A Technological Framework to Teach Music Online via Machine Learning with the Focus on Automated Chord Detection

by

Fatemeh Jamshidi

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 11, 2021

Keywords: Human-machine Music Ensemble, Multimodal Interaction, Machine Learning, Score Following, Live Concert System, Music Education, On-line Learning

Approved by

Richard Chapman, Chair, Professor of Computer Science and Software Engineering
Daniela Marghitu, Co-chair, Professor of Computer Science and Software Engineering
Nancy Barry, Professor of Music Education

Abstract

This thesis proposes possible approaches to adding technology to music courses and examines the impact online music courses may have on music teaching and practicing. The main question in this research is:
How can we build an artificial system that, improves its ability to sense and coordinate with human musician as they are learning and practicing music?

Artificial Intelligence and Human-Computer Interaction have enhanced computer music systems' capability to perform with humans through a broad spectrum of applications. However, musical interaction between humans and current applications is still less musical than the interaction between actual humans. This thesis incorporates various techniques, especially machine learning and deep learning algorithms, to make the experience of learning and practicing music more intuitive and efficient for music lovers. The current system covers three fundamental aspects of human-computer collaborative music performance and practice: 1) music theory curriculum, 2) chord detection, and 3) score following.

We developed a Web-based prototype to teach people with little to no music background the music theory basics and how to play the piano. To address this problem, we developed our prototype in two phases. The first one is for students to learn about the basics of music theory. Secondly, students will be able to practice what they learned by playing their favorite songs on the application which connects to their own digital or acoustic piano and provides proper guidance.

We implemented a model that trains a different set of parameters based on each individual measure and focuses on predicting the number of chords and notes per chord. A nearest-neighbor search algorithm will decode an improvised score to select the training example closest to the estimation given the model prediction.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisors, Professor Richard Chapman and Dr. Daniela Marghitu, for their continuous support of my Master's study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Master's study.

Besides my Computer Science and Software Engineering advisors, I would like to thank Dr. Nancy Barry for her encouragement, insightful comments, and challenging questions, making this research more meaningful.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Musical Human-Computer Interaction (HCI) techniques have empowered computer music systems to perform with humans via a broad spectrum of applications [2]. It is vital to have a consistent and focused approach when moving from traditional face-to-face (F2F) courses to Remote Learning (RL) courses using an online music teaching environment. Music students and teachers use different websites, online apps, and computer programs to learn, remix, and compose music. Existing music technologies allow digital or MIDI-enabled acoustic pianos to connect synchronously over the Internet, producing reliable instrumental audio, separate from the video-conferencing platform. Research is required to help teachers transition to the online format.

As RL music courses become more common, graduate teaching assistants, or tutors, will become essential as instructor support mechanisms. We should train online teachers to have the necessary skills of online-music-teaching, communication, and assessment.

Based on [3], I revealed four essential elements for online music courses:

1. Online music pedagogy (e.g., teaching philosophies, authentic music learning, openness to online music learning, institutional support, and learning approaches)
2. Course design (e.g., planning, organization, multimedia use, and curriculum)
3. Assessment (e.g., meaningful opportunities to demonstrate music learning)
4. Communication (e.g., methods for exploring subject content and technology tools)

This research identifies critical elements for developing a program for online music teaching and practicing. The goal is to train music teachers to master online skills and provide an online platform for people to practice music individually or as a group. Differences between F2F courses and online platforms include multimedia technology, social constructivist[1] learning activities and developing practical online communication skills.

This research has two main questions:

1. What are the key components to teaching online music skills to music teachers?

2. How can these components be adapted to implement an automated online tool that can train itself based on students' skills and learning styles?

We are developing a teaching framework that helps music faculty members transition from traditional F2F classroom teaching to online. This teaching framework is divided into three phases: 1) hybrid online courses; 2) a fully online study focused on social constructivist learning and; 3) fully online classes with student interactions.

In this research, we created a music technology curriculum for middle school students, assessed by expert music teachers. After analyzing the results provided by music teachers, we changed the music modules accordingly. The second phase is to record different F2F classes in which music teachers teach our curriculum to students. In phase three, we implement an automatic online teaching model that can train itself based on the rehearsals and face-to-face recorded courses provided by music instructors. The system incorporates techniques from different realms, including real-time music tracking (score following), beat estimation, chord detection, and body movement generation. In our system, the virtual music teachers' and students' behavior is captured based on the given music audio alone, and such an approach results in a low-cost, efficient and scalable way to produce human and virtual musicians' co-performance [4].

This thesis presents various techniques, especially Machine Learning (ML) algorithms, to create Artificial Intelligence (AI) tutors and musicians that perform with humans. We focus on

[1]Social constructivism: Social constructivism teaches that all knowledge develops as a result of social interaction and language use, and is, therefore, a shared, rather than an individual, experience. Knowledge is additionally not a result of observing the world; it results from many social processes and interactions.

four aspects of expression in human-computer collaborative performance: 1) Chord and pitch detection, 2) timing and dynamics, 3) basic improvisation techniques, and 4) facial and body gestures.

Two of the most fundamental aspects of online music teaching are timing and dynamics. We create a model of different teachers performing as co-evolving time series. ”Based on this representation, we develop a set of algorithms, to discover regularities of expressive musical interaction from rehearsals” [2]. Providing the learned model, an artificial performer generates its musical expression by interacting with a human performer, given a predefined curriculum. The results show that ML can create a more expressive and human-like collaborative performance with a small number of rehearsals than the baseline automatic accompaniment algorithm.

Body and facial movements are also essential aspects of online music teaching. We study body and facial expressions using the feature extraction models to create features based on teacher recordings. We contribute the first algorithm to enable our virtual teaching model to perform an accompaniment for a musician and react to human performance with gestural and facial expression. The current system uses rule-based performance-motion mapping and separates virtual tutor motions into three groups: finger motions, body movements, and eyebrow movements. Our result shows that the virtual tutor embodiment and expression enable more musical, interactive, and engaging human-computer collaborative performance [2].

1.1 Literature Context

Internet-based learning environments, called online learning, are becoming more practical in modern education and describe courses that have over 80 percent of instructions online [5]. Post-secondary online courses have steadily increased as institutions recognize the financial savings associated with e-learning over the provision of face-to-face courses [**?**].

Music courses in a Bachelor of Music program contain highly specialized instruction to assist students in attaining professional-level musical abilities. This instruction requires a master musician-instructor for all music courses in addition to one-to-one instruction in private applied lessons. However, the traditional one-to-one model of teaching is geographically and financially limiting due to students needing to relocate to the same place as the instructor.

While many scholars and researchers have highlighted the need for change in music education, addressing the limitations imposed by geographic and financial challenges in post-secondary music requires adjustment to new pedagogy. This new pedagogy involves a paradigmatic shift for instructors, students, policymakers, and new teaching methods and ideas.

Technology may assist in reforming post-secondary music courses as a shift in pedagogy transitions face-to-face instruction toward the inclusion of online learning. Current music education research [6]suggests that online learning technologies can assist by providing a means of engagement that will encourage the necessary reforms in the music education experience.

Developing a framework for online music courses requires integrating the educational fields of design and technology within pedagogy. In other words, the online musical framework had to bring together online instructional course design, communication technologies, and pedagogy informed by appropriate research in the fields of online learning, music, and machine learning. This intersection of creative learning technology and teaching music research reveals how complex online music education may be if artistic individuality and the practice of learning music are also included.

1.2 Purpose of the Study

The purpose of this research was to identify the essential elements of current online learning approaches used in academic music courses and to critically consider their application in the development of an online pedagogical framework for online music courses. This research also aims to implement an online open-source platform for musicians and music lovers to practice their favorite songs based on each individual's skills and learning levels.

Multiple pilot studies investigated in this study consisted of a two-phase method of data collection and analysis. In the first phase, middle school students from Alabama and surrounding were invited to participate in the study. Case study one had 24 total participants, case two had 15 total participants.

Phase one provided student participants with an opportunity to identify their experiences in online music learning through the Community of Inquiry (CoI) survey instrument. These surveys sought to identify the effectiveness of online practices and tools of cognitive-, social-, and teaching-presences. Students further described specific examples of teaching and learning practices through individual semi-structured interviews. Phase two consisted of focus group discussions in the format of two graduate-level music education courses where participants were music teachers. Participants were asked to explore an online teaching framework developed from the literature and data analyzed from the previously completed phase one.

As posited initially, the overall data helped various articulate components that made up online music courses and highlighted the challenges. These findings are presented in a formal discussion. Implications for the further study of online music courses are presented as well.

1.3 Research Questions

The specific research questions were:

1. What are the essential elements of current online music courses at the middle and high school levels?

2. From the Community of Inquiry framework, what identifiable elements influence instructional design and facilitation in online music courses?

3. Using the elements identified from the data, how do these elements inform a pedagogical framework for developing online music courses?

1.4 Research Statement

With computational techniques, especially advanced machine learning algorithms, we can create an intelligent artificial performer that can understand and respond to human musical expression.

There are many issues to be addressed. Questions regarding the data and feature representations include:

- How should we extract useful features to represent a musical expression of different scales of music structure? E.g., how to represent a crescendo (becoming louder) phrase and represent a ritardando (slowing down) measure?

- How should we choose the level of abstraction to generate artificial performance? E.g., should we decode the artificial performance note-by-note, chord-by-chord, or measure-by-measure?
- What are the dominant aspects of music performance that affect the expressive interaction? E.g., is expressive timing affected more by rhythm or by pitch contour?

Questions regarding the learning task include:

- How can we design machine-learning algorithms to distill models from rehearsals? In other words, how can we learn regularity from seemingly irregular data?
- What are the limits of validity of the learned models? E.g., which model generalizes across a whole piece or even across different pieces of music, and which model only applies to some specific score locations?
- How does performance/performers style affect the learning? E.g., would the model train on similar performances or the same performers lead to better results?
- How many rehearsals and actual music courses are needed to train the artificial performer, and how does the number of rehearsals affect the learning? E.g., would the minimum number of required rehearsals be reasonable in practice, and would adding more rehearsals lead to better results?

Questions regarding the evaluation scheme include:

- How should we design objective and subjective measurements to validate the generated artificial performance?
- How much better is the generated performance compared with our baseline, and how far/close is it from a human performance?
- Is the evaluation of the artificial performance consistent under different measurements and criteria?

1.5 Thesis Overview

This thesis uses machine-learning algorithms to create expressive chord detection deviations and basic improvisation techniques for music curriculum design to address the questions above. In addition, the research evaluates the performance of students using the curriculum designed in this research and identifies the fundamental aspects of an online music framework to learn and practice music. The following list, show the organization and an overview of the rest of the thesis:

- Chapter 2 reviews related literature in three research fields: Computer Music, Machine Learning Statistics, Online Music Applications.
- Chapter 3 presents the first case study with middle school and high school students.
- Chapter 4 presents the different training strategies under realistic constraints of data collection. We show the results of music teachers' evaluation of the design music curriculum.
- Chapter 5 describes the data collection and pre-processing techniques. It also presents the feature representation for piano chords, i.e., the intermediate layer between the data and the computational models. We contribute a general feature-extraction scheme for music chords. We also contribute a method that derives the onset and dynamics features of a chord based on an actual performance where notes do not begin simultaneously or have precisely the same dynamics.
- Chapter 6 presents the conclusion and future work.

Chapter 2

Literature Review

2.1 Computer Music

Computers nowadays have become part of everyday life; however, interest in programming remains relatively low, especially in underrepresented groups. Integrating visual and interactive art and music with Computer Science (CS) is an approachable domain, and music production relies on several essential concepts in computer science [7][8][9]. The MediaCom course at Georgia Tech aims to engage women in computing practices by manipulating different media forms (images and sounds) with code [10][11][12].

Other attempts have been made to reinforce programming concepts through music education to enhance recruitment and retention in computing. In a four-year curriculum study on material linking CS and digital audio, students implemented a boost to their understanding of the science of sound [13]. It was also found that students benefited most from practical projects they felt were relevant to themselves and others. Another project called Musicomputation, an advanced computer music/science course, ”found success in crafting coursework that was relevant to the student’s areas of interest and linking those back to computer science” [14]. Other curricula have investigated other realms of music creation. For example, in the Squeak-based curriculum Sound Thinking, students are asked to design and program novel musical instruments [15].

Several projects have attempted to bring a new approach to CS and mathematics by adding an artistic or cultural context. For instance, a web-based software application, Culturally Situated Design Tool (CSDTs), allows students to create cultural arts simulations such as Native

American beadwork, African American cornrow hairstyles, and urban graffiti, and so forth using underlying mathematical principles [16].

Besides, Culturally Situated Design Tool, Musical Rhythms, allows students to simulate traditional musical rhythms and create new rhythms of their own. Both Scratch and Alice teach programming through the creation of interactive media, improving students' creativity while lowering the technical barriers to entry [17] [18]. These platforms support diverse projects, giving students the tools to create something that captivates them. Music is a unique approach with broad applications, and its production has significant concepts in computer science. There have been some attempts to teach programming concepts through music education, hoping to draw a broad range of students' attention. The results from the four-year curriculum study on material linking CS and digital audio illustrate that students benefited most from practical projects that they felt were relevant to themselves and others. While the above approaches have reported success, it is unclear how authentic the learning experience is in both target domains. For example, the MediaCom course [16] uses audio at a deficient level (e.g., writing an algorithm to compute reverb) while Scratch presents a coding environment that is visually oriented rather than using a typical IDE. Therefore, we aim to go one step further in this research: writing computer programs to create beats for the games that students will develop using EarSketch and Scratch.

