positive feedback in these evaluations. Participants appreciated that the analysis modes reinforced concepts in the help sections and provided clear visual cues about (1) which notes were in the chord or mode, (2) which chord tones users were playing more frequently, (3) which chord tones users could try to use more to create tension, and (4) which chords gave users more trouble with identifying chord tones. Some users particularly enjoyed the educational reinforcement of chord tones and tensions that color-coding provided, noting that "it was really cool to know that my ear was picking up a lot of the chord tones and I wasn't just playing random things" or that "you can make your solos sound a particular way (for ex, more dissonant) just by paying attention to the colors)." A few beginner music students expressed some uncertainty about how to evaluate themselves using the color-coding analysis. Providing more guidance on how to use the color-coding hand-in-hand with the soloing tips might have better supported these users.

The primary drawbacks that users expressed about the transcription controls were:

- Having a color-coded diagram instead of either a text legend or a separate page with a chord sheet (both of which were available instead of the color-coded diagram) would have been more useful for reminding users of the color associated with each chord tone.

- Requesting a different color-coding mode required re-generating the transcription each time. Saving previous versions of transcriptions would be faster.

Both issues would likely improve usability and would certainly merit attention in a future revision of the system. One frustration or point of confusion that a few participants expressed was that dissonant notes should have been colored red to stand out as "wrong." This feedback was interesting to hear because the dissonant notes were colored red in the initial version of the system, but we revised the coloring in version 2 to avoid presenting dissonant notes as inherently bad to play. However, we did not address the nuance of when or how to use dissonant notes in the educational sections of the application. Providing at least a brief mention of this idea could have been more educational to users. Another curious observation in the feedback was that certain users cared significantly more about familiarizing themselves with chord tones and learning what sounds "good" to play, while other users placed much more importance on learning what *not* to play. Exploring these different mindsets in a future revision of the system could lead to better customization of users' educational experiences.

		1	2	3	4	5	6	7	Mean	Std
Study 1	How easy to use are the recording and playback controls?	0	0	1	1	1	5	1	5.44	1.23
	How easy to use are the score controls?	0	3	1	1	0	4	0	4.11	1.90
Study 2	How easy to use are the recording and playback controls?	0	0	0	0	4	7	3	5.93	0.73
	How easy to use are the score controls?	0	0	0	5	2	2	5	5.50	1.34
	How easy to use are the transcription/analysis controls?	0	0	1	3	3	5	2	5.29	1.09

Figure 5-6: Comparison of study 1 vs study 2 summary statistics for users' evaluations of the application control components.

The recording and playback controls received slightly better feedback during the second user study. The average rating for these controls increased by 0.49, as shown in Figure 5-6. This component did not receive any major updates in version 2 of Practica. We only polished the user interface elements by adding tooltips and replacing more of the text in the controls with icons and simpler labels. The component design received good reviews. Users said that the straightforward layout was easy to learn. Users enjoyed that they could adjust the playback latency and record multiple takes. Some users found the latency adjustment difficult to use because they were unsure how much to adjust the latency in order to appropriately align their recording to the backing track. Creating a visual representation for the latency tool could help make this adjustment easier. Another useful feature that users requested was the ability to delete takes. A few users expressed confusion about the difference between the "pause" and "stop" functionalities.

The feedback for the score controls improved as well. The average rating for this component increased by 1.39 points (Figure 5-6). The usability ratings for this component were split between "middling difficulty" and "very easy to use." As in the previous study, users appreciated that they could easily shift note pitches. Users who participated in the previous study noted that the score controls had improved from version 1. The main improvements users mentioned were that the score tracks progress through the song well and that the score controls were "linked well with the recording/playback controls." Based on this feedback, we believe that the integration of the audio playback and score components in version 2 was a key part of the system's overall improvement. The four most frequent points of negative feedback were as follows:

- Users could not play back the score without the recording. Implementing this would require generating a MIDI version of the solo and passing that to the score embed, which could be investigated as a possible addition to a future version of the system.

- The transcription would fail on occasion, so the controls would not display a score. This issue is related to the server-side transcription generation and not the score controls.

- The score controls contained numerous features that users did not need and thought could be removed, although these features did not confound the interface. This issue is related to the score embed design that Flat uses. From our understanding, removing these extraneous features would require hiding all note or score editing options.

- The now bar would sometimes go out of sync with the audio playback. The current implementation of the system sends the start and end times of the solo audio to the score embed and uses these as endpoints for synchronization. The system could benefit from a redesigned version of this feature that records the time at which each measure starts in the generated transcription and sends those times to the client for use as synchronization points.

	1	2	3	4	5	6	7	Mean	Std
Would you be interested in using this app to practice improvisation in the future (assuming some technical improvements were made ?	0	0	0	1	4	1	8	6.14	1.06
If you do not have much experience with jazz improvisation, do you feel that you gained any insight into jazz improvisation by using this app?	0	0	0	0	4	2	5	6.09	0.9
After using this app, do you think you are more or less likely to explore improvisation further?	0	0	0	1	3	5	2	5.73	0.86
If you could have used this app while you were first learning to improvise, how helpful do you think it would have been for learning how to improvise?	0	0	0	0	1	2	2	6.20	0.75

Figure 5-7: Summary statistics for users' reflections on the educational aspect of the application.

After evaluating the system components, users were asked to reflect on the educational value of the system and answer four questions. Each question used a 1-to-7 rating with 1 representing the most negative response and 7 representing the most positive or affirmative response. Question 1 was mandatory for all participants. Questions 2 through 4 were optional and encouraged users to respond based on their experience level with improvisation. Figure 5-7 summarizes users' responses to each of these questions.

1. Would you be interested in using this app to practice improvisation in the future (assuming some technical improvements were made)?

   The average response to this question was a rating of 6.143 out of 7. Users did not provide additional feedback to supplement their answers to this question.

2. If you do not have much experience with jazz improvisation, do you feel that you gained any insight into jazz improvisation by using this app?

   Eleven of fourteen participants responded with an average rating of 6.091. When asked if they had additional feedback to supplement this answer, multiple users mentioned that they had been reluctant to try improvisation because they did not play jazz music and found it intimidating, but that the educational experience of the application was "just right to get my feet wet in jazz" and that trying to improvise using the application felt "super chill and low pressure, which is really nice."

3. If you do not have much experience with jazz improvisation, after using this app, do you think you are more or less likely to explore improvisation further?

