2023

A multiple case study of high school perspectives making music with code in Sonic Pi

BOSTON UNIVERSITY

COLLEGE OF FINE ARTS

Dissertation

A MULTIPLE CASE STUDY OF HIGH SCHOOL PERSPECTIVES MAKING MUSIC WITH CODE IN SONIC PI

by

NATHANIEL STOTTLEMYER

B.M., The Catholic University of America, 2003
M.A., Liberty University, 2014

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Musical Arts

2023

Approved by

First Reader ____
Jared D. O'Leary, Ph.D.
Online Course Facilitator

Second Reader ____
Kelly Bylica, Ph.D.
Assistant Professor of Music, Music Education

Third Reader ____
André de Quadros, Ed.D.
Professor of Music, Music Education

Anything is possible; we're not limited by technology, we're not limited by the computer. We're limited only by our mind-sets.

- Don Buchla

DEDICATION

To my beautiful and brilliant wife Heather, and my joyful, gifted children, Grayson and Mona.

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to my dissertation supervisor, Dr. Jared O'Leary, for your expertise, encouragement, and patience. Your perspectives were thought-provoking and motivated me to work through the challenges I faced during this process. Thank you to my committee members, Dr. Kelly Bylica and Dr. André de Quadros for your critical feedback during the development of this dissertation. I would also like to thank members of the dedicated faculty at Boston University who helped shape this dissertation, including: Dr. Karin Hendricks, Dr. Tawnya Smith, Dr. Andrew Goodrich, Dr. Kinh Tiến Vũ, and Dr. Ronald Kos.

This dissertation would not have been possible without the cooperation of Collegium Charter School and its administration, especially Marita Barber and Kristin Gallahan — thank you. I would also like to thank my colleague, Brian Cotter, for your enthusiasm in my research and your friendship.

The following scholars have contributed to my understanding and research. Thank you, Dr. Ge Wang at Stanford University for your ideas about Artful Design and aesthetics, and insights into my work. Also, thank you, Dr. Jeff Snyder at Princeton University for your thoughts about electronic music and for your recommendations.

Thank you, Dr. Sam Aaron for creating Sonic Pi and providing an accessible way to make music for everyone. I believe Sonic Pi is a modern realization of Seymour Papert's vision for music.

The following musicians have inspired me on my musical journey. Thank you, cEvin Key, Rhys Fulber, Alessandro Cortini, and Tom Jenkinson for your music. Your musics drove me forward, solaced me when I was in need, and influenced my musical works.

Finally, I am extremely grateful for the love and support of my family. Heather, thank you for your patience and for lifting me up when I was met with challenges. Your wisdom is a pillar of strength. Grayson, thank you for checking in on me as I worked on this dissertation. Your thoughtfulness has prompted me to think more of others. Mona, thank you for asking me questions about my ideas. Your inquisitiveness inspires me to ask more questions. Heather, Grayson, and Mona — I love you.

A MULTIPLE CASE STUDY OF HIGH SCHOOL PERSPECTIVES MAKING MUSIC WITH CODE IN SONIC PI

NATHANIEL STOTTLEMYER

Boston University College of Fine Arts, 2023

Major Professor: Jared D. O'Leary, Ph.D., Online Course Facilitator

ABSTRACT

The purpose of this study was to investigate perceptions of high school students who made music with code in Sonic Pi. This qualitative multiple case study focused on individuals in an extracurricular club at a public charter high school who volunteered to participate on-site and remotely asynchronously via Canvas learning management system. This study was guided by five research questions, including: (1) What musical ideas, if any, do participants report learning or demonstrate through making music with code in Sonic Pi? (2) How does making music with code impact participants' perceptions of their music making? (3) How does making music with code impact participants' perceptions of their ability to learn to make music? (4) How does making music with code impact participants' interest in music courses? (5) How does making music with code impact participants' interest in computer science courses? Participants completed research study materials, including a series of tutorials for Sonic Pi. Data included answers to questionnaires and surveys, multimedia artifacts including the source code and exported audio of participants' music making, and interviews of participants that were codified and analyzed in two cycles, utilizing descriptive coding, values coding, and longitudinal coding. Participants' code and multimedia artifacts revealed a close alignment to the four properties of sound, including: pitch, duration, intensity/amplitude, and timbre. Participants' artifacts revealed themes and demonstrated ideas extending beyond the four properties, including: form, non-traditional music notation, and randomization. Participants all agreed their coded artifacts are music. Additionally, participants' varied responses about musicianship and composers suggests that making music is something anyone can engage in, regardless of how one identifies themself. All participants agreed that Sonic Pi is a useful tool for learning and understanding musical concepts and that Western staff notation is not required knowledge for making music. Participants' interests in music or computer science courses were impacted by their prior experiences in music and/or coding. This study concludes with a discussion of themes based on the findings.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

ADSR. An acronym for *attack, decay, sustain, and release.* An ADSR is an example of an envelope.