2.2 Machine Learning in Music

Many practical music teaching applications are using ML techniques. The main tasks in music that ML can solve are music score following, chord recognition, musical instrument identification, beat tracking, rhythm tracking, source separation, genre classification, and emotion detection [19].

In music, most of the tasks need an initial feature extraction and classification. Some of the feature extractions that can be used for Music Information Retrieval (MIR) tasks include Mel-frequency cepstral coefficients (MFCCs), chroma-based features, spectral flux, spectral centroid, spectral dissonance, and percussiveness [20]. Modeling the pattern of the extracted features plays a vital role in training the online automated model. Some of these models include

Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs), and support vector machines (SVMs). With all of the current ML techniques available, we can automatically retrieve information about a piece of music [19]. This information may include the instrument type(s), key, tempo, musical notation, pitch(es), segments, and chords present in a song. By automatically obtaining this information from a piece of music, whether a novice students recording or professionally recorded song, we may efficiently use it for other automation tasks.

Several metrics have been proposed to evaluate a real-time music tracking system. These metrics are primarily based on measuring the latency/error of every note event [21], or calculating the number of missing/misaligned notes during the process of score following [22]. There are, however, two significant issues in such evaluation methods. First, the performance of score following cannot fully represent an automatic accompaniment system operating in real-world environments, as it ignores the latency introduced in sound synthesis, data communication, and even reverberation of the environment. Second, note-level evaluation is suitable only for hard-onset instruments such as piano, while it is limited for soft-onset instruments such as violin, as the uncertainty of violin onset detection could propagate errors in the final evaluation results [4]. To solve these issues, we propose an experimental setup that allows evaluating the system in a real-world environment. Further, we provide a frame-level evaluation approach for general types of instruments, with intuitive visual diagrams demonstrating how the system interacts with humans during the performance.

2.3 The Pros of Online Piano Lessons for Users

The advantages of online piano lessons for middle school and high school students:

- 1. No firm schedule for students

  Beginners can learn to play the piano at their convenient timing. Anytime they are willing to practice, students have access to the online courses, which will be generated based on the user's first performance.

- 2. There is a variety of ways to Take Lessons

  The Internet has millions of different resources available.

Lessons are available through video and text tutorials, graphics, e-books, and even discussion forums. Many expert piano professionals are sharing what they know through video libraries and websites.

- 3. They are probably less expensive

  There is an abundance of free options available to learn music online.

- 4. Courses and Methods are updated frequently

  Online resources get updated more frequently.

  Online lessons are going to feature the latest methodologies and trends far faster than physical manuals or books. Students also will not be locked into a teacher who only has one way to teach.

- 5. Students can backtrack to previous lessons if they forget something and record their progress automatically.

  If there is any step that students do not remember or master immediately, students can revert a step or two. It is easy to go back and repeat a lesson.

  Users can proceed at their own pace. If they are not using an online teacher in real-time, they can even stop for breaks when required.

- 6. Users can play and follow along for the fun

  The user does not require a music background or the fundamentals of piano instruments to get started.

  Learning can be limited to playing just for fun, or a student can start at the primary, intermediate, or advanced learning levels.

- 7. Students can follow along via videos or live interaction

  With a high-resolution camera, an online lesson can become interactive. This is a great way to mimic what it would be like to have a real teacher in the house.

  Using online lessons from teachers in different time zones is possible online, making scheduling availability and convenience more powerful.

2.4 The Cons of Online Piano Lessons for Students

The disadvantages of online piano lessons for middle school and high school students:

1. It is easier for middle school students to get distracted by online lessons

When middle school students learn something online, mainly if it does not involve a human tutor, distraction is a real threat.

Emails, advertisements, and especially social media are all things that might take user's attention away from their piano work. Some students need a physical presence to keep them focused on the task at hand.

2. Free lessons will only get students so far

If the user does not take to the freebies, he/she might still have to pay for lessons with an in-person teacher or sign up for premium content at a website.

3. Students' playing technique may not be as good

Without a teacher being physically present, it can be challenging for an online human to judge the physical technique of a piano student, no matter how good their webcam is.

Also, sound quality can keep an online teacher away from truly hearing how well the piano is being played.

4. It takes longer or might be impossible to get answers to specific questions

It is not easy to get instant feedback from an online piano teacher.

Online teachers might not respond to questions promptly, especially in group lessons that do not involve individual attention or one-on-one interaction. Students could make mistakes that do not get caught, meaning repetition reinforces them into bad habits.

2.5 Free Sites to Learn Piano Online for Music Learners

Here is a list of free websites to learn music online:

YouTube

There are thousands, or more tutorials students can browse anytime to pick up a specific song or learn more about a particular skill.

It is unlikely that these tutorials can entirely replace a seasoned teacher. Nevertheless, they go a long way in answering questions the user might have between their formal lessons.

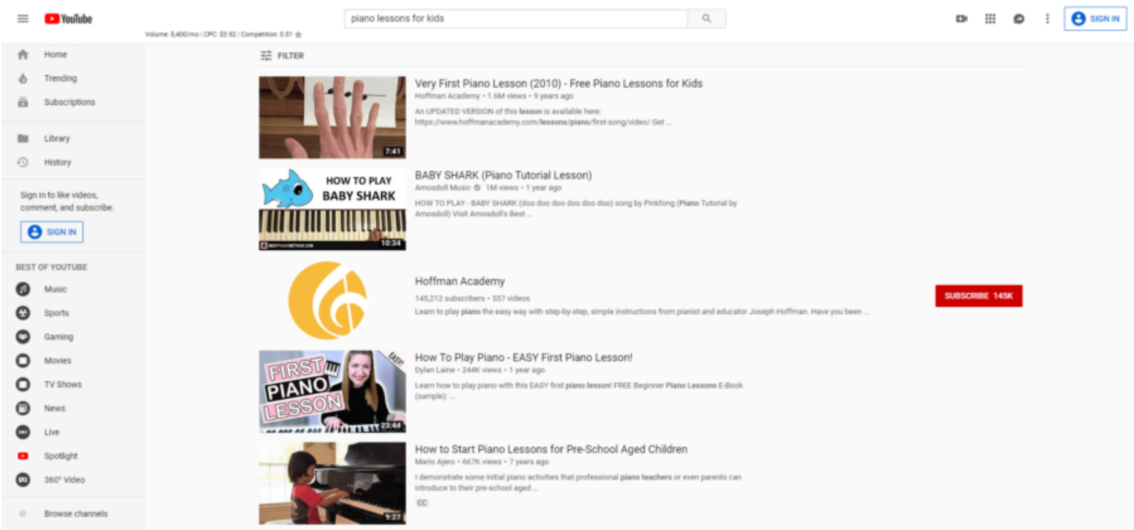


Figure 2.1. YouTube music lessons.

Piano Nanny

Piano on the internet was widely considered the first source of free online piano lessons, which started in early 1994. Piano Nanny now owns it, and they continue the award-winning legacy.

The sequence of 35-minute lessons can be worked at students' own pace slowly or quickly.

The course instructor is Clinton S. Clark, a Jazz musician, award-winning film composer, and member of both The Society of Composers Lyricists and ASCAP.

PianoLessons4Children.com

Maria Miller's project offers easily understood and free piano lessons. These lessons are not explicitly designed for the young generation but also teachers, parents, and beginners of any age.

Popular children's songs are an emphasis that makes the content friendly to middle school students. Parents that want to teach and learn at the same time will find this site useful.

Other helpful content includes lessons on famous composers, reading musical notes, and sing-along songs.

Zebra Keys

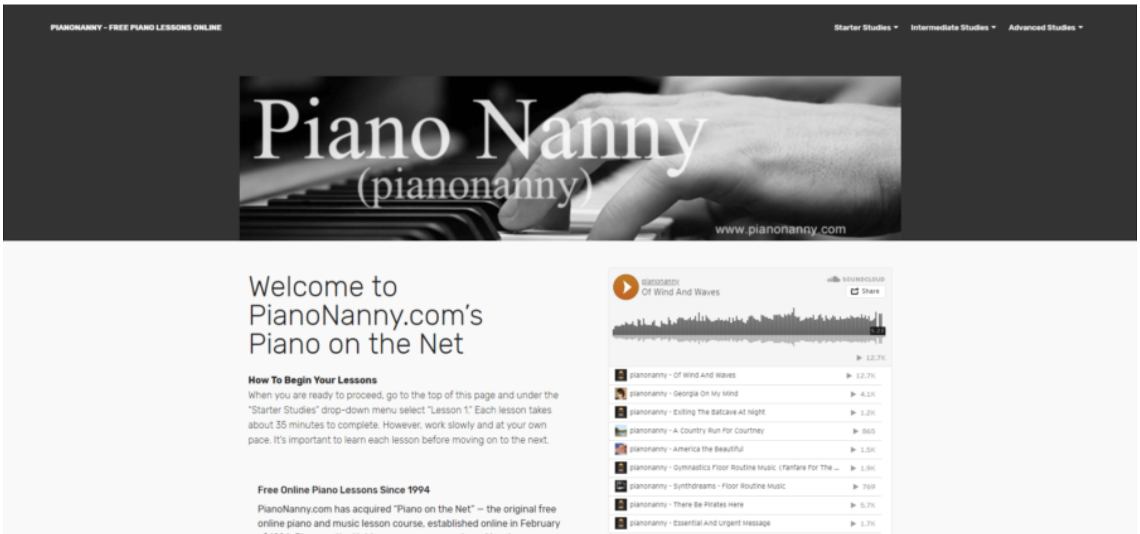


Figure 2.2. Piano Nanny demonstration.

Zebra Keys offers more than 50 piano lessons free across a range of difficulty levels.

Every piano lesson has visual flash animations that help the learners see and hear the song they are playing. The target audience is 13 years old and older.

Zebra Keys is also developed for ear training, public domain and free sheet music, and a constantly growing collection of reviews and songs.

HoffmanPiano.com

This site takes free lessons because every video piano lesson is completely free of charge to any visitor.

There are supplemental activity sheets that include practice tips, activity pages, and sheet music that students can buy; however, the video lessons are entirely free of charge.

2.6 The Best Online Piano Lessons for Kids

For more investment in online music lesson development, students can also consider the following premium possibilities to learn piano online:

Flowkey

The user-friendly interface helps make the lessons easy to enjoy and intuitive, considering popular songs from which learners can pick. If the song library were a bit bigger and theory lessons were added, Flowkey could rival the best of all online piano lessons.

Figure 2.3. PianoLessons4Children.com demonstration.

Flowkey is designed to get the user through the setup process as easily and quickly as possible so he/she can begin taking online piano lessons immediately. It is available as an app for users' favorite tablets or as online software for computers. Students can try out the software for free or purchase a monthly or yearly subscription. The free trial limits the number of songs and lessons, but full access to browse the list of songs and lessons available in premium. Students start the setup process by answering a couple of questions about whether they have played piano before and have access to a piano. After students answer those questions, the software is set up, and students can start learning.

The different modes and features of Flowkey are:

- Slow Mode - This feature allows the player to play along with the song at a slow speed to make the user feel comfortable with the virtual sheet music notes. In this section, the video also slows down without disturbing the audio.

- Fast Mode - This mode allows the player to play along with the song's original tempo for that specific song.

- Loop Mode - the player can choose a portion of the video tutorial and keep it in the loop until he/she gets it right.

Figure 2.4. Zebra Keys.

- Hand Selection - This feature is beneficial for more complicated songs. It is designed for beginners if the player initially gets confused by multiple keys being played and would like to master on the one hand first and then switch to another hand.

- Wait Mode - The virtual sheet music notes and the video wait for the player to play the notes after learning from the tutorial. Through the built-in microphone in the app, the wait mode detects the player's movement without making him/her connect the actual digital or acoustic piano to the app. Therefore, the Flowkey app provides feedback on each note the player plays.

Hoffman Academy

Joesph Hoffman created this online lesson program. He uses a classical approach as the video lesson instructor, but he excels in teaching the young generation. This works because of his classic starter songs and light-hearted stories along the way.

The program does not 'age' particularly well with middle school students, though. As they get into their preteens or teens, they might be more curious or interested in modern songs, which these lessons lack.

There are not enough reviews to objectively judge a quality rating for this program, but it is undoubtedly an affordable option at only $ 15 a month.

Playground Sessions

Figure 2.5. HoffmanPiano.com demonstration.

Playground Sessions is a Web-based music learning software that helps users subscribe to music theory lessons and provides a fun and practical experience for people learning the piano online.

Playground Sessions has several different elements that contribute to the learning experience.

- Interactive Lessons: This section is under the ”Bootcamp” tab in the app, where excerpts from well-known songs are designated to teach the player specific music concepts related to the song, with written instruction and game-like practice.
- Video Lessons: Interactive lessons follow the video lessons. They cover more details about the interactive lessons and allow the player to practice what was taught.
- Forums: This is a place for users to share tips, stay up to date on Playground Sessions news, ask questions, and lodge complaints.

Piano Marvel

Interactive piano lessons simplify learning the instrument into a straightforward sequence of lessons that will not overwhelm any child. It is cheap, complete, and entertaining, using a format that almost feels like a game that middle school students enjoy. However, finger numbering is not included in the very first lesson.