   The same eleven participants responded with an average rating of 5.727. As Figure 5-7 demonstrates, all but one participant responded with scores of 6 or 7. The one participant who responded with less than a 5 for this question explained that "I don't play much jazz music in general, so I am not sure if I will explore improvisation further. But, that is just a personal preference and is not a reflection of the app in any way." When elaborating on their answers to this question, some users emphasized that having visual feedback on their solos through the color-coded transcriptions made them feel "like I was learning something quickly that normally would be very difficult to do without a well-developed ear and a lot of transcription practice." Users also mentioned that they enjoyed being more creative and learning different ways of playing by practicing improvisation with the application.

4. (For participants who had some improvisation experience before testing the application) If you could have used this app while you were first learning to improvise, how helpful do you think it would have been for learning how to improvise?

   Five participants answered this question with an average rating of 6.2. When given the option to elaborate on their answer, one user explained: "Being able to identify and see chord tones and extensions while hearing back my solos would've been a really sweet feedback loop to establish while building jazz vocabulary. E.g I could understand harmonically why I liked (or didn't like) the sound of particular phrases in my solos." Another user mentioned that while they did not have much experience learning about basic principles of improvisation before trying it out in a musical performance group, they thought that "this app would have surely closed those gaps."

	1	2	3	4	5	6	7	Mean	Std
How helpful do you think proposed feature 1 would be for learning about jazz improvisation?	0	0	0	0	2	4	8	6.43	0.73
How helpful do you think proposed feature 2 would be for learning about jazz improvisation?	0	0	0	1	2	4	7	6.21	0.94

Figure 5-8: Summary statistics for users' feedback on potential features to add in a future iteration of the application.

The final section of the post-test survey asked users to consider two potential features for future development: (1) the system would show transcriptions of solos by famous jazz musicians for the provided backing tracks and allow users to apply different color-coding analysis modes to transcriptions, and (2) the system would offer a quiz mode in which users would identify chord tones and tensions for each chord in the backing track. Figure 5-8 shows a summary of users' feedback on how helpful they thought each feature would be (on a scale of 1 to 7, where 1 means not helpful at all).

The first potential feature had an average interest rating of 6.429. Users mentioned that this feature would be even more useful if these transcriptions of famous solos included additional annotations as a learning resource. Multiple users noted that having this feature could give them more insight on how musicians use chord tones, tensions, and scale modes in their solos. Users also stated that this feature would encourage them to apply new techniques to their own solos.

The second potential feature had an average interest rating of 6.214. Multiple users thought this feature would be very helpful because it could solidify their chord recognition skills. Some users specifically mentioned that because they had trouble with recalling all the notes in each chord instead of reading sheet music with every note written, using a quiz feature to learn chord tones and tensions could help remedy this issue. A few users stated that while they thought this feature could be useful, they found it less interesting because other music education tools already provide this feature.

Chapter 6

Conclusion

When designing Practica, we set out to create a music education application that encourages students to explore jazz and musical improvisation in an interactive manner. We also sought to give students improvisation knowledge and performance feedback that they might receive from instructors or peers, but in a lower-pressure environment that supports students of all experience levels. From a technical perspective, improving the transcription implementation would be necessary for Practica to become a reliable music education tool. Adding more detailed educational content and creating more advanced levels of harmonic analysis would also help make Practica more effective. As it stands, the system provides students with a unique and accessible practice experience. Participants generally enjoyed testing Practica and appreciated that the application taught them about improvisation and harmony through casual experimentation.

6.1 Future Work

The basic system of Practica has been tested as a proof of concept. Future development could follow various paths in both the technical and educational domains.

6.1.1 Transcription Improvement

The current transcription system is relatively simple and could benefit from a more sophisticated implementation to improve note detection and quantization.

More Accurate Onset Detection

Both versions of Practica evaluated in this thesis used the same onset detection method. Table 5.7 shows the average precision, recall, and F1 score of the onset detection method. While the recall was reasonably high compared to the F1 score and precision, all three measures showed that our system would benefit from a more accurate onset detection method. The onset detection method tends to return background noise as extra onsets. Incorporating a noise reduction algorithm into our audio processing pipeline could improve the implementation's precision.

More Accurate Quantization

As mentioned in Section 5.2, the average pitch, quantization, and duration accuracy of the implementation improved from v1 to v2 (Table 5.10). While the reported accuracy measures in the second study are high compared to the previous values, we would still need to improve further. When we evaluated transcription accuracy, we allowed a sixteenth-note error threshold for onset and duration quantization. While other systems do not always generate perfect transcriptions, we would need to improve the transcription accuracy without this threshold to make Practica a more reliable transcription tool.

Additionally, as we explained in Section 4.2.1, the transcription implementation assumes that triplets and swung notes do not exist. The rhythm simplification method implemented in version 2 remedies some of the transcription errors caused by trying to quantize triplets or swing eighths as if they were straight-ahead. Because Practica is geared towards jazz improvisation, support for triplets and swung notes (perhaps even with different amounts of swing quantization, as some applications offer) would make a significant improvement to the transcription system [28].

6.1.2 Additional Features

During the second user study, we requested participants' feedback on potential features to add to a future version of Practica. We expand upon these features here and propose other additions as well.

Features Proposed in User Study 2

In user study 2, we asked participants what they thought of the following two features: (1) the system could show transcriptions of solos by famous jazz musicians for the provided backing tracks and allow users to apply different color-coding analysis modes to transcriptions, and (2) the system would offer a quiz mode in which users would identify chord tones and tensions for each chord in the backing track. Both were positively received (Figure 5-8).

To practice improvisation, many musicians recommend transcribing and analyzing solos. We thought that students with little transcription experience might find this too challenging, so we proposed the idea of providing the transcriptions and letting users apply the different color-coding analysis methods to learn more from some well-known solos. It could also be interesting to add a feature allowing users to try transcribing these solos themselves in the score editor and compare their results with the actual transcription. While we would likely focus first on implementing the originally-proposed version of this feature, adding the option to test users' transcription skills could provide a good challenge or practice method for more experienced students.

During the second user study, some participants mentioned that they struggled to remember chord tones or scales when recording solos. Adding a quiz mode for users to test their understanding of chord tones and their memory of different chord scales could help these users build a more solid foundation for improvisation.

Levels of Harmonic Analysis

Once students feel comfortable with the concept of chord tones and tensions (or if they start using Practica with this experience already), they may want the system to provide more detailed harmonic analysis. The system could teach students how to identify and play neighbor tones or passing tones in solos. With a more robust technical implementation, perhaps the system could also analyze musical phrases over a chord or set of chords instead of analyzing each note individually. The system could then provide more in-depth analysis of how longer phrases in students' solos related to the underlying chord progressions.