API. An acronym for *application programming interface*. The way in which two computer applications interact, as opposed to *user interface*, which is the way in which a user interacts with a computer. In computer science, an API may be a library of instructions pre-built for a programming language.

ASSEMBLY LANGUAGE. Any low-level programming language containing instructions that correspond to machine code. Assembly language is specific to computer architecture. For example, assembly language was used to program the PSG inside the Commodore 64 personal computer.

BASIC. An acronym for *beginners' all-purpose symbolic instruction code*. A family of high-level programming languages developed for new coders. BASIC was used to program devices like the PSG inside the Nintendo Entertainment System.

CONCRETIZE. (As it relates to constructionism) To model what are otherwise abstract ideas.

CONSTRUCTIONISM. A theory of education developed by Seymour Papert emphasizing that learners construct knowledge for themselves through creativity, experiences, and problem solving.

CREATIVE LEARNING SPIRAL. A procedure for creativity and learning ideated by Mitchel Resnick in which learners iterate several cycles of imaging, creating, playing, sharing, and reflecting.

CS. An acronym for *computer science*. Computer science is the study of computers and their uses.

DAW. An acronym for *digital audio workstation*. A DAW is a software application with a graphical user interface for recording, editing, and producing music or sound utilizing various audio formats and MIDI data.

ENVELOPE. A programmed control of signal amplitude over a course of time.

FUNCTION. A function is a procedure in computer programming that performs a task by either passing information through its input(s), known as parameters or arguments, or by calling information from other parts of the program.

IDE. An acronym for *integrated development environment*. A software application for developing software. An IDE typically includes a code editor, language tools, and a debugger.

LEGO. A toy consisting of interlocking plastic pieces. Lego pieces can be used to build creative projects and can be given movement through electronic parts that are programmed by a user.

LIVE CODING. The ability to edit code while a program is running, followed by re-running the code for changes to take place. For example, live coding music. (See also ON-THE-FLY PROGRAMMING)

`live_loop`. A built-in function in Sonic Pi for continuous iteration of code. The `live_loop` function can run concurrently with other instances of `live_loop`.

LMS. An acronym for *learning management system*. A software application for the administration, recording, and reporting of educational or learning materials and activities.

LO. An acronym for *learning objective*. A learning objective is a target one aims to achieve through a learning experience.

LOGIC. A system for the arrangement of elements in a digital device to perform a specific task.

LOGIC ERROR. An unexpected result when running code.

LOGO. A programming language developed by Seymour Papert to enable learners to make concrete representations of abstract ideas using a computer.

MIDI. An acronym for *musical instrument digital interface*. MIDI is a communication standard that facilitates the sending and receiving of pitch, duration, and velocity information. MIDI can be used to control one or several MIDI-enabled instruments.

MUSIC N. A family of music programming languages. Music N programming languages laid the foundation for modern live coding/on-the-fly programming languages.

ON-THE-FLY PROGRAMMING. The ability to edit code while it is running, followed by re-running the code for changes to take place. (See also LIVE CODING)

PSG. An acronym for *programmable sound generator*. PSGs are integrated circuits capable of producing different waveforms and envelopes that were often used in retro computing devices like gaming consoles.

SCRATCH. Scratch is a high-level, block-based programming language that was developed to help young learners to express themselves through programming.

SMF. An acronym for *Standard MIDI Files* specification. SMF standardized the transfer of MIDI data between systems.

SONIC PI. A live coding/on-the-fly programming environment based on the Ruby programming language. Sonic Pi is designed to help users create live or precomposed music through codes.

STEM. An acronym for *science, technology, engineering, mathematics*. STEM is often used to describe the interdisciplinary relationship between science, technology, engineering, and mathematics.

SYNTAX. The rules governing the arrangement of elements in a programming language.

SYNTAX ERROR. The prevention of a program running due to erroneous syntax entry.

VIRTUAL INSTRUMENT. A software musical instrument with an integrated graphical user interface.

WESTERN STAFF NOTATION. A system for visually documenting music that developed in Europe. It is the standard notation format for many Western classical musics.

CHAPTER 1: INTRODUCTION

Music education in the United States has a long tradition of performance, emphasizing singing and the playing of instruments.[1] With the advent of computers and synthesizers during the 20th century, music technology emerged as an educational topic.[2] Music technology, however, is often not a major focus of music education or is met with resistance by many music educators.[3] Meanwhile, making music has become a popular tool to introduce coding in computer science education contexts.[4] The merging of music technology and computer science has

---

1. David Beckstead, "Will Technology Transform Music Education?," *Music Educators Journal* 87, no. 6 (2001): 45; David Brian Williams, "Reaching the 'Other 80%:' Using Technology to Engage 'Non-Traditional Music Students' in Creative Activities," in *Tanglewood II "Technology and Music Education" Symposium* (University of Minnesota, 2007); Herbert A Deutsch, "Where Was Technology and Music Education Twenty Years Ago?," *Journal of Popular Music Studies* 21, no. 1 (2009): 93, https://doi.org/10.1111/j.1533-1598.2009.01171.x; Josh A Bula, "Technology-Based Music Courses and Non-Traditional Music Students in Secondary Schools" (ProQuest Dissertations Publishing, 2011), 1–2.