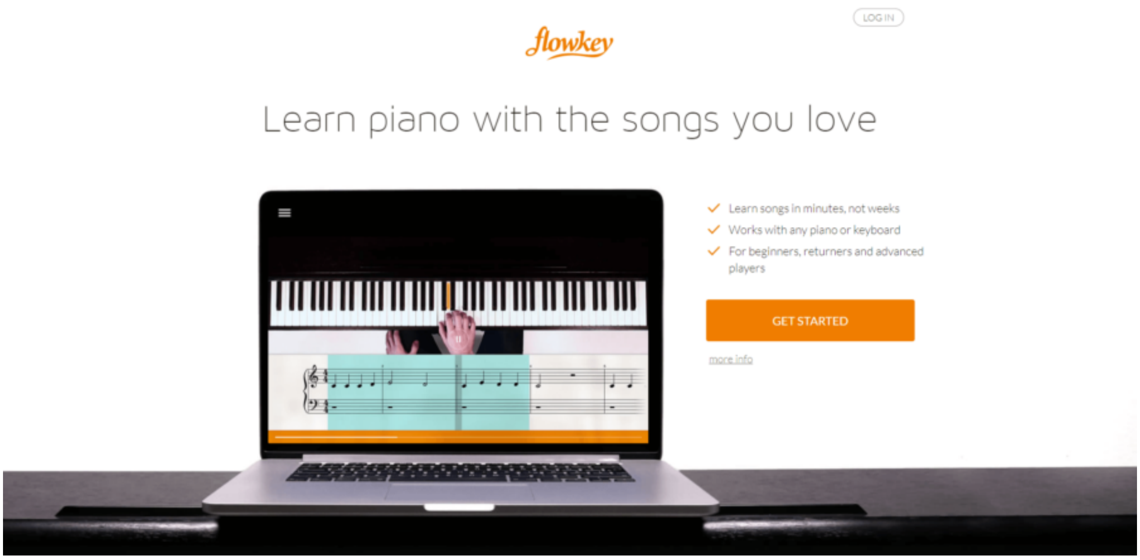


Figure 2.6. Flowkey home page screen shot.

LearnPianoIn30Days.com

Singer and songwriter Rachel James lead kids through a month-long series of piano lessons online.

The video lessons are informative and short, but they cover the fundamentals necessary for learners to learn piano successfully.

The lessons can be enjoyable, although self-motivation is more necessary than with other sites.

Skoove

This software package for learning piano online can be used on desktop computers but also Apple mobile devices. If students are learning on a MIDI keyboard, then this program is compatible.

The interface is a simple way for middle school students to have fun learning. That is because of having popular songs in many different musical genres and across various skill levels. The selection of modern music is diverse.

However, if the user wants to learn the crucial tenets of music theory, then Skoove is not a great choice. It lacks the music theory education included with the other options.

Also, pricing can be complicated. Paying $ 19.99 a month seems a bit expensive for what it offers. However, getting three months for a total of $ 39.99 is better.

Figure 2.7. Hoffman Academy online piano lessons for kids screen shot

Unlike Flowkey and Playground Sessions, our Web-based music learning application consists of a curriculum that teaches music concepts to students and teaches them how to compose a music piece. The other extra feature our application has compared to the other existing ones is that the player can review the other players' performances and rate them. The user also may want to join the other users to play a duet, which is the ongoing feature of our application.

Figure 2.8. Playground sessions home page screen shot.

Figure 2.9. Piano marvel home page screen shot.

Figure 2.10. Learn Piano in 30 days home page.

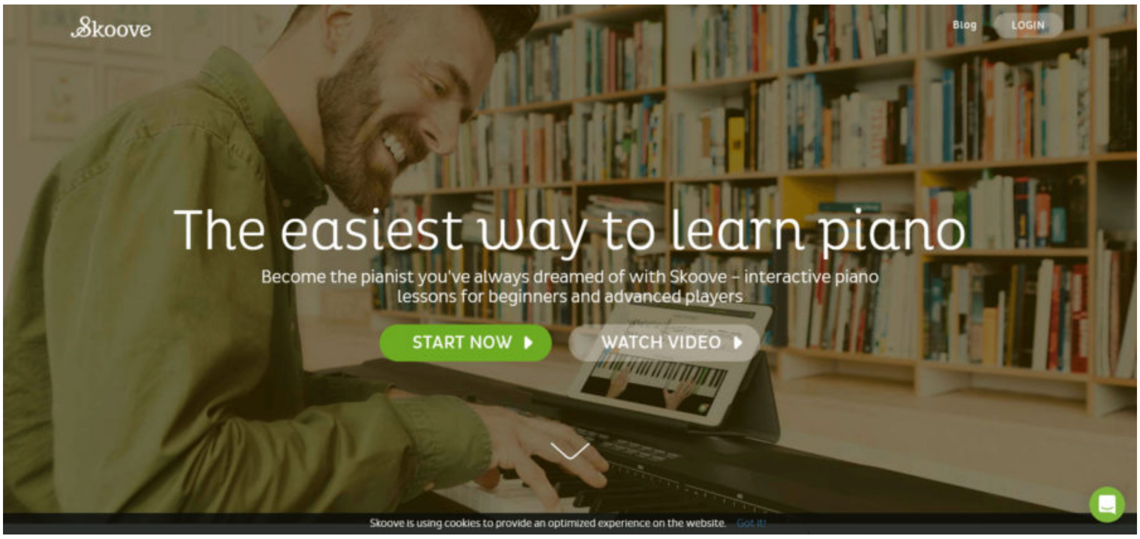


Figure 2.11. Skoove home page screen shot.

Chapter 3

Case Study One: Middle School and High School Students

The gender gap in technical fields is likely to persist for generations. Potential reasons include gender differences in aptitude, interest, and academic environment. Prior research [23] at the high school level has shown remarkable impacts on student engagement and intention to participate in computing and music, especially for female students. Computer music is one way to engage students in computer science (CS) by prioritizing personal expression, creativity, and aesthetics. This chapter aims to introduce a mixed-methods study to determine ways to promote CS among underrepresented groups. It describes an adaptation of EarSketch and Scratch for use in an secondary-school-level introductory programming course at an open-access summer camp at Auburn University. American girls aged 12 to 16 participated in pre-, post-, and follow-up surveys while attending the CS camp. To evaluate students' performance, we conducted path analysis exploring factors related to student engagement and intention to persist in computer music field.

3.1 Introduction

In this research, we aim to broaden the participation of underrepresented populations in computer music. Various approaches have sought to increase underrepresented recruitment and retention in computer music education. Some ideas include making CS and music more relevant or engaging through culturally specific technology [24]. These technologies include computer games and digital media or integrating artistic creativity within pedagogical practices as an in for motivating students [25][26].

In case one we describe our experience based on the paper Engaging underrepresented groups in high school introductory computing through computational remixing with EarSketch [27] with a focus on high school female students. According to the paper, ”the cultural relevance of an artistic domain relates to how central the artistic practice is to the target student culture”. Therefore, after a great deal of research on the camp population culture, we designed a Science, Technology, Engineering, Art, and Mathematics (STEAM) curriculum considering artistic practices that have the closest connection to students' background and culture (connection with music).

EarSketch (as visualized in Figure 3.1) is an authentic learning environment for engaging students interest in CS concepts through the remixing of musical sounds using the Python programming language and a Digital Audio Workstation. It includes a programming environment, digital audio workstation, structured curriculum, an audio loop library, and collaboration tools, allowing students an opportunity to experience both JavaScript and Python programming [23].

Scratch is a block-based platform developed to encourage young students to develop programming concepts and develop multimedia communication skills. It includes a visual system of Tiles” that contain blocks where students can connect to create programs. These programs direct the characters and objects in the game. Scratch can generate and play sounds using various components within the Sound category [28].

This research defines the most critical difficulties in integrating the arts into introductory CS courses (and approaches to address the challenges); it presents evidence to suggest the significant impact of artistic STEAM learning on student content knowledge and interest in computer music.

In contrast to other similar programs [23] [29][30], which mostly focus on teaching musical note structures and music remixing through computational thinking, this investigation goes one step further: improving students creativity and imagination by teaching them how to construct beats with EarSketch for the games they have developed using Scratch. The remaining sections of this research summarize work related to this theory, present the methods, findings, and results from the high-school pilot study, and discuss the results and future work.

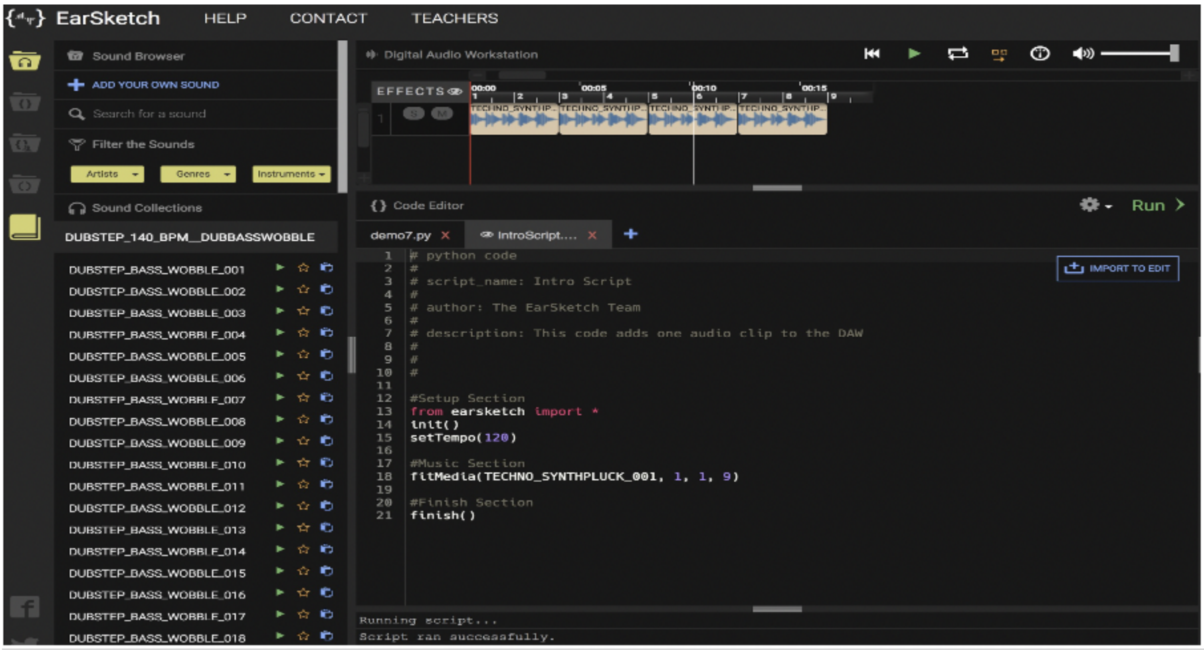


Figure 3.1. The Reaper Digital Audio Workstation (DAW) Music Production Software Integrated into EarSketch.

3.2 Curriculum and Teaching Strategies

3.2.1 Design and Development

We designed an online teaching platform, where teachers can have a one-on-one or a group session with their students. The software helps teachers to review the progress of their students throughout the week's practice. Besides, teachers will be able to upload their own recorded courses for the students. Thirdly, we developed the music theory teaching scriptwriting, a work classification, courseware development, and other links. Figure 3.2 presents the primary teaching module architecture.

3.2.2 Development of the Curriculum

In the non-technology-based teaching research, developing the different teaching models had to be according to students' perceptions and music learning ability. This model could be practical when educators share the same goal, and students have the same understanding level. Under the current teaching models of College Music teaching, a teaching object's attribute is not consistent. Therefore, to conduct the trusted quality of teaching objectives, it is essential to develop students' multi-attribute professional features and multi-objective future careers.

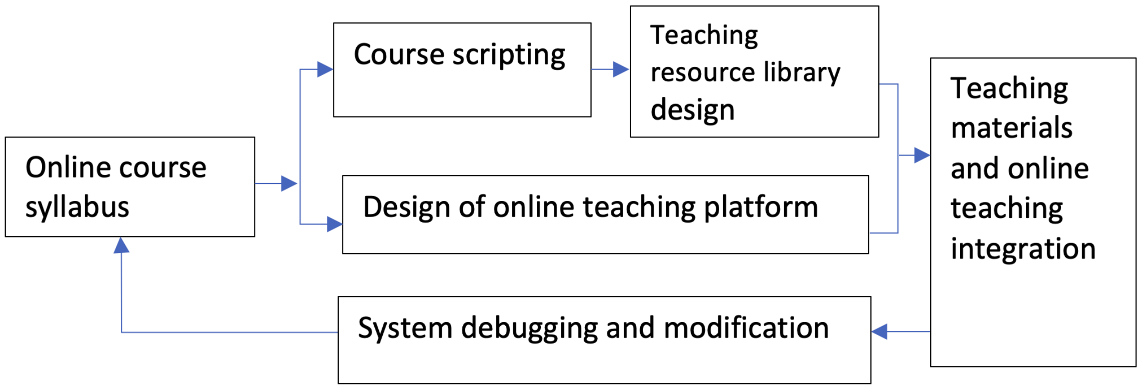


Figure 3.2. Primary teaching module architecture.

In the research and practice of ”diversity”, teaching, content, and target design is based on the students’ knowledge base, understanding ability, and so on. Therefore, in this project, we aim to implement an Artificially Intelligent application that can train itself based on the students’ different expertise and performance skills. The students’ attribute difference is not limited to the learning factor in the trusted teaching mode, but to individual behavior attribute, knowledge attribute, and personal attribute under corresponding training objective.