Integration with Jazz Lick Databases

Projects such as Dig That Lick, the Weimar Jazz Database, and BopLand provide databases of solos and jazz licks [29, 3, 30]. Users can search by note, melodic phrases, or chord progressions to find different licks. Integrating one of these databases into Practica could provide students with a wealth of information about how to solo over the supported backing tracks. A new feature could either (1) offer a selected set of licks for each backing track or (2) take user input to create a lick. The system could then show some examples of solos using that particular lick. Showing a solo using that lick would provide context of how to play this musical phrase in a solo at different points in a chord progression.

Appendix A

Analysis Mode Color-Coding

This appendix includes the mapping of scale degrees to colors for each analysis mode implemented in Practica. Table A.1 displays the color mappings side-by-side.

Chord tones: highlights the root, 3rd, 5th, and 7th of every chord.

Tensions: highlights the 9th, 11th, and 13th of every chord.

Dissonant notes: highlights notes if they are not in the list of scale degrees. Notes outside the chord are colored ■.

Dorian: highlights notes if they are one of the scale degrees for the Dorian mode. Notes outside the Dorian mode are colored ■.

Table A.1: Map of scale degrees to colors for each analysis mode implemented in Practica.

	Root	9th	3rd	11th	5th	13th	7th
Chord Tones	■	■	■	■	■	■	■
Tensions	■	■	■	■	■	■	■
Dissonant Notes	■	■	■	■	■	■	■
Dorian	■	■	■	■	■	■	■

Appendix B

Harmony and Soloing Help Sections

This appendix includes the help sections for each backing track. Each help section includes relevant harmonic information and soloing tips.

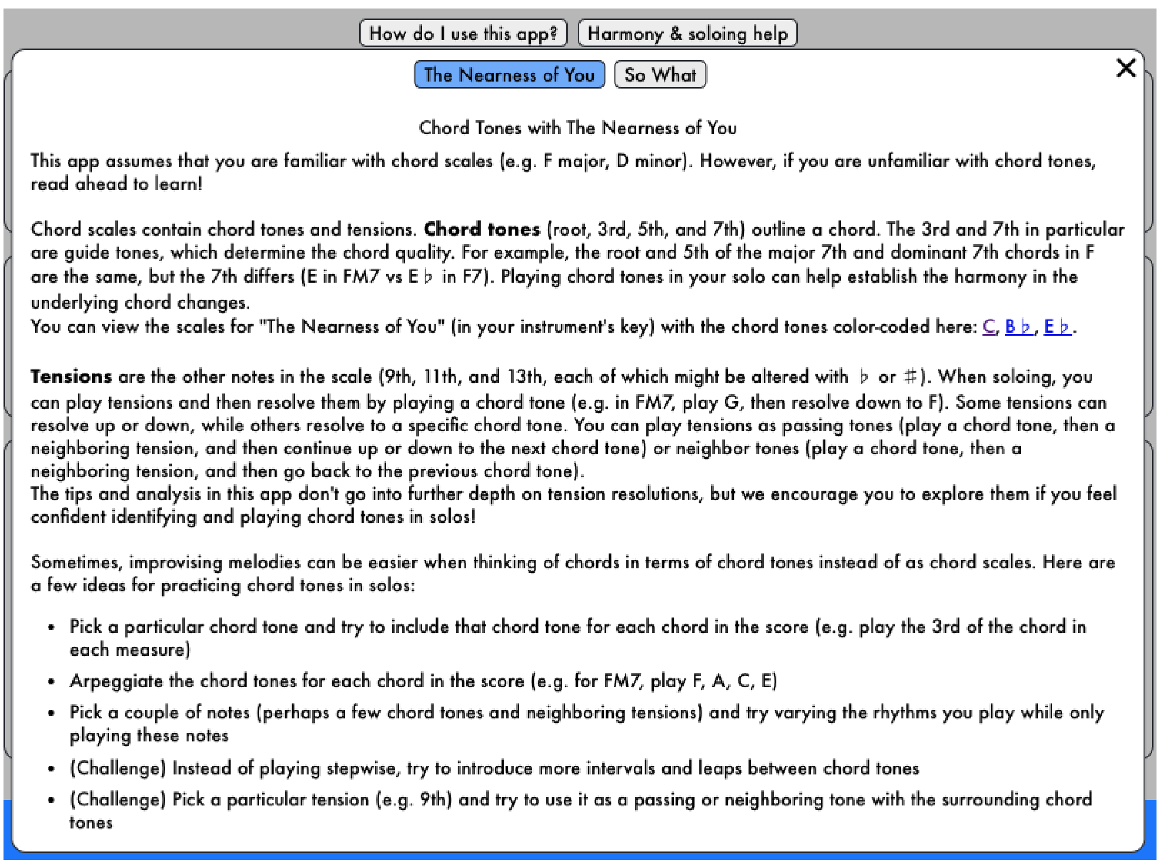


Figure B-1: Help section for "The Nearness Of You."