To determine students’ needs to support Web-based learning, we designed a music-technology-based curriculum to evaluate students’ interaction with the existing music learning applications and better address the missing components in our prototype.

We conducted this study in two phases, each with separate surveys and strategies. The first phase’s purpose was to develop and evaluate an online pilot program’s possibilities and requirements integrating computer technologies and music composition concepts for middle-school students. Opportunities to view and critique pilot online instructional units developed were included as a regular part of class activities for two graduate music education courses at Auburn University: CTMU 7520-26 (Curriculum and Teaching in Music Education) and CTMU 7540-46 (Evaluation of Programs in Music Education). In chapter 4, we present the review of music teachers of the curriculum and they changes suggested based on our data analysis.

The specific research questions for the study are as follows:

First Survey: Administrators

1. To what extent do comprehensive public high schools in the United States offer technology-based music classes?
2. To what extent does the district's socioeconomic status affect the likelihood of offering a technology-based music class?
3. To what extent does the district's geographic location affect the likelihood of offering a technology-based music class?
4. To what extent do/would school administrators value technology-based music classes?

Second Survey: Teachers

1. What is the curricular nature of these classes?
2. To what extent do these classes address nontraditional music students?
3. What is the professional background of teachers of technology-based music classes?
4. What types of software and hardware are being utilized in technology-based music classes?
5. How long have these classes been offered, and how were they initiated?
6. What level of support do school districts provide for these classes?

3.3 Methods

3.3.1 Pilot Population and Learning Context

Case one of the research is called Mentoring Alabama Girls in Computing (MAGIC) camp which was piloted with 24 middle school and high school students as a computing and music camp at Auburn University. The 1-week pilot took place from June 16th to June 20th, 2019.

The MAGIC curriculum includes EarSketch and Scratch, with the new addition of leadership concepts, is the first curriculum in a three-camp computing pathway - Robotics, App

Development and CS for all girls programs in Auburn and surrounding communities. Its curriculum includes some programming and CS principles and modules on topics such as leadership and music remixing. The EarSketch and Scratch pilot served as the programming module of the course.

Many programs have aimed to increase awareness about STEAM education for female students across the globe, such as the ”Girls, Music and Computer Science” and ”Music Education Meets Computer Science and Engineering Education” [31][32]. In our efforts to attract underrepresented female students towards future careers in CS and Music, we developed the MAGIC + M program - a curriculum combining music, CS, and leadership. In this project's context, we believe that Perfomatics [24] is a potential area that may attract students to CS. The main steps of the curriculum are:

- 1) introducing students to CS concepts and skills using Scratch music-related games [16].
- 2) introducing students to music-based storytelling.
- 3) teaching the basics of music programming using EarSketch
- 4) introducing students to basic concepts of leadership
- 5) teaching how to generate their musical composition, and
- 6) providing an opportunity for student presentations in a music composition competition.

The MAGIC teaching methodology is founded upon a project-based, ”flipped classroom” learning model, which resembles a studio-based learning approach that has seen practical application in CS curricula [33].

The pilot instructors created a curriculum derived from the openly accessible EarSketch and Scratch online curriculum and leadership concepts.

The students enrolled in the MAGIC pilot did not self-select for a course with EarSketch, Blockly programming, and leadership or even a course in CS. Their interest was in the broader camp program at Auburn. Students did not begin the class with substantial computer and computing experience, as proven by their responses to our engagement survey.

3.3.2 Curriculum and Teaching Strategies

During the pilot, instructors mostly focused on:

- 1) maximizing student motivation,
- 2) interweaving the teaching of programming, music, and leadership concepts and techniques,
- 3) encouraging creative applications of the acquired knowledge via individual and group projects, and
- 4) benchmarking progress via quizzes.

Maximizing Student Motivation:

MAGIC project was a collaborative project between experts in their academic areas. Instructors employed several complementary strategies to motivate students. Students' immediate goal of producing personally expressive, innovative music to share with the class encouraged their fellow peers to learn to code more entertainingly. In particular, the course emphasized how programming (instead of creating music manually in the DAW) enables students to work more efficiently. We approached this by automating repetitive tasks, to more rapidly experiment with different musical possibilities and variations, and to create unique musical structures and sounds to make their music and game stand out. However, grasping the basics of CS concepts and applying them in composing songs at the beginning seemed complicated for students. Therefore, we first familiarized students with block programming using Scratch, where students could define their game story and implement their game with instructors' help. Students showed a great interest in making their own game and were motivated to compose their songs for each component of their game after fully understanding CS concepts and their connection with music. Instructors also emphasized the importance of programming in the music industry's future and the opportunities to become industry leaders by developing new tools for making, performing, and distributing music.

Figure 3.3. Students sharing their projects.

Interweaving Programming and Music:

Throughout the pilot, students learned to design and develop musical applications using their newly grasped music, game development, and leadership skills. Days 1 and 2, students described their game story using the computing concepts learned in Scratch to develop their game interfaces. In days 2 and 3, we introduced fundamental music and melody/ rhythm pattern concepts, including tracks, tempo, instrument pairings, and sound effects and filters. In the afternoon of each day, we moved to leadership development sessions, and relevant programming concepts, including variables, function calls, and core EarSketch Application programming interface (API) functions for placing sounds and effects on tracks. The course also addressed sharing both music and code, open licensing models in both domains, and sharing practices on the EarSketch and Scratch social media site. Throughout the course, especially early on, we played popular music to demonstrate specific concepts and motivate students to make their versions.

On days 3 and 4, we introduced rhythms and repetitions in music and their relevance to programming concepts, including strings and iteration. Day 5 focused on musical compositions' arrangement as a series of musical sections, with students learning to define their functions to encapsulate and re-use musical sections. At the end of the pilot, students reviewed, and reinforced previous materials learned in the camp. They deepened their understanding of computing concepts, API functions, musical techniques, and individually developed projects that served as final group projects.

Quizzes and Projects:

To ensure that students effectively acquired and internalized the MAGIC curriculum, the instructor created two types of assessments: 1) quizzes, which were not multiple-choice exams but instead asked for answers in the form of online Python code; and 2) Individual and group projects that demonstrated integration and application of music, programming concepts, and leadership.

The quizzes asked the students to write Python scripts to make music with specific computational and musical requirements. The requirements included programming techniques (such as variables, constants, strings, loops, functions, and lists) [27] in Scratch and EarSketch API functions (to perform tasks such as playing audio files on the timeline, creating rhythmic beats, and controlling track effects and effect envelopes). For instance, Quiz 1 included writing a Python script that creates a minimum of two tracks of varying lengths, which uses multiple sound files and applies filter level adjustments to the tracks that go alternatively through the sound files. Projects were similar to the quizzes, as they required the development of a code to create music but were more open-ended to provide students with more opportunities for artistic creativity.

Students were asked to take online quizzes during the camp and do individual and group projects where the instructor could use a technique similar to pedagogical code review processes to grade students [33] and engaged in individual discussions with students to peruse each student's code and composition. Individual discussions with students enabled the instructor to detect errors that an individual student was making, such as incorrect API usage or application of iteration, to improve the student's comprehension. At the end of Days 4 and 5, students had their presentations to share their projects with the class and get peer reviews on their projects.

3.4 Software Developed by Participants

This section represents two of the most innovative projects developed by the girls who participated in the camp.

3.4.1 Keyboard Music

Student one developed the Keyboard Music Game. It was a game developed to improve the player's musical perception of rhythmic and melodic patterns. The game has three levels of difficulty: Easy, medium, and challenging. After choosing the difficulty setting, the user needs to follow the rhythm and melody given in a sequence and then play that particular sequence simultaneously (shown in Figure 3.4). The game's idea was to analyze what the user played on the digital piano and give him visual feedback by showing what was played correctly in green and erroneously in red; however, the last part of the project was not implemented due to the lack of time.

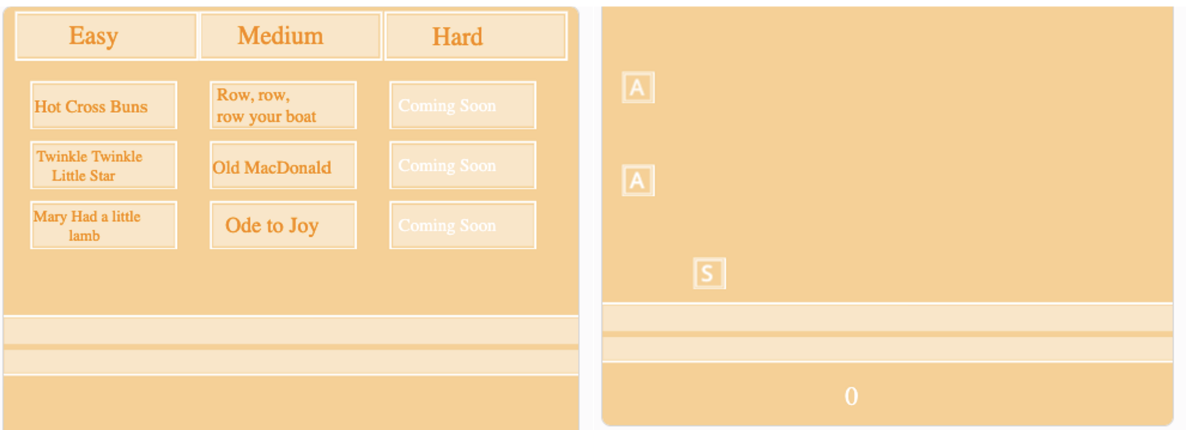
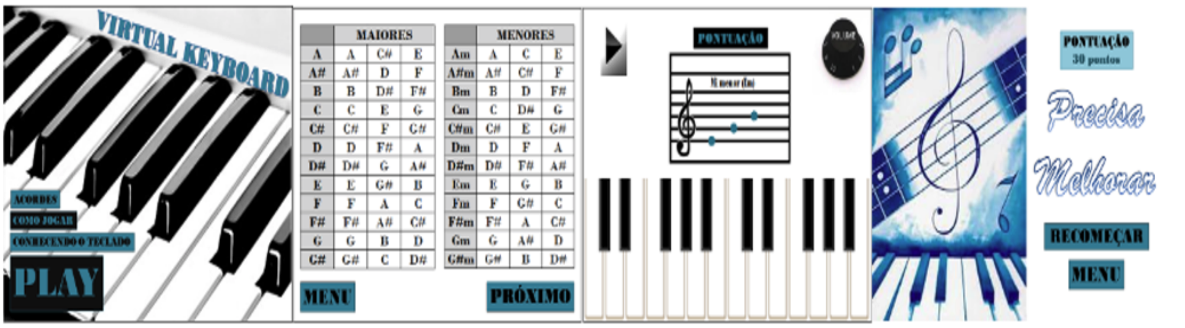


Figure 3.4. Student project. Keyboard Music

3.4.2 Keyboard Chord Trainer

The Keyboard Trainer was developed by student two. The goal of this game was to improve the users music score reading at first glance, focusing on major, minor and seventh Chords. From the main screen, it is possible to select to look through the student's animated videos explaining how the chords were formed and to play the game. The game shows the user a succession of chord formations that the user needs to execute on the piano simulator. After a given number of chords, the game ends, showing the score screen (Figure 3.5).



(a) Main Screen (b) Chord Formation (c) Game Screen (d) Score Screen

Figure 3.5. The main screens of the Keyboard Chord Trainer

3.5 Evaluation

In this pilot study, we developed qualitative and quantitative surveys where students were measured on attitude changes and content knowledge. A student engagement survey was given to students as a pre/post retrospective at the pilot's beginning and end. The approach asked students at the beginning of the camp to consider how they feel about the course and their background in music and computer science concepts. At the end of the camp, we asked students to indicate their perceptions of the course.

The instrument used in this project is based on Engaging Underrepresented Groups in High School Introductory Computing through Computational Remixing with EarSketch" in which we measure six psychosocial constructs: 1) computing confidence (e.g., "Computers can enhance the presentation of my work to a degree which justifies extra effort."); 2) computing enjoyment (e.g., "I like working with computers and programming"); 3) computing importance/perceived usefulness (e.g., "Computers help me to organize my work better"); 4) motivation to succeed (e.g., "I can make the computer do what I want it to do"); 5) identity and belonging in computing (e.g., "I take pride in my computer abilities") and 6) creativity in computing (e.g., ”I can be very expressive and creative while doing computing”). The fifteen-question survey was designed to demonstrate these six psychosocial constructs’ changes after attending the program activities.

Our analysis includes a paired samples t-test to address significant changes from pretest to post-test.

Our content knowledge assessment (CKA) measures students’ understanding of generic computer science concepts. It was administered as a pre/post assessment with five multiple-choice items. It covers specific introductory computing topics - commands, variables, conditionals, loops, debugging, and EarSketch and Scratch functionality.

Overall, the student engagement survey (Table 3.1) indicate a positive and statistically significant increase in students’ attitudes towards computing across all constructs at p¡0.2. Before the course, female students were statistically significantly low in computing confidence. Additionally, gains in the female ”motivation to succeed” shows that the course may be particularly useful in increasing female motivation to persist in computing problems.

Students across all racial/ethnic groups rated the course as being good or excellent. Many open-ended questions specifically addressed the thickly authentic aspects of the learning environment. One student stated, I favored making beats with Earsketch. I say this because it felt exciting, like a breath of fresh air. Consequently, I could use this information to teach my classmates who make beats. I coherently believe that this would be useful research for them.” Another student said, ”I loved the part where we got to make our music using EarSketch. We got to know things that people in the music industry do nowadays,” Additionally, another articulated, ”I like that we used different techniques to do different things on making videos and music”. Though not statistically significant, female and minority students rated the course higher than their counterparts, suggesting the addition of music to computer science through EarSketch is specifically significant among underrepresented/underserved groups in computing.

Furthermore, we designed CKAs to determine the progress of students in a more quantitative way. Our assessment results showed statistically significant increases in computing content

psychosocial_constructs	Pre_test	Post_test	T-test
computing_confidence	3.9784334	3.75	0.15861459
computing_enjoyment	3.64	4.10	0.15861459
computing_usefulness/importance	4.02083333	4.00	0.15861459
motivation_to_succeed	4.16337719	4.43	0.15861459
identity_and_belonging_in_computing	3.93796296	4.29	0.15861459
creativity_in_computing	4.30	4.50	0.15861459

Table 3.1. Students Engagement Survey

knowledge from before to after the camp. On average, students answered 15% of the items correct at the beginning of the course (pre) and more than 70% correct at the end of the course (post).

3.5.1 Theory of Change

We are using path analysis as an extension of multiple regression, which determines the relationships between many variables. Therefore, instead of building multiple regression models, the path analysis can analyze the theory of change with a single model. In this chapter, we are using three fit indices to check the stability of the model. The Comparative Fit Index (CFI), the Root Mean Square Error of Approximation (RMSEA), and the Standardized Root Mean Square Residual (SRMR). Recommended cut-off values for the fit indices are CFI above .90, RMSEA below .08, and SRMR below .06 [10].

We used path analysis to determine the magnitude of the hypothesized factors in the theory of change model (see Figure 3.6).

This research mostly focuses on relationships between authenticity and the six student attitude constructs (computing confidence, enjoyment, importance and perceived usefulness, motivation to succeed, identity/belongingness, and personal creativity).

3.6 Discussion and Future Work

These evaluation results suggest that EarSketchs music and computing learning environment effectively teaches introductory computing concepts to high school students in an informal program and that it substantially improves student attitudes towards computing. Some of the data

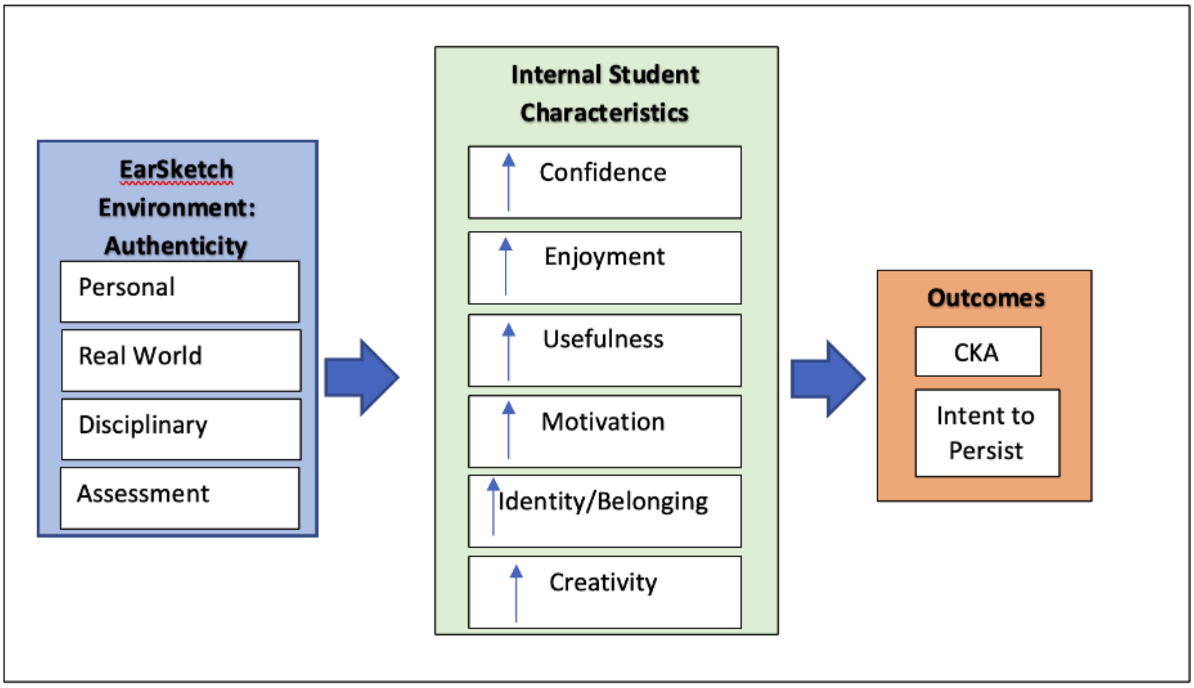


Figure 3.6. Theory of change model.

suggest that EarSketch may be particularly useful for traditionally underrepresented groups in computing. Moving forward, we plan to deploy and evaluate the addition of more technical musical concepts on a larger scale and determine the effectiveness of EarSketch and other similar software. To this end, we: a) have made our curriculum available online, b) are developing online instructor training materials and planning instructor training., and c) are collaborating with the music education department at Auburn University to further expand this project to an undergraduate course for music and CS students. We are also beginning to create an entirely web-based version of our application for students with disabilities to learn CS through music. This research intends to lay the foundation so scholars can learn more about why EarSketch has been significant, develop new assessment tools and software enhancements, be informed about learning sciences, and better understand the specific aspects of EarSketch that have contributed to particular student engagement and content knowledge outcomes. Finally, as an additional part of this project, we are developing an online music learning platform where students will learn music concepts more intuitively.

Chapter 4

Case Study Two: Using Qualitative Methods to Assess the Impact of a Multi-modal Online Music Education Program on Middle-school Students

The purpose of this study was to identify the characteristics of teaching music online used to accompany a music computing curriculum and investigate the effects of such curriculum using a qualitative approach. Semi-structured interviews were conducted with a sample of music teacher participants (N = 15), all with at least three years of experience in teaching music. Opportunities to view and critique pilot online instructional curriculum units were included as a regular part of class activities for two graduate-level music education courses: CTMU 7520-26 (Curriculum and Teaching in Music Education) and CTMU 7540-46 (Evaluation of Programs in Music Education). We conducted an inductive content analysis of the interview data and will present the results in this chapter. The findings demonstrated the importance of musical (e.g., rhythm, harmony, melody), contextual (e.g., scaffolding, skill-building practice strategy, relevance), and individual factors (e.g., prior background and knowledge) in determining essential characteristics of online music courses. The findings point towards a more comprehensive conceptual framework. In particular, facets of teaching music such as instruction, motivation, philosophy, skills, strategies, and technology are discussed. Findings have implications for the use of technology in music curriculum settings [34].

4.1 Software Tools to Teach Music

Music teaching and learning software could be a tool that helps musical practices [35]. The ”ubiquitous music” trend [36], proposes participatory practices of music education that exploit the available technological infrastructure. On the other hand, it expands such values as musical versatility and flexibility, as well as mobility between various musical communities of practice [37]. These methods encompass multiple modes of digital artistry: in face-to-face pedagogical situations, informal learning contexts, and open networked learning environments as online musical platforms. This model allows students to experience transitions between their musical engagement and learning in and out of school.

4.2 Method

4.2.1 Research Design

This research was designed as a qualitative study. Initially, a scale development study consisting of two phases was conducted. The first research aimed to explore the factor structure of the newly developed Attitude towards Using Technology in Music Education Scale. In contrast, the second research aimed to confirm the extracted factor structure. Therefore, we designed a comprehensive Music computing curriculum based on case study one to teach middle school students the basics of music theory and programming. Then, we conducted a qualitative analysis on music teachers' perspectives towards using technology in music education, their demographic information, and musical background.

4.2.2 Participants

The sample 15 participants were selected according to their experience of teaching music; the population of interest was students of two graduate music education courses: CTMU 7520-26 (Curriculum and Teaching in Music Education) and CTMU 7540-46 (Evaluation of Programs in Music Education) at Auburn University. Each participant had at least three years of experience teaching Music.

4.2.3 Data Collection

In order to conduct a qualitative analysis of outcomes and underlying reasons for these outcomes, we gathered feedback from 15 expert music teachers, reviewing the music and programming curriculum designed for middle school students. The curriculum includes five units teaching rhythm, melody, dynamics, and effects, and harmony.

We created an online open-ended questionnaire to examine the weaknesses and strengths of the curriculum. Teachers in the sample were invited to complete a survey that included questions about their overall insight, recommendations, and strong points of the program. Following the initial survey, I constructed an individual phenomenological text for each participant. These texts were summaries and initial analyses of the data from the open-ended surveys. The open-ended questionnaire was as follows:

1. What do you view as any strong points of this program?
2. What are your recommendations for improving this program?
3. Other thoughts or talking points?

We asked questions designed to assess the program's impact on both teachers and students. This resulted in some excellent insights into how music teachers perceived the impact on both groups in response to program participation. The following year, we dug deeper to establish a more comprehensive curriculum based on our qualitative analysis of music teachers' program reviews.

4.2.4 Data Analysis

To code the data, I used three cycles of coding [38] [**?**]. In the first cycle, I used open coding, creating in vivo codes [39], of the quotes which were more relevant to the purpose of the study. In the second cycle of coding, I combined the first cycle codes into pattern codes [38] [40], and in the third cycle of coding, I combined the pattern codes into five themes with narrative descriptions.

Data included written reflections from each music teacher (fifteen total) and transcripts from responses to the open-ended questionnaire. In an attempt to establish trustworthiness, I used multiple-member checks and triangulation of sources and methods. For this study, formal member checks occurred after the first analysis. The fifteen music teachers in this study constitute multiple sources, and the written reflections and surveys constitute different methods to collect data.

Immediately after consenting to participate in the study, mentors completed written reflections via a course discussion. After collecting the written reflections, I began the "descriptive phase" of analysis [41]. I read the responses, compared them to the research question, and devised an initial list of codes.

I uploaded the written reflections to Atlas.ti, a practical software program for qualitative analysis. I added to the initial code list where appropriate to represent new emerging codes and revised initial codes where necessary in light of new data from the surveys. I completed the first round of analysis by coding the surveys and recoding the written reflections using the updated code list.

Then, I began the "interpretive phase" of analysis [41]. I reviewed and compared the coded reflections to identify similarities and differences.

During the interpretive phase, I engaged in a "theorizing" process in which the researcher thinks "about how items, patterns, and structures, as well as variables, factors, and domains, are related and reasons for their occurrence" [42]. I referred to the "code co-occurrence" and "query tool" in Atlas.ti and continued refining and revising codes.

During this process, codes merged into categories, and categories began to form themes. I returned to the coded data to choose quotes that highlighted the themes.

4.3 Findings

In this section we describe some of the quotes to uncover some of the possible reasons why the program appeared to be engaging and practical in learning music. We begin by describing The themes of the research (see Table 4.1).

Based on the importance and relevance of the categories we divided our themes into five themes: Instruction, Philosophy, Skills, Strategies, and Technology. The most common recurring theme dealt with teaching philosophies (see Figure 4.1).

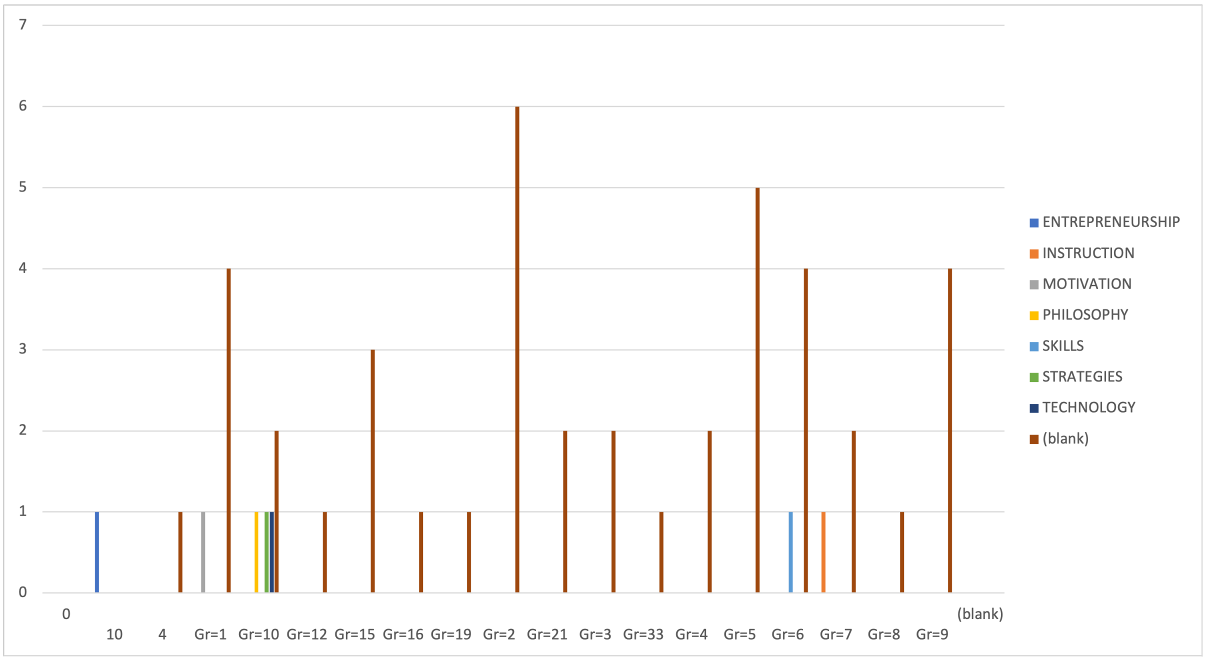


Figure 4.1. Qualitative Content Analysis (Co-occurrence and Query tools).