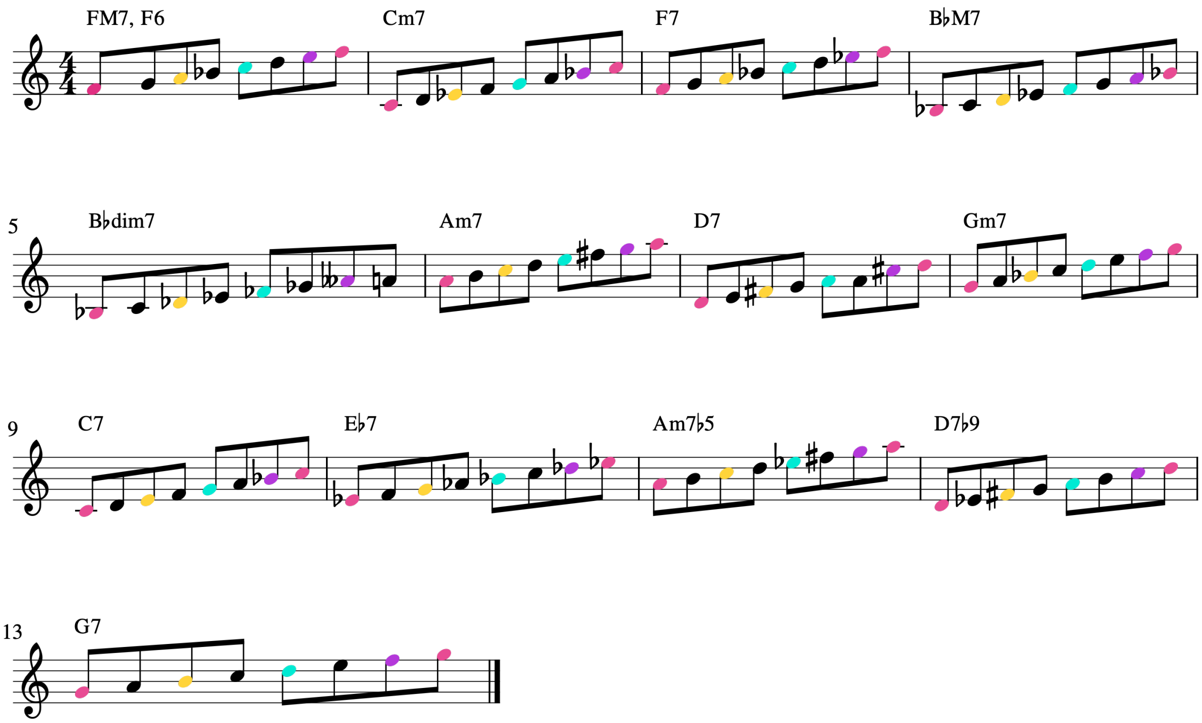


Figure B-2: (For C instruments) Scales for each chord in "The Nearness Of You" with chord tones highlighted. Help section provided scale sheets for C, B♭, and E♭ instruments.

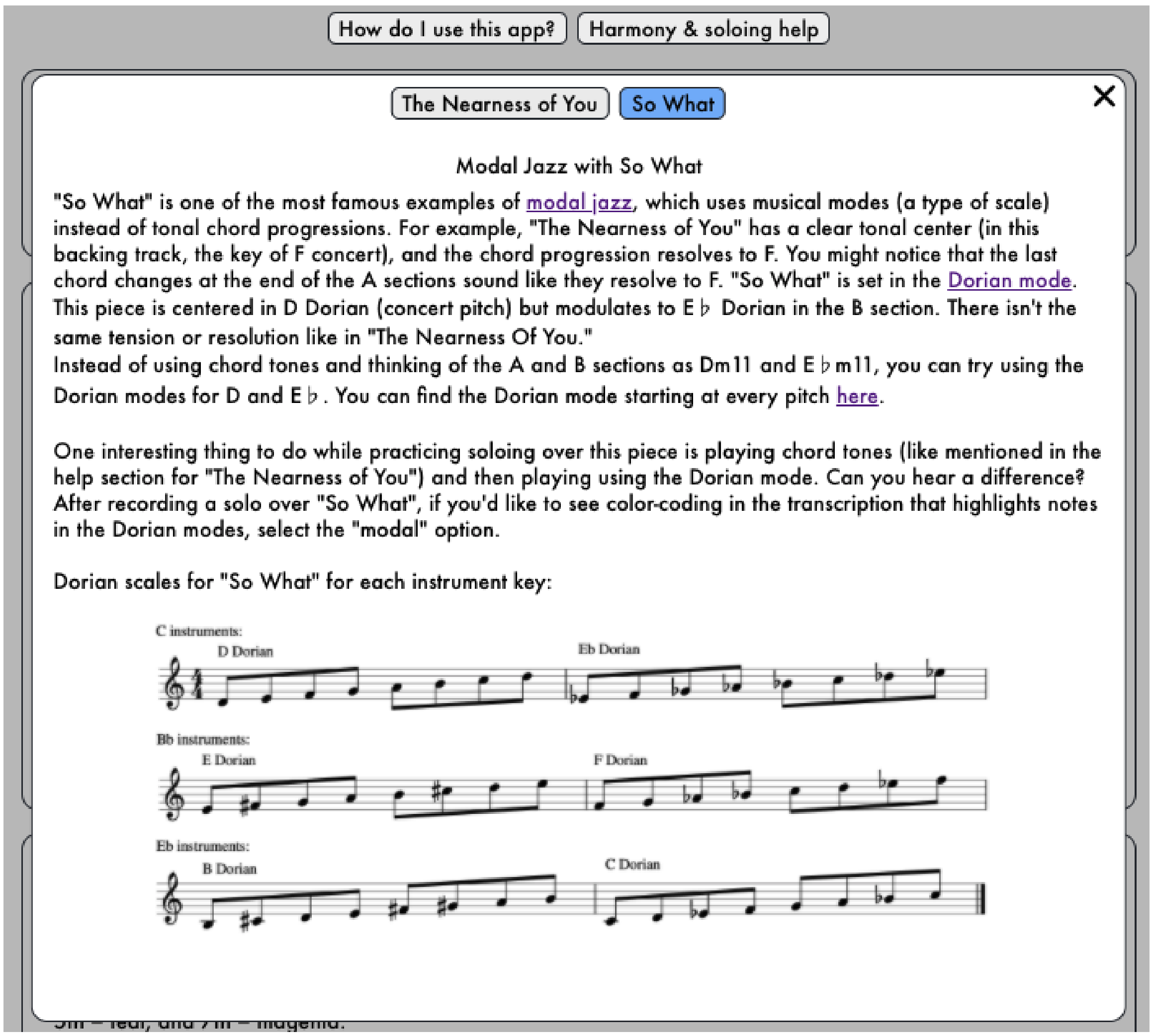


Figure B-3: Help section for "So What."

Appendix C

User Study Questions

This appendix includes the PDF versions of the surveys given to participants after completing each user study. The survey for study 1 starts on the following page.

practica user test 1

Link: https://practica-music.herokuapp.com

User instructions:
1. Make sure you are using headphones to listen to the audio so that when you record, the only audio recorded is from the notes you play.
2. Select "The Nearness of You."
3. Record a monophonic solo through at least the first eight measures. More is welcome, but please note that you will be asked to report how many notes (if unsure, you can provide an estimate) were incorrect in the system transcription.
4. Have the system transcribe the solo. If a note is red, it is considered outside the current chord.
5. Export this transcription. Rename the file to "{your kerberos}-system-test1" for MIT students or "{your name}-system-test-1" otherwise.
5. If you see any mistakes, you can change them using the score editor.
6. Export the solo transcription and rename the file to "{your kerberos}-user-test1".
7. Download the audio file for your solo take and rename it to "{your kerberos}-audio-test1".
8. Send both transcription files and the audio file to fiksin@mit.edu.

The survey contains four sections: background, evaluation of system controls, evaluation of transcription, and thoughts on future work.