Specifically, music teachers described the necessity of students prior music knowledge and the proper scaffolding of the curriculum. They found the curriculum engaging, intuitive, and relevant to the research questions. It is important to note, however, that in spite of the predominantly positive feedback on the curriculum, many teachers made comments that could be construed as outliers, adding depth and complexity to the findings reported earlier. For example, one of the teachers commented, I believe the designers approached the music portion of their curriculum as if it were a content subject and not a skills subject. This content raised questions about how best to describe change in the context of such program. Music teachers also noted educational pacing and sequence of the curriculum. In Appendix we represent typical comments relating to the outcomes of music teachers written reflections to the curriculum.

Categories	**Codes**
ENTREPRENEURSHIP	Entrepreneurship: Innovation Entrepreneurship: Promotion
INSTRUCTION	instruction: individualized instruction instruction: kid-oriented and friendly instruction: modeling: teacher instruction: research-based instruction: scaffolding instruction: student centered instruction: teacher centered instruction: tools
MOTIVATION	Motivation: Extrinsic Motivation: Intrinsic Motivation: Perseverance
PHILOSOPHY	philosophy: complexity level philosophy: learning: essential skill philosophy: learning: problem solving philosophy: learning: Self-Awareness: cognitive philosophy: practice philosophy: prior knowledge, and sequencing. philosophy: relevance philosophy: teaching quality philosophy: wellness refinement of this curriculum secondary music education philosophy: Production quality issue
SKILLS	skills subject. Skills: educational pacing. Skills: Effects Skills: Efficiency Skills: Harmony Skills: Melody Skills: Rhythm: Note Values Skills: Time signatures (Tempo/beats)
STRATEGIES	strategies: goal setting strategies: thinking strategies: analysis strategies: creativity strategies: distributed practice strategies: engagement strategies: peer-evaluation strategies: repetition and understanding strategies: self-evaluation strategies: skill-building practice strategies: slow practice
TECHNOLOGY	technology model cornerstone assessment. technology: Apps technology: Instructional blogs technology: Instructional videos technology: LANDSCAPE/CHALLENGES

Table 4.1. Qualitative Code Cloud

Chapter 5

Automated Chord Detection

Massive Open Online Courses (MOOCs) are growing as we are moving to online classes. Current music courses through MOOCs mostly focus on peer evaluation for assessing the students' performance. However, this technique may not be practical when it is applied to larger class sizes. Therefore, in this research, the main goal is to reduce the instructor's load and provide online real-time performance feedback. As a contribution to music education, we propose a new technological framework to automate music lessons for learning how to play any favorite songs via existing machine learning (ML) techniques for adaptability to various learning styles.

We discuss the main problems with existing online music lessons that ML techniques can resolve:

1. Finding or developing a music lesson based on the student's learning style, musical background, or preference.

2. Providing quantitative and qualitative assessments of the student's performance.

This chapter discusses the tools for facilitating assessment where there is a semi-automatic assessment system that can train itself based on the instructors' real-life assessments on a small group and further assess larger sets of performances.

5.1 Proposed Framework of MOOCs for Music Learning and Performance: Module 1: Self-learning Tutorials

Online learning environments are divided into three categories: online video tutorials (e.g., YouTube), face-to-face video calls with the teacher, and Massive Online Music (MOOCs)

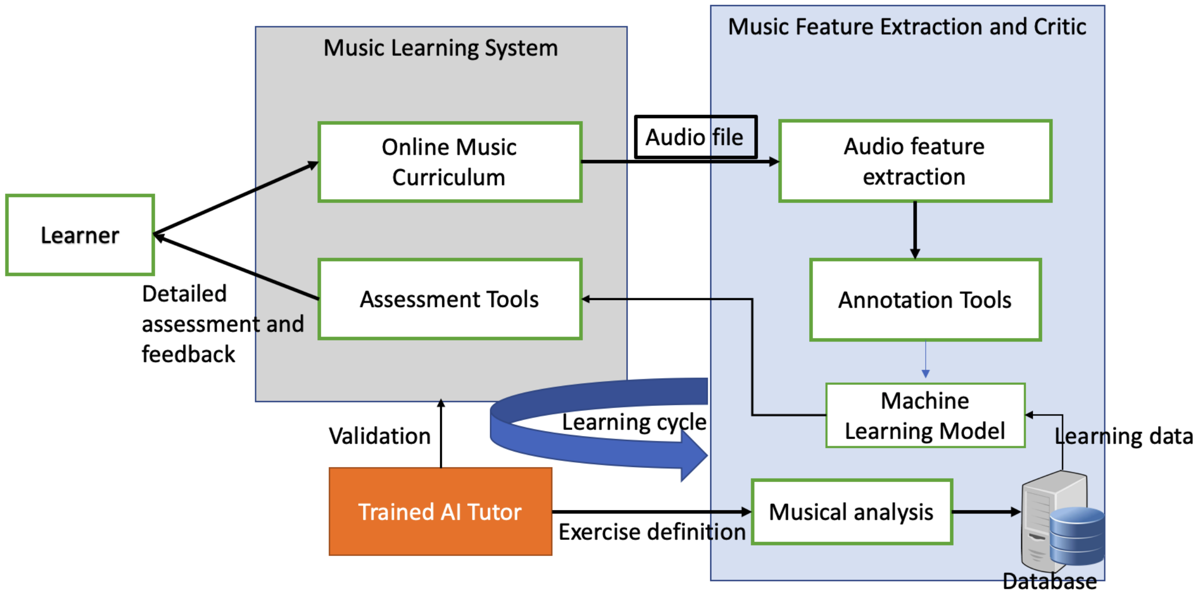


Figure 5.1. Music education framework

Courses. To learn music online, one needs to be self-motivated and self-disciplined. Online video tutorials may lack a real-time interaction with the learner, which is essential for music learning. The one-on-one video calls may present an interactive learning environment; however, they can not be practical in music learning when used for larger audiences. Therefore, to address the interactivity, scalability, and automating the development of music tutorials, machine learning models can be used. Figure 6.1 presents a framework for supporting MOOCs to increase their scalability to large audiences [43][44].

The basics of the framework are: (1) Trained AI Tutor provides a music exercise using the Music Learning System (MLS); (2) learner uses interfaces for practicing and learning. Then the learner uploads his/her exercise's music performance to the MLS; (3) MLS sends the audio recording to Music Feature Extraction and Critic, where it is analyzed and further presented to the trained AI tutor for assessment.

To create automated lesson plans based on class recordings, some necessary information, including pitch, chord, beat, duration, rhythm, and dynamics, should be retrieved from the music file. This data can be used as valuable features for training a model to assess a students' performance while learning to play an instrument. One component of the model is the student practice and recording interface that can be easily tailored to specific exercises by the education content designer (music instructor). Our initial model provides better results with a simple instrument such as a flute in which the player can play one note at a time.

In this chapter, we focus on the chord recognition task which is one of the most important tasks in Music Information Retrieval (MIR).

5.2 Data Collection and Pre-processing

5.2.1 Feature Extraction

In music, the term chroma feature or chromagram closely relates to the twelve different pitch classes. Chroma-based features, which are also referred to as ”pitch class profiles”, are a powerful tool for analyzing music whose pitches can be meaningfully categorized (often into twelve categories) and whose tuning approximates to the equal-tempered scale. One main property of chroma features is that they capture harmonic and melodic characteristics of music, while being robust to changes in timbre and instrumentation.

Identifying pitches that differ by an octave, chroma features show a high degree of robustness to variations in timbre and closely correlate to the musical aspect of harmony. This is the reason why chroma features are a well-established tool for processing and analyzing music data. For example, basically every chord recognition procedure relies on some kind of chroma representation. Also, chroma features have become the de facto standard for tasks such as music alignment and synchronization as well as audio structure analysis.

Extraction process

- Speech is analyzed over short analysis window
- For each short analysis window a spectrum is obtained using FFT (Fast Fourier Transform)
- Spectrum is passed through Mel-Filters to obtain MelSpectrum

These Mel-filters are non-uniformly spaced on the frequency axis more filters in the low frequency regions and less no. of filters in high frequency regions (similar to human ear).

An object of type MelSpectrogram represents an acoustic time-frequency representation of a sound: the power spectral density P(f, t) . It is sampled into a number of points around equally spaced times ti and frequencies fj (on a Mel frequency scale).

The mel frequency scale is defined as:

mel = 2595 * log10 (1 + hertz / 700)

We used librosa library to extract mel spectrogram chroma features from the audio datasets. We built a new dataset consisting of mel spectrograms of the chroma features of all the audio files in the guitar chord datasets as input and corresponding class id as output. Since the audio files in our dataset are of varying duration (up to 4 s), we fixed the size of the input to 2 seconds (128 frames), i.e. X $\in R^{128}\ X^{87}$

5.3 Proposed Framework of MOOCs for Music Learning and Performance: Module 2: Chord Recognition

In music, a combination of different notes that are played simultaneously is called harmony. The main components of harmony are chords, which are musical constructs that consist of multiple notes (three or more).

The result of a chord recognition task consists of dividing an audio file into smaller segments and assigning a chord label to each segment. ”The segmentation represents the timeline of a chord, and the chord label classifies which chord is played during a specific period of time. A typical chord recognition system consists of two essential steps” [1].

- In the first step, the given audio recording is cut into frames, and each frame is transformed into an appropriate feature vector. Most recognition systems use chroma-based audio features, ”which correlate to the underlying tonal information contained in the audio signal”.

- In the second step, pattern recognition techniques are used to map each feature vector to a set of predefined chord labels.

Figure 6.2 represents a diagram of chord recognition process.

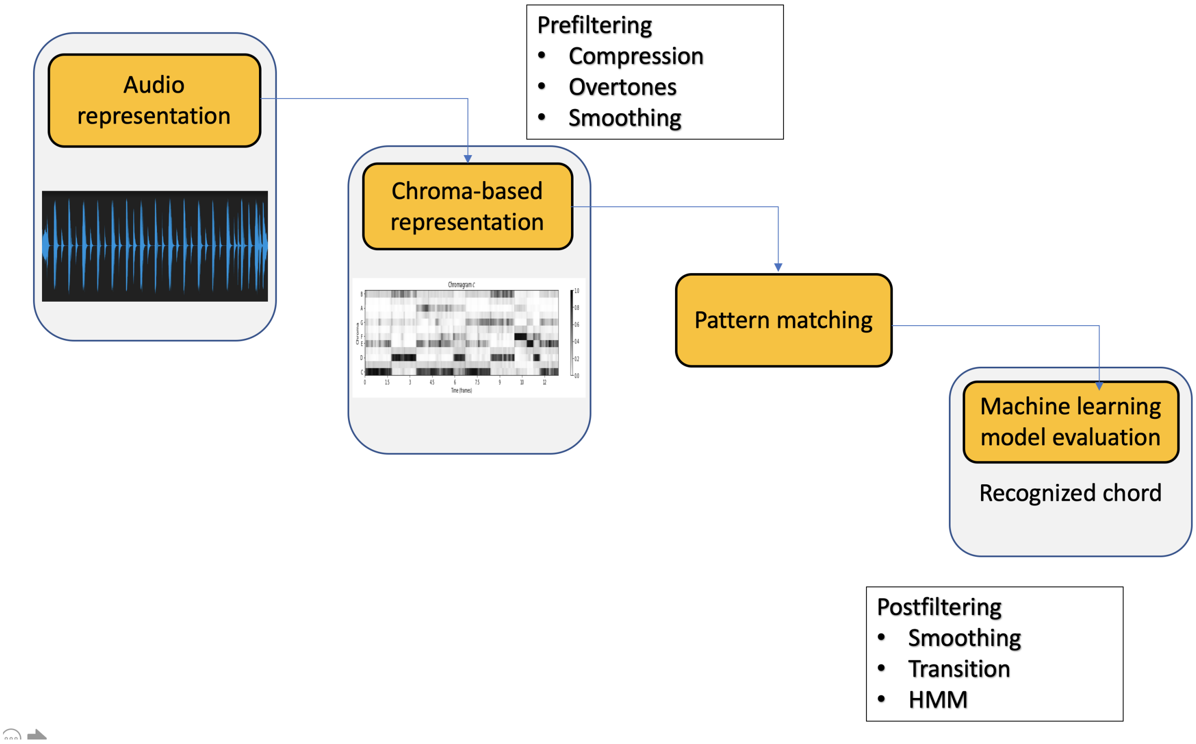


Figure 5.2. Chord recognition diagram

5.3.1 Template-based Pattern Matching

One of the techniques to detect a chord is through matching the chromagram of each segment with a predefined Template-based pattern matrix. For example we assume given a sequence X = $\{x_1, x_2, ..., x_N\}$ and a set Λ of all chord labels. Template-based chord recognition aims to map each chromagram vector $x_n \in R^{12}$ to a chord label $\lambda_N \in \Lambda$, $n \in [1 : N]$.