* Required

1. Email *

____

Background

2. Which instruments do you play? *

____

3. Which instrument did you use for this test? *

____

4. For how many years have you played this instrument? *

Mark only one oval.

- ◯ <1 year
- ◯ 1-2 years
- ◯ 2-4 years
- ◯ 4-8 years
- ◯ 8+ years
- ◯ Other:

5. Please describe your musical education. *

Check all that apply.

- ☐ Private lessons
- ☐ Participation in a performance group
- ☐ Participation as a solo performer
- ☐ Music theory lessons
- ☐ Music performance lessons
- ☐ Self-taught performance
- ☐ Self-taught music theory

Other: ☐ ______________________________

6. What, if anything, is missing from your music education (e.g. improvisation tips, ear training)?

7. How much experience do you have with transcribing music? *

Mark only one oval.

	1	2	3	4	5	6	7	
No experience at all								I'm a pro

8. How much experience do you have with arranging or composing pieces? *

Mark only one oval.

	1	2	3	4	5	6	7	
No experience at all								I'm a pro

9. How much experience do you have with musical improvisation? *

Mark only one oval.

	1	2	3	4	5	6	7	
No experience at all								I'm a pro

10. Have you ever taken private lessons for any instrument? *

Mark only one oval.

- Yes
- No
- Other: ______

If no, please skip the next two questions.

11. For how many years have you taken private lessons?

Mark only one oval.

- <1 year
- 1-2 years
- 2-4 years
- 4-8 years
- 8+ years
- Other:

12. What kind of topics did these private lessons cover (e.g. playing technique, music theory, transcription)?

____

____

____

____

____

If you have never taken private lessons for an instrument, please answer the following questions.

13. What were your reasons for not taking private music lessons?

Check all that apply.

- [ ] Too expensive
- [ ] Not enough time
- [ ] Lack of interest
- [ ] Could not find an instructor

Other: [ ] ________________________________

14. If given the option, would you have taken private lessons for your instrument?

Mark only one oval.

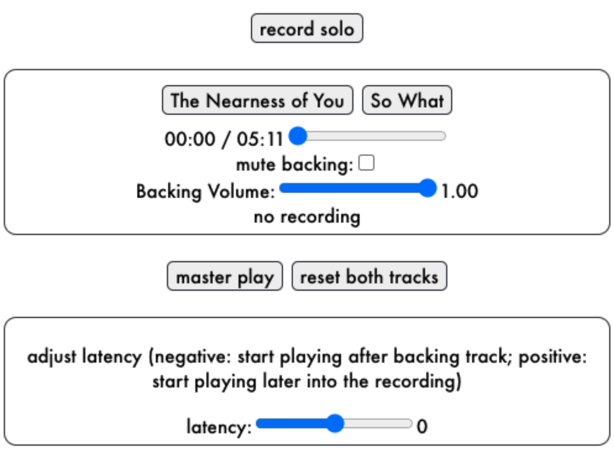
	1	2	3	4	5	6	7	
Definitely no	◯	◯	◯	◯	◯	◯	◯	Definitely yes

15. If you would like, you may explain the reasoning behind your response to the question above.

____

Evaluating system controls

Recording and playback controls:



16. How easy to use are the recording and playback controls? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very difficult	◯	◯	◯	◯	◯	◯	◯	Very easy

17. What worked well about the user interface for recording and playback? *

18. What was frustrating or difficult to use about the recording and playback UI, and why? *

19. How easy to use are the score controls? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very difficult	◯	◯	◯	◯	◯	◯	◯	Very easy

20. What worked well about the user interface for the score? *

21. What was frustrating or difficult to use about the score UI, and why? *

Evaluating transcription

This section asks questions regarding the performance of the system transcription.

Note: if the transcription button stays greyed out for more than thirty seconds or so, the system has likely crashed (oops!). Please download your solo take, restart the system (just refresh the page), and make a note of this occurrence later in the survey. Thanks!

22. How many notes did you play in your solo? *

23. How many pitches were incorrect in the transcription? *

24. How many notes or rests had incorrect durations in the transcription? *

25. How many notes were missing? *

26. How accurate do you think the system was in transcribing rhythms? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very inaccurate	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very accurate

27. What kind of audio input device (microphone) did you use? *

28. What kind of audio output device (speaker) did you use? *

29. How useful was the note color-coding for identifying notes outside the chord? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very useful

30. Would you prefer this information conveyed in a written list that the system would show next to your transcription, or you would you prefer the current color-coding method? *

Mark only one oval.

	1	2	3	4	5	6	7	
Would prefer the system giving me a written list next to the transcription	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Would prefer the system color-coding notes in the tr

Future work

This section addresses potential new features for the system.

Proposed feature 1: the system offers you the choice of transcribing your recorded solo yourself in the score editor. Once you finish, the system provides feedback on the accuracy of your transcription.

Proposed feature 2: the system shows you annotations about how the notes you played fit into the chord progressions of the piece (e.g. this note is a leading tone; this chord is part of a ii-V-I).

31. How useful do you think proposed feature 1 would be? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very useful

32. If the accuracy of the system transcription improved, how useful do you think proposed feature 1 would be? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	◯	◯	◯	◯	◯	◯	◯	Very useful

33. Please explain your reasoning behind your previous two responses about proposed feature 1.

34. How useful do you think proposed feature 2 would be? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	◯	◯	◯	◯	◯	◯	◯	Very useful

35. Please explain your reasoning behind your previous response about proposed feature 2.

36. Would you prefer to see any analysis/annotations directly in the score (color-coding? text?) or in a separate panel? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	◯	◯	◯	◯	◯	◯	◯	Very useful

37. If you would like to elaborate on your previous answer, please do so here:

38. If you were learning how to improvise, what would you like to see from a practice tool to help you learn (e.g. tips on things to try playing, information about the chord changes, etc)?