Consider the following set:

$$\Lambda = \{C, C^{\#}, D, ..., B\} \tag{5.1}$$

To simplify the problem, we convert all the possible intervals of chords to the main twelve major and twelve minor triads. Therefore, each frame $n \in [1 : N]$ is mapped to a major chord or a minor chord considered as λ_n.

We first pre-compute a set

$$\tau \subset F = R^{12} \tag{5.2}$$

of templates denoted by $t_\lambda \in \tau$, $\lambda \in \Lambda$. Each template can be considered as a prototypical chromagram vector that represents a musical chord. Moreover, we fix a similarity measure by

$$s : F \times F \to R \tag{5.3}$$

that allows comparing different chromagram vectors. Then, the Template-based procedure consists of classifying the chord label that maximizes the similarity between the corresponding template and the given feature vector x_n:

$$\lambda := argmaxs(t_\lambda, x_n) \tag{5.4}$$

In this procedure, there are three main concerns:

1. Which chords should be considered in τ ?
2. How are the chord templates defined?
3. What is the best evaluation method to compare the feature vectors with the chord templates?

Based on [1], for the chord label set Λ , we select the twelve major and twelve minor triads. Considering chords up to enharmonic and up to octave shifts, each triad can be coded by a three-element subset of [0:11]. For example, the C major chord C corresponds to the subset 0,4,7. Each subset, in turn, can be classified with a binary twelve-dimensional chroma vector $x = (x(0), x(1), , x(11))$, where x(i)=1 if and only if the chroma value $i \in [0 : 11]$ is in the chord.

For example, for the C -major chord, the resulting chroma vector is

$$t_C := x = (1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0)^T \tag{5.5}$$

The Template-based pattern mappings based on twelve major and twelve minor chords are shown in Figure 6.3.

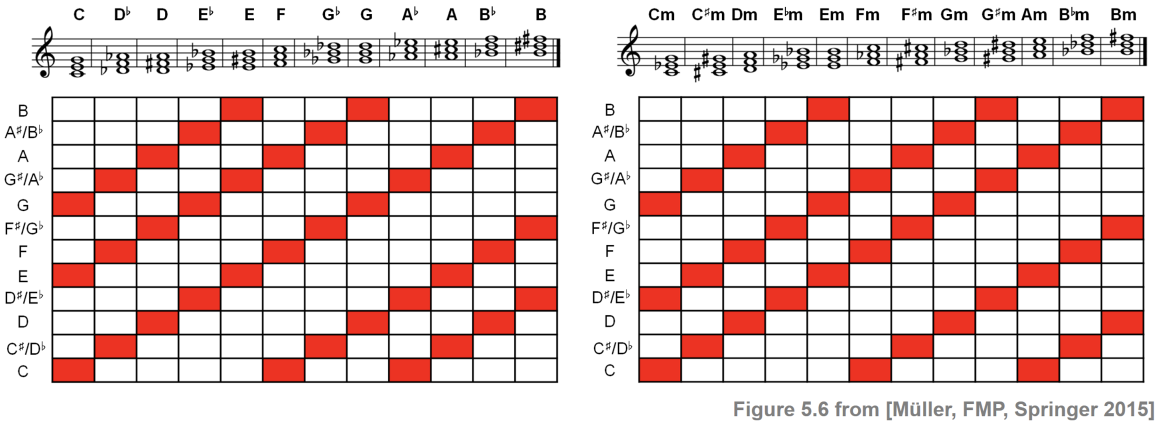


Figure 5.3. Pattern Matching [1]

5.3.2 Implementation

The following steps are performed and visualized in Template-based chord recognition:

1. First, the audio recording is converted into a chromagram representation. We use the STFT-variant.

2. Second, each chromagram vector is compared with each of the 24 binary chord templates, which yields 24 similarity values per segment. These similarity values are visualized in the form of a timechord representation.

3. Third, for each frame, there is a chord label λ_n of the template that addresses the similarity value over all 24 chord templates. This yields our final chord recognition result, which is shown in the form of a binary timechord representation.

4. Fourth, the manually generated chord annotations are visualized.

Figure 6.4 represents Template-based chord recognition results.

5.3.3 Hidden Markov Model (HMM)

”A Markov chain (MC) is useful when we need to compute a probability for a sequence of observable events. In many cases, however, the events we are interested in are hidden: we dont observe them directly. For example we dont normally observe the chord labels in a music audio

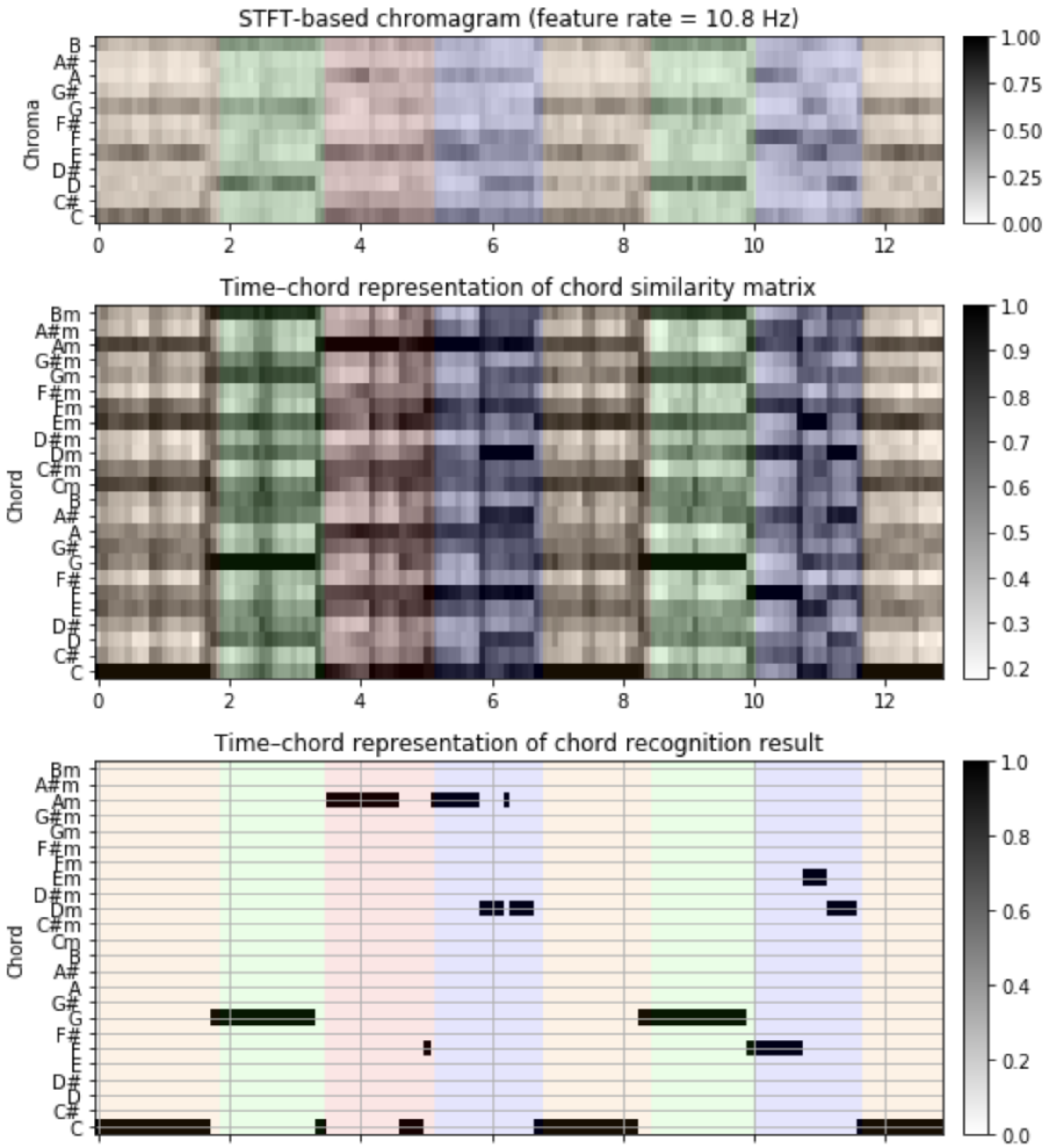


Figure 5.4. Template-based chord recognition results

signal” [45]. Rather, we see the audio sound and must infer the chords out of it. The sequence is called hidden because the elements has not yet been observed.

The HMM will provide an opportunity to add more features to our observation and keep with the same framework of that MC. In this chapter, we will cover the main intuitions of HMM in chord detection.

The main answer that a HMM can give us is:

What is the most probable sequence of chords for a given sequence of observations?

In order to answer this question we will need a few things from the MC and some new features:

- Chord Transition Probability Matrix: These are the notes probabilities explained in the MC section but having the chord transition probabilities instead.
- Emission probabilities: Probability of an observation to belong to each one of the chords $P(observation||chord)$.
- Initial State Probability Matrix: Indicates what is the probability of a sequence to begin with a specific chord.

5.3.4 Annotation of Music Data

In order to be able to generate the probabilities above, we need:

1. The music files in order to extract the chromagrams
2. An annotated dataset, so we can join the chord labels with the corresponding windowed chromagrams.

5.3.5 Calculate Framed Chromagram

Music data annotations provide the time period during which each chord was played in a particular piece of music. The idea is to create a definition of ”what is a C Major chord in a chromagram” so we can create the emission probabilities matrix.

```
x, Fs = librosa.load(let_it_be_intro_path)
let_it_be_chromagram = calc_chromagram(x, Fs, plot=False)
frames_per_sec, frame_duration_sec = get_frame_stats(chromagram=let_it_be_chromagram,
                                                     signal=x,
                                                     Fs=Fs)

pcp = chromagram_2_dataframe(let_it_be_chromagram, frame_duration_sec, test_version=True)

pcp['chord'] = get_annotated_chord_sequence(pcp, chords_annotation, test_version=True)
pcp.head()
```

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	start	end	chord
0	0.332066	0.384424	0.431916	0.414366	0.131415	0.091793	0.108283	0.152824	0.190040	0.224761	0.313280	0.371681	0.000000	0.046419	C
1	0.664177	0.352762	0.109753	0.110625	0.366219	0.163517	0.087432	0.182613	0.090595	0.068443	0.157639	0.412985	0.046419	0.092837	C
2	0.754787	0.184479	0.045408	0.124241	0.429377	0.168346	0.106199	0.206016	0.082103	0.026816	0.040539	0.321304	0.092837	0.139256	C
3	0.734501	0.146538	0.047722	0.111363	0.507049	0.167623	0.137559	0.243638	0.065114	0.015817	0.024087	0.236237	0.139256	0.185674	C
4	0.833918	0.198937	0.056337	0.078762	0.421176	0.114144	0.089199	0.182769	0.042916	0.006769	0.013399	0.147586	0.185674	0.232093	C

Figure 5.5. HMM annotated dataset

```
transition_matrix = calc_transition_prob_matrix(pcp, test_version=True)
transition_matrix
```

sequence_chord	A:min	C	F	G
initial_chords				
A:min	0.972222	0.000000	0.027778	0.000000
C	0.000000	0.980198	0.000000	0.019802
F	0.000000	0.028169	0.971831	0.000000
G	0.014493	0.000000	0.014493	0.971014

Figure 5.6. A representation of state transition probability matrix

Before merging our chromagram with the annotation files, we need to know how much time each chromagram windows takes in order to be able to merge. To calculate the framed chromagram we send the windowed chromagram, the signal, and its sampling frequency to the model in order to know how many seconds each window is.

5.3.6 Calculate State Transition Probability Matrix

To calculate the state transition probability matrix, the model runs through all chords in the dataframe and will count all of the possible chord-to-chord transitions. Finally, in order to turn the count values into probability values, we will normalise all values so the probabilities of going from one chord to all others is always 1. A portion of the representation of state transition probability matrix is shown in Figure 6.6.

5.3.7 Calculate Emission Probability Matrix

We calculated distribution assumptions of each chord according to the mean energy in the labeled chromagram. For the HMM model with M states, we generate a state array shape [M, 12], where 12 are the 12 notes from the chromagram. It will show the average energy in each one of the 12 notes for each chord. For each chord of the HMM model, we will have a matrix of shape [12, 12] for each note. It will tell the HMM how, in each chord, the notes vary among themselves. For example, in a C major chord, we expect when the C note is high, the notes E and G are expected to be high as well. In this condition, state covariance matrix is of shape [M , 12, 12]. Then the model repeatedly runs on every chord and their respective chromagrams to calculates the mean energy and their covariance.

5.3.8 Calculate Initial State Probability Matrix (ISPM)

The reason initial state probability is needed is because at the beginning of the model training, the model doesn't know the history of the chords played before. Therefore, an initial state probability matrix will start the estimation process by randomly estimating the beginning chord. For every step in the algorithm, we always calculate $P(chord_i||chord_{i-1})$, i.e., the probability of observing a chord in a window given the previous chord at the previous window.

In this case, in order to calculate our ISPM, we ran through all our audio annotated files, counted all the initial chords, i.e., first chord of the music except for silence, and then normalized the number so the sum of the ISPM = 1.

5.3.9 Implementation

To implement HMM model we used the hmmlearn python package []. This package abstract a few complicated mathematics, leaving an interface similar to sklearn machine learning packages, where we build and then predict over new observations.