39. If you already feel comfortable with improvisation, what (if anything in particular) helped you learn how to improvise?

practica user test 2

Link: https://practica-music.herokuapp.com

Make sure you are using headphones to listen to the audio so that when you record, the only audio recorded is from the notes you play. Wired headphones are preferable to wireless. Solos must be monophonic.

~~~~~~~~~~~~~~~

User instructions

First, click on "How do I use this app?" and read through the system instructions to familiarize yourself with the controls.

Phase 1: "The Nearness of You"

1. Select “The Nearness of You.”
2. Record a monophonic solo through at least the first eight measures and adjust the latency value to line up the audio of the recording and backing tracks.
3. Have the system transcribe the solo. Test at least one of the three available modes (chord tones, tensions, and dissonant notes) and explore the color-coding results.
4. Export this transcription. Rename the file to "{your name}-nearness-v1". Download the audio file for this solo take and rename it to "{your name}-nearness-v1".
5. Click on "Harmony & soloing help" and read the help section for "The Nearness of You."
6. Using at least one of the soloing tips, record another solo.
7. Have the system transcribe the solo. Test all three of the available modes (chord tones, tensions, and dissonant notes) and explore the color-coding results. To see the results of different color-coding on your current solo, select a different mode and press the transcription button again.
8. Export this transcription (can use any of the available analysis modes). Rename the file to "{your name}-nearness-v2". Download the audio file for this solo take and rename it to "{your name}-nearness-v2".

Phase 2: "So What"

1. Select “So What.”
2. Record a monophonic solo through at least the first eight measures and adjust the latency value to line up the audio of the recording and backing tracks.
3. Have the system transcribe the solo. Test the modal analysis mode and explore the color-coding results.
4. Export this transcription. Rename the file to "{your name}-sowhat-v1". Download the audio file for this solo take and rename it to "{your name}-sowhat-v1".
5. Click on "Harmony & soloing help" and read the help section for "So What."
6. After reading the help section for this track, record another solo.
7. Have the system transcribe the solo. Test both available modes (chord tones and modal) and explore the color-coding results. To see the results of different color-coding on your current solo, select a different mode and press the transcription button again.
8. Export this transcription (can use any of the available analysis modes). Rename the file to "{your name}-sowhat-v2". Download the audio file for this solo take and rename it to "{your name}-sowhat-v2".

Wrap-Up

1. Complete this survey.
2. Send all of your exported transcription files and audio files to fiksin@mit.edu.
3. You will receive a $10 Amazon gift card as a thank you for contributing to my thesis!
~~~~~~~~~~~~~~~

The survey contains six sections: background, soloing help and harmonic analysis, transcription performance, system controls, reflection, and thoughts on future work.

* Required

1. Email *

Background

2. Which instrument did you use for this test? *

3. Did you participate in user test 1? If so, you may skip the background questions below. If not, please answer the following questions. *

Mark only one oval.

- ◯ Yes
- ◯ No

4. Which instruments do you play?

5. For how many years have you played this instrument?

Mark only one oval.

- ◯ <1 year
- ◯ 1-2 years
- ◯ 2-4 years
- ◯ 4-8 years
- ◯ 8+ years
- ◯ Other:

6. Please describe your musical education.

Check all that apply.

- [ ] Private lessons
- [ ] Participation in a performance group
- [ ] Participation as a solo performer
- [ ] Music theory lessons
- [ ] Music performance lessons
- [ ] Self-taught performance
- [ ] Self-taught music theory

Other: [ ] ____________________

7. What, if anything, is missing from your music education (e.g. improvisation tips, ear training)?

8. How much experience do you have with transcribing music?

Mark only one oval.

	1	2	3	4	5	6	7	
No experience at all	( )	( )	( )	( )	( )	( )	( )	I'm a pro

9. How much experience do you have with arranging or composing pieces?

Mark only one oval.

	1	2	3	4	5	6	7	
No experience at all	( )	( )	( )	( )	( )	( )	( )	I'm a pro

10. How much experience do you have with musical improvisation?

Mark only one oval.

	1	2	3	4	5	6	7	
No experience at all	◯	◯	◯	◯	◯	◯	◯	I'm a pro

11. Have you ever taken private lessons for any instrument?

Mark only one oval.

◯ Yes
◯ No
◯ Other: ______________________

If no, please skip the next two questions.

12. For how many years have you taken private lessons?

Mark only one oval.

◯ <1 year
◯ 1-2 years
◯ 2-4 years
◯ 4-8 years
◯ 8+ years
◯ Other:

13. What kind of topics did these private lessons cover (e.g. playing technique, music theory, transcription)?

____
____
____
____
____

If you have never taken private lessons for an instrument, please answer the following questions.

14. What were your reasons for not taking private music lessons?

Check all that apply.

- [ ] Too expensive
- [ ] Not enough time
- [ ] Lack of interest
- [ ] Could not find an instructor

Other: [ ] ____________________

15. If given the option, would you have taken private lessons for your instrument?

Mark only one oval.

	1	2	3	4	5	6	7	
Definitely no	( )	( )	( )	( )	( )	( )	( )	Definitely yes

16. If you would like, you may explain the reasoning behind your response to the question above.

____

Evaluating soloing help and harmonic analysis

Help controls:

How do I use this app? Harmony & soloing help

17. How helpful was the information in the "How do I use this app?" section? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not helpful at all	◯	◯	◯	◯	◯	◯	◯	Very helpful

18. How helpful was the "Chord Tones with The Nearness of You" section (under "Harmony & soloing help")? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not helpful at all	◯	◯	◯	◯	◯	◯	◯	Very helpful

19. How helpful was the "Modal Jazz with So What" section (under "Harmony & soloing help")? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not helpful at all	◯	◯	◯	◯	◯	◯	◯	Very helpful

20. What did you like about the harmony and soloing help sections? What did you find useful? *

____

____

____

____

____

21. What would you have changed or added to the harmony/soloing help sections? *

____

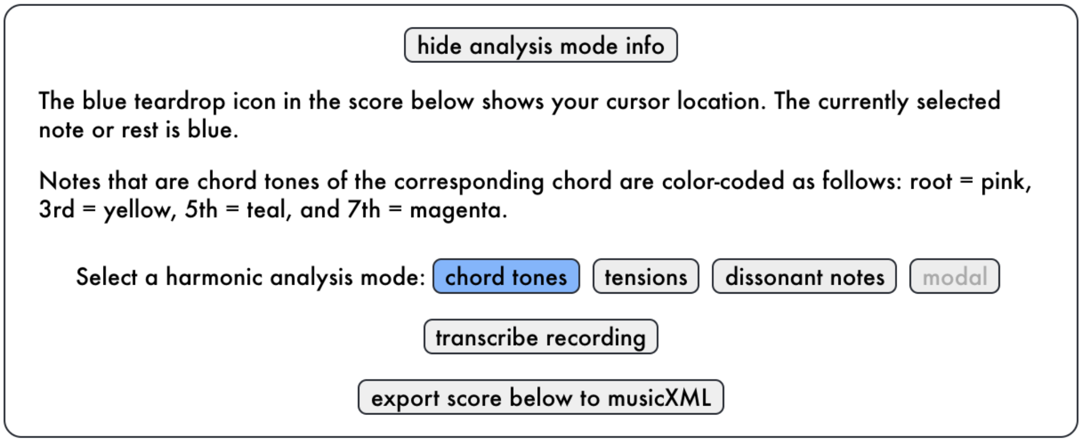
____

____

____

____

Transcription and analysis controls:



22. How easy to use are the transcription/analysis controls? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very difficult	◯	◯	◯	◯	◯	◯	◯	Very easy

23. How helpful were the color-coded results from transcription using different analysis modes? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not helpful at all	◯	◯	◯	◯	◯	◯	◯	Very helpful

24. What worked well about the analysis controls and different color-coding modes? What did you like about them? *

25. Did you find any color-coding analysis modes particularly useful? If so, why? *

26. What was frustrating or difficult to use about the analysis controls, and why? *

Evaluating transcription performance

This section asks questions regarding the performance of the system transcription. For quantitative questions, please pick whichever solo you think was most accurate for this question.