- Because we are working with continuous emission probabilities, we build a hmm.GaussianHMM first and send the number of states. The covariance-type="full" defines that the HMM understands that the notes can have a relationship, i.e., the amount of energy in one note

```
COL_NAMES_NOTES = ["C", "C#", "D", "D#", "E", "F", "F#", "G", "G#", "A", "A#", "B"]

chord_ix_predictions = h_markov_model.predict(pcp[COL_NAMES_NOTES])
print('HMM output predictions:')
print(chord_ix_predictions[:50])

# create map between numerical state index and chord
chord_numbers = range(len(mu_array.index.values))
chords = mu_array.index.values
ix_2_chord = {ix_: chord_str for ix_,chord_str in zip(chord_numbers,chords)}

chord_str_predictions = get_hmm_predictions(chord_ix_predictions, ix_2_chord)
print('Translated chords HMM output predictions:')
print(chord_str_predictions[:50])

pcp['predicted'] = chord_str_predictions
```

```
HMM output predictions:
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 3 3 3 3 3 3 3 3 3 3 3]
Translated chords HMM output predictions:
['C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C'
 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C'
 'C' 'C' 'C' 'G' 'G' 'G' 'G' 'G' 'G' 'G' 'G' 'G' 'G' 'G' 'G']
```

Visual Evaluation

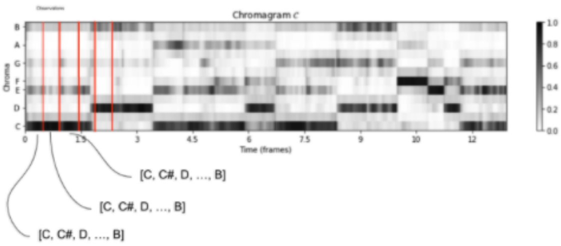


Figure 5.7. The results of HMM model

is not independent of anyone else. For different HMM with different types of emission probabilities.

- We set the Initial State Probability Matrix
- Then Transition Probability Matrix
- Means and Covariance indicates the two parts of our emission probability as explained in section ”Emission Probabilities”.

Figure 6.7 illustrates the result of HMM model tested on the first 10 second of the song ”Let-it-be”.

5.4 Proposed Framework of MOOCs for Music Assessment

It is essential to have an automated performance assessment for large-scale class sizes to provide quantitative metrics for a faster assessment and real-time feedback to each music learner.

The real-time assessment report is similar to the feedback provided by teachers in face-to-face classes. Feedback providing task involves extracting features from the student's audio and video performance to provide standard performance measures and quantitative metrics. Such methods would be more practical in music lessons since young generations learn faster and more efficiently when the provided report is instant and intuitive.

Since spectral modulation features via the modulation spectrum provide a simple visualization of rhythmic structures in music, these features can provide instant feedback for learning the rhythm of various simultaneous parts in a musical piece. In the next subsection, we focus on these features as an example to provide instant feedback for learning piano.

5.4.1 Modulation Spectral Features for Rhythmic Structures

Spectral Modulation features from the modulation spectrum can be practical in audio data mining tasks. In the music technology era, spectral modulation can address the classification and visualization of long-term and short-term rhythmic structures in music tempo and repeating patterns [46].

To demonstrate the potential of automatic assessment tools for this task, we plan on implementing a benchmark system that uses a well-known approach: assigning performance grades via mapping note level deviations computed from aligned transcriptions of the performance and the reference. Our benchmark system will be tested through a case study in Summer 2021 in a real-life scenario. The results of the study case will be presented at the conference.

Chapter 6

Conclusion and Future Work

Massive Online Courses (MOOCs) present a set of practical and thoughtful challenges that may not generally happen in smaller class sizes. This chapter addressed two primary aspects of online music learning where machine learning techniques can be applied to enhance online music lessons for diverse learning styles and audiences' verity.

We demonstrated our proposal for the automated development of lesson plans that can train music lovers to play their favorite instruments and simultaneously have access to automated unique learning styles, varied musical backgrounds, and/or skills.

Furthermore, this chapter proposed a quantitative assessment method of a student's progress in learning how to play a particular musical instrument. Proposed solutions in this chapter can be useful for both individual learners or as an instructor who may need to assess the quality of a massive number of students' performance. ”Previous research shows that interactive learning environments can also significantly contribute towards a student's interest, motivation, and discipline, and thereby enhance the commitment to learning” [43][47][30].

Future work would include the implementation of proposed methods and improving chord detection methods presented. In addition, accessibility methods of the online music teaching platform will be researched and implemented to support persons with disabilities such as visually impaired students.

References

[1] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015.

[2] G. Xia, "Expressive collaborative music performance via machine learning," 2016.

[3] C. Johnson, "Developing a teaching framework for online music courses," 2016.

[4] Y.-J. Lin, H.-K. Kao, Y.-C. Tseng, M. Tsai, and L. Su, "A human-computer duet system for music performance," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 772–780.

[5] I. E. Allen and J. Seaman, "Grade change: Tracking online education in the united states." *Babson Survey Research Group*, 2014.

[6] S. Wise, J. Greenwood, and N. Davis, "Teachers' use of digital technology in secondary music education: illustrations of changing classrooms," *British Journal of Music Education*, vol. 28, no. 2, pp. 117–134, 2011.

[7] B. Magerko, J. Freeman, T. McKlin, S. McCoid, T. Jenkins, and E. Livingston, "Tackling engagement in computing with computational music remixing," in *Proceeding of the 44th ACM technical symposium on Computer science education*, 2013, pp. 657–662.

[8] P. P. Apostolou and M. D. Avgerinou, "The coding maestros project: Blending steam and non-steam subjects through computational thinking," in *Handbook of Research on K-12 Blended and Virtual Learning Through the i^2Flex Classroom Model*. IGI Global, 2021, pp. 504–518.

[9] M. Kong, "Thinking like a computer: An exploratory study of introductory programmers'learning processes in scratch," Ph.D. dissertation, University of Delaware, 2020.

[10] M. Guzdial, "A media computation course for non-majors," in *Proceedings of the 8th annual conference on Innovation and technology in computer science education*, 2003, pp. 104–108.

[11] M. Guzdial, D. Ranum, B. Miller, B. Simon, B. Ericson, S. A. Rebelsky, J. Davis, K. Deepak, and D. Blank, "Variations on a theme: role of media in motivating computing education," in *Proceedings of the 41st ACM technical symposium on Computer science education*, 2010, pp. 66–67.

[12] Y. Kafai, C. Proctor, and D. Lui, "From theory bias to theory dialogue: embracing cognitive, situated, and critical framings of computational thinking in k-12 cs education," *ACM Inroads*, vol. 11, no. 1, pp. 44–53, 2020.

[13] J. Burg, J. Romney, and E. Schwartz, "Computer science" big ideas" play well in digital sound and music," in *Proceeding of the 44th ACM technical symposium on Computer science education*, 2013, pp. 663–668.

[14] A. L. Meyers, M. C. Cole, E. Korth, and S. Pluta, "Musicomputation: teaching computer science to teenage musicians," in *Proceedings of the seventh ACM conference on Creativity and cognition*, 2009, pp. 29–38.

[15] J. M. Heines, G. R. Greher, and S. A. Ruthmann, "Techniques at the intersection of computing and music," in *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*, 2012, pp. 372–372.

[16] R. Eglash, A. Bennett, C. O'donnell, S. Jennings, and M. Cintorino, "Culturally situated design tools: Ethnocomputing from field site to classroom," *American anthropologist*, vol. 108, no. 2, pp. 347–362, 2006.

[17] I. Peretz and R. J. Zatorre, "Brain organization for music processing," *Annu. Rev. Psychol.*, vol. 56, pp. 89–114, 2005.

[18] C. Kelleher, R. Pausch, and S. Kiesler, "Storytelling alice motivates middle school girls to learn computer programming," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, pp. 1455–1464.

[19] M. Muller, D. P. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE Journal of selected topics in signal processing*, vol. 5, no. 6, pp. 1088–1110, 2011.

[20] C. Uhle, C. Dittmar, and T. Sporer, "Extraction of drum tracks from polyphonic music using independent subspace analysis," in *Proc. ICA*, 2003, pp. 843–847.

[21] F. Soulez, X. Rodet, and D. Schwarz, "Improving polyphonic and poly-instrumental music to score alignment," 2003.

[22] N. Orio, S. Lemouton, and D. Schwarz, "Score following: State of the art and new developments," in *New Interfaces for Musical Expression (NIME)*, 2003.

[23] S. Siva, T. Im, T. McKlin, J. Freeman, and B. Magerko, "Using music to engage students in an introductory undergraduate programming course for non-majors," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018, pp. 975–980.

[24] J. D. Burge and T. L. Suarez, "Preliminary analysis of factors affecting women and african americans in the computing sciences," in *2005 Richard Tapia Celebration of Diversity in Computing Conference*. IEEE, 2005, pp. 53–56.

[25] J. Preston and B. Morrison, "Entertaining education–using games-based and service-oriented learning to improve stem education," in *Transactions on edutainment III*. Springer, 2009, pp. 70–81.

[26] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman *et al.*, "Scratch: programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009.

[27] J. Freeman, B. Magerko, T. McKlin, M. Reilly, J. Permar, C. Summers, and E. Fruchter, "Engaging underrepresented groups in high school introductory computing through computational remixing with earsketch," in *Proceedings of the 45th ACM technical symposium on Computer science education*, 2014, pp. 85–90.

[28] A. Ruthmann, J. M. Heines, G. R. Greher, P. Laidler, and C. Saulters, "Teaching computational thinking through musical live coding in scratch," in *Proceedings of the 41st ACM technical symposium on Computer science education*, 2010, pp. 351–355.

[29] J. M. Heines and D. A. Walzer, "Teaching a computer to sing (tacts): Integrating computing and music in a middle school, after-school program," *Journal of computing sciences in colleges*, vol. 33, no. 6, 2018.

[30] F. Jamshidi and D. Marghitu, "Using music to foster engagement in introductory computing courses," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1278. [Online]. Available: https://doi.org/10.1145/3287324.3293855

[31] C. N. Silla, A. L. Przybysz, A. Rivolli, T. Gimenez, C. Barroso, and J. Machado, "Girls, music and computer science," in *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2018, pp. 1–6.

[32] C. N. Silla, A. L. Przybysz, and W. V. Leal, "Music education meets computer science and engineering education," in *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2016, pp. 1–7.

[33] C. D. Hundhausen, N. H. Narayanan, and M. E. Crosby, "Exploring studio-based instructional models for computing education," in *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, 2008, pp. 392–396.

[34] D.-L. Priest and C. I. Karageorghis, "A qualitative investigation into the characteristics and effects of music accompanying exercise," *European physical education review*, vol. 14, no. 3, pp. 347–366, 2008.

[35] O.-E. Bolívar-Chávez, J. Paredes-Labra, Y.-V. Palma-García, and Y.-A. Mendieta-Torres, "Educational technologies and their application to music education: An action-research study in an ecuadorian university," *Mathematics*, vol. 9, no. 4, p. 412, 2021.

[36] A. R. Brown, D. Keller, and M. H. de Lima, "How ubiquitous technologies support ubiquitous music," *The oxford handbook of community music*, pp. 131–151, 2018.

[37] H. Partti, "Cosmopolitan musicianship under construction: Digital musicians illuminating emerging values in music education," *International Journal of Music Education*, vol. 32, no. 1, pp. 3–18, 2014.

[38] M. B. Miles, A. M. Huberman, J. Saldana *et al.*, "Qualitative data analysis: A methods sourcebook," 2014.

[39] J. W. Creswell, W. E. Hanson, V. L. Clark Plano, and A. Morales, "Qualitative research designs: Selection and implementation," *The counseling psychologist*, vol. 35, no. 2, pp. 236–264, 2007.

[40] T. Robison, "Male elementary general music teachers: A phenomenological study," *Journal of Music Teacher Education*, vol. 26, no. 2, pp. 77–89, 2017.

[41] M. Q. Patton, "Two decades of developments in qualitative inquiry: A personal, experiential perspective," *Qualitative social work*, vol. 1, no. 3, pp. 261–283, 2002.

[42] J. Schensul and M. LeCompte, *Essential Ethnographic Methods: A Mixed Methods Approach*, ser. Ethnographer's toolkit. AltaMira Press, 2013. [Online]. Available: https://books.google.com/books?id=iyvtr5_3mAoC

[43] B. Bozkurt, S. Gulati, O. Romani Picas, and X. Serra, "Musiccritic: a technological framework to support online music teaching for large audiences," in *Forrest D, editor. Proceedings of the International Society for Music Education. 33rd World Conference on Music Education (ISME); 2018 Jul 15-20; Baku, Azerbaijan. Malvern: International Society for Music Education; 2018.* International Society for Music Education, 2018.

[44] N. H. Sephus, T. O. Olubanjo, and D. V. Anderson, "Enhancing online music lessons with applications in automating self-learning tutorials and performance assessment," in *2013 12th International Conference on Machine Learning and Applications*, vol. 2. IEEE, 2013, pp. 568–571.

[45] J. H. M. Daniel Jurafsky, "Sequence labeling for parts of speech and named entities," in *Speech and Language Processing.*, 2020.

[46] N. H. Sephus, A. D. Lanterman, and D. V. Anderson, "Exploring frequency modulation features and resolution in the modulation spectrum," in *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*. IEEE, 2013, pp. 169–174.

[47] F. Jamshidi and D. Marghitu, "A web-based platform to teach music online."