Note: if the transcription button stays greyed out for more than thirty seconds or so, the system has likely crashed (oops!). Please download your solo take (you don't need to redo it, just make sure to send it in the email), restart the system (just refresh the page), and make a note of this occurrence later in the survey. Thanks!

If you are unsure as to how many notes you played or how many pitches/rhythms were incorrect, please note this and give a ballpark amount (or say "I don't know" and I'll go back and check the results).

27. How many notes did you play in your solo?

28. How many pitches were incorrect in the transcription?

29. How many notes or rests had incorrect durations in the transcription?

30. How many notes were missing?

31. How accurate do you think the system was in transcribing rhythms? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very inaccurate	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very accurate

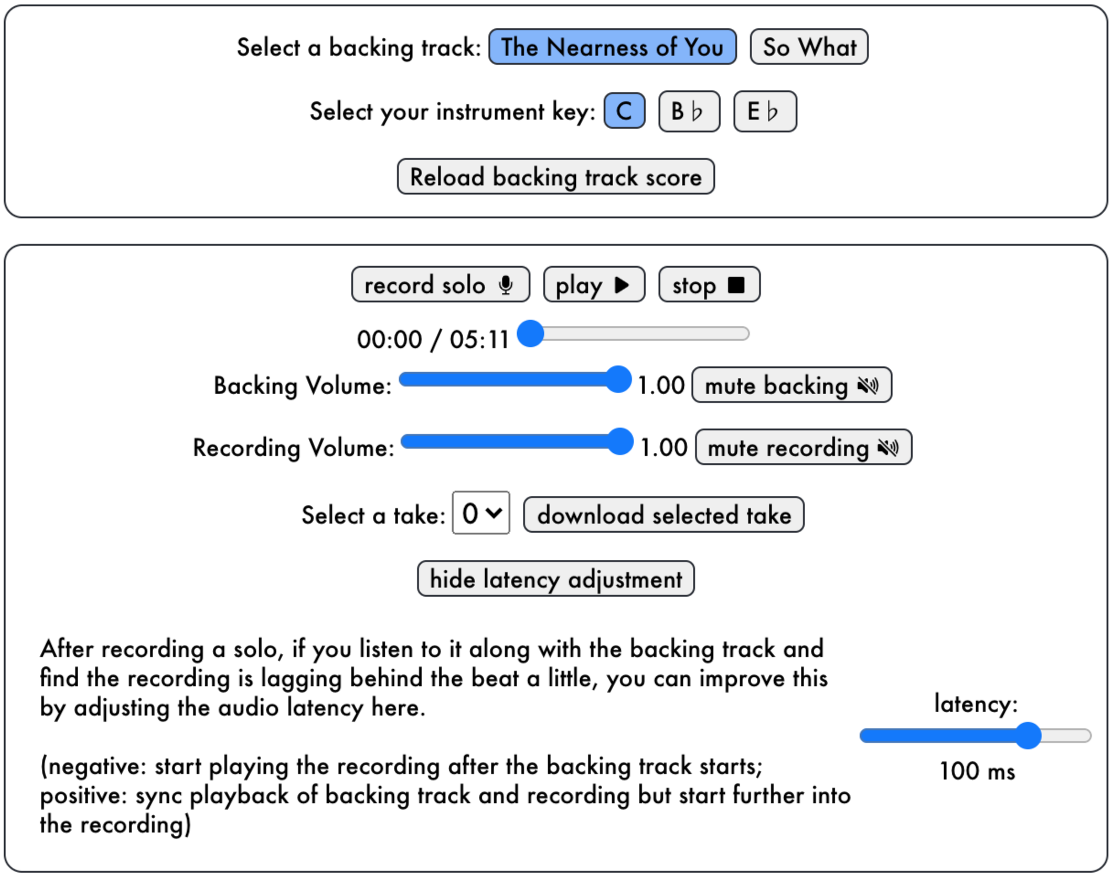
32. What kind of audio input device (microphone) did you use? *

33. What kind of audio output device (speaker) did you use? *

34. Which browser did you use? *

Evaluating system controls

Recording and playback controls:



35. How easy to use are the recording and playback controls? *

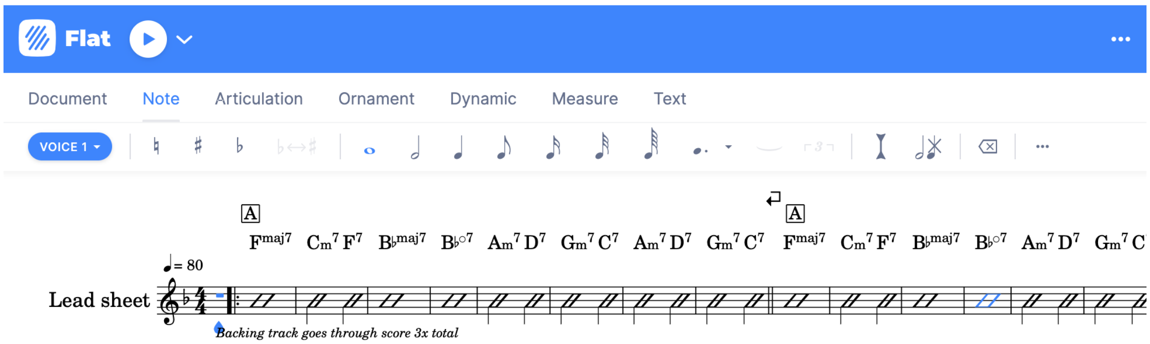
Mark only one oval.

	1	2	3	4	5	6	7	
Very difficult	◯	◯	◯	◯	◯	◯	◯	Very easy

36. What worked well about the user interface for recording and playback? *

37. What was frustrating or difficult to use about the recording and playback UI, and why? *

Score controls:



38. How easy to use are the score controls? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very difficult	◯	◯	◯	◯	◯	◯	◯	Very easy

39. What worked well about the score controls? *

40. What was frustrating or difficult to use about the score controls, and why? *

Reflection

41. Would you be interested in using this app to practice improvisation in the future (assuming some technical improvements were made ? *

Mark only one oval.

	1	2	3	4	5	6	7	
Definitely no	◯	◯	◯	◯	◯	◯	◯	Definitely yes

If you do not have much experience with jazz improvisation, please answer the next two scale questions (and the subsequent written question, if you are so inclined).

42. If you do not have much experience with jazz improvisation, do you feel that you gained any insight into jazz improvisation by using this app?

Mark only one oval.

	1	2	3	4	5	6	7	
Definitely no	◯	◯	◯	◯	◯	◯	◯	Definitely yes

43. After using this app, do you think you are more or less likely to explore improvisation further?

Mark only one oval.

	1	2	3	4	5	6	7	
Will definitely not explore further	◯	◯	◯	◯	◯	◯	◯	Will definitely explore further

44. You may elaborate on your above answers here if you would like to do so.

If you already had some improvisation experience before testing this app, please answer the following scale question (and the subsequent written question, if you are so inclined).

45. If you could have used this app while you were first learning to improvise, how helpful do you think it would have been for learning how to improvise?

Mark only one oval.

	1	2	3	4	5	6	7	
Not helpful at all	◯	◯	◯	◯	◯	◯	◯	Very helpful

46. You may elaborate on your above answer here if you would like to do so.

Future work

This section addresses potential new features for the system.

Proposed feature 1: the system shows you transcriptions of solos by famous jazz musicians for the provided backing tracks and lets you apply the different color-coding analysis modes to the transcriptions.

47. How helpful do you think proposed feature 1 would be for learning about jazz improvisation? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	◯	◯	◯	◯	◯	◯	◯	Very useful

48. Please explain your reasoning behind your previous response about proposed feature 1.

Proposed feature 2: the system offers a quiz mode in which you identify chord tones and tensions for each chord in the backing track.

49. How helpful do you think proposed feature 2 would be for learning about jazz improvisation? *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	◯	◯	◯	◯	◯	◯	◯	Very useful

50. Please explain your reasoning behind your previous response about proposed feature 2.

51. What other information or features do you think would make this app more effective in helping people learn how to improvise? *

This content is neither created nor endorsed by Google.

Google Forms

Bibliography

[1] Shunit Haviv Hakimi, Nadav Bhonker, and Ran El-Yaniv. Bebopnet: Deep neural models for personalized jazz improvisations. In *Proceedings of the 21st International Society for Music Information Retrieval Conference, ISMIR 2020*, 2020.

[2] Ji-Sung Kim. deepjazz. `https://deepjazz.io/`, 2019.

[3] Martin Pfleiderer, Klaus Frieler, Jakob Abeßer, Wolf-Georg Zaddach, and Benjamin Burkhart, editors. *Inside the Jazzomat - New Perspectives for Jazz Research*. Schott Campus, 2017.

[4] EarMaster. `https://www.earmaster.com/`, 2021.

[5] Harmony Cloud. `https://www.stefonharris.com/harmony-cloud`, 2015.

[6] ScoreCloud. `https://scorecloud.com/`, 2018.

[7] AnthemScore. `https://www.lunaverus.com/`, 2021.

[8] Eita Nakamura, Emmanouil Benetos, Kazuyoshi Yoshii, and Simon Dixon. Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 101–105, 2018.

[9] Keith D. Martin. Automatic transcription of simple polyphonic music: . . ., 1996.

[10] Impro-Visor. `https://www.cs.hmc.edu/~keller/jazz/improvisor/`, 2019.

[11] Technimo. iReal Pro. `https://irealpro.com/`, 2021.

[12] mDecks Music. Mapping Tonal Harmony Pro. `https://mdecks.com/mapharmony.phtml`, 2020.

[13] Flat. `https://flat.io/`, 2020.

[14] Hoagy Carmichael, Ned Washington. The nearness of you, 1938.

[15] Miles Davis. So what, 1959.

[16] Guitare Improvisation Martin Gioani. “so what (140 bpm): Jazz backing track.” july 27, 2017. [YouTube video].

[17] Roger B Dannenberg. Music performance over networks and the latency problem, Jul 2020.

[18] Francis Rumsey. Desktop Audio Technology: Digital Audio and MIDI Principles, 2004.

[19] reacthowler. `https://github.com/thangngoc89/react-howler`, 2021.

[20] MediaStream Recording API. `https://developer.mozilla.org/en-US/docs/Web/API/MediaStream`, 2021.

[21] lamejs. `https://github.com/zhuker/lamejs`, 2020.

[22] Sonic Visualizer. `https://www.sonicvisualiser.org/`, 2021.

[23] Brian McFee, Alexandros Metsai, Matt McVicar, Stefan Balke, Carl Thomé, Colin Raffel, Frank Zalkow, Ayoub Malek, Dana, Kyungyun Lee, Oriol Nieto, Dan Ellis, Jack Mason, Eric Battenberg, Scott Seyfarth, Ryuichi Yamamoto, viktorandreevichmorozov, Keunwoo Choi, Josh Moore, Rachel Bittner, Shunsuke Hidaka, Ziyao Wei, nullmightybofo, Darío Hereñú, Fabian-Robert Stöter, Pius Friesch, Adam Weiss, Matt Vollrath, Taewoon Kim, and Thassilo. librosa/librosa: 0.8.1rc2, May 2021.

[24] Paul Brossier, Tintamar, Eduard Müller, Nils Philippsen, Tres Seaver, Hannes Fritz, cyclopsian, Sam Alexander, Jon Williams, James Cowgill, and Ancor Cruz. aubio/aubio: 0.4.9, February 2019.

[25] Alain de Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[26] Colin Raffel, Brian Mcfee, Eric Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel Ellis. `mir_eval`: A transparent implementation of common mir metrics. 10 2014.

[27] Sebastian Böck, Florian Krebs, and Markus Schedl. Evaluating the online capabilities of onset detection methods. 10 2012.

[28] Logic Pro X. `https://www.apple.com/logic-pro/`, 2021.

[29] Frank Höger, K. Frieler, and Martin Pfleiderer. Dig that lick: Exploring melodic patterns in jazz improvisation. 2019.

[30] The BopLand Project. Bopland. `https://www.bopland.org/`, 2018.