2. Jerome N Margolis, "A School Synthesizer Program Comes of Age," *Music Educators Journal* 74, no. 4 (1987): 33, https://doi.org/10.2307/3397960; Deutsch, "Where Was Technology and Music Education Twenty Years Ago?," 94.

3. Deutsch, "Where Was Technology and Music Education Twenty Years Ago?," 95.

4. Jennifer Burg, Susan Reiser, and Wayne Kirby, "Hot Fusion: Music, Art, and Computer Science Education," *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 2013, 313, https://doi.org/10.1145/2462476.2483798; Scott McCoid et al., "EarSketch: An

resulted in a popular trend using coding to create novel music.[5] However, the integration of music in computer science curriculums often situates music in a subservient role.

Research Problem

Support for computer science education continues to gain momentum in the United States. Improved access to technology and learning resources for coding is facilitating growth in computer science education.[6] Companies including Code.org, CodeHS, and Codecademy are developing free and paid resources for individuals and classrooms, and they are lobbying for the importance of computer science education in schools with endorsements by high-profile personalities and media outlets.[7] As a result, computer science is often prioritized in schools

---

Integrated Approach to Teaching Introductory Computer Music," *Org. Sound* 18, no. 2 (2013): 146, https://doi.org/10.1017/S135577181300006X.

5. Thor Magnusson, "Algorithms as Scores: Coding Live Music," *Leonardo Music Journal* 21 (2011): 19, https://doi.org/10.1162/LMJ_a_00056; Ge Wang, Perry R Cook, and Spencer Salazar, "ChucK: A Strongly Timed Computer Music Language," *Computer Music Journal* 39, no. 4 (2015): 10, https://doi.org/10.1162/COMJa00324; Alexandra Cardenas, "Live Coding: A New Approach to Musical Composition," *Dancecult* 10, no. 1 (2018), https://doi.org/10.12801/1947-5403.2018.10.01.14.

6. Annette Vee, *Coding Literacy: How Computer Programming Is Changing Writing* (Cambridge: Cambridge: MIT Press, 2017), 5.

7. Kevin Brooks and Chris Lindgren, "Responding to the Coding Crisis: From Code Year to Computational Literacy," in *Strategic Discourse: The Politics*

because "teaching programming is an increasingly important aspect of our curriculum."[8]

Coding is often emphasized as a single skill that will lead to high-paying jobs.[9] However, learning to code is applicable to a broader range of ideas and not just useful "for high-profile creative or business applications but also for organizing personal information, analyzing literature, publishing creative projects, interfacing with government data, and even simply participating in society."[10] Computer science pedagogues and instructional designers often use making music with code to engage learners in computer science education because the integration of music and computer science has increased student motivation and positive attitudes toward learning coding.[11]

---

of (New) Literacy Crises, edited by Lynn C. Lewis. Computers and Composition Digital Press, 2015: 1-2. http://hdl.handle.net/10919/96777.

8. Sam Aaron, "Live Coding Education," *The MagPi Educator's Edition*, Education Special Issue 2, 44, https://www.raspberrypi.org/magpi-issues/MagPi-EduEdition02.pdf.

9. Kevin Brooks and Chris Lindgren, "Responding to the Coding Crisis: From Code Year to Computational Literacy," in *Strategic Discourse: The Politics of (New) Literacy Crises*, ed. Lynn C. Lewis (Computers and Composition Digital Press, 2015), http://hdl.handle.net/10919/96777, 5.

10. Vee, *Coding Literacy: How Computer Programming Is Changing Writing,* 5.

11. Brian Magerko, Jason Freeman, Tom Mcklin, Mike Reilly, Elise Livingston, Scott Mccoid, and Andrea Crews-Brown, "EarSketch: A STEAM-

Making music with code is being used successfully to engage students in computer science education. Several platforms have emerged aimed at teaching coding through music, including: EarSketch, TunePad, JythonMusic, and Sonic Pi. EarSketch is intended to engage learners "in computing concepts through computational music remixing."[12] On TunePad's website, users are encouraged to "Learn to code by creating tunes that sound great."[13] JythonMusic places emphasis on "creative exploration" of programming rather than the rules and procedures of the programming language because focusing on creativity attracts non-traditional students (including artists and musicians) to computer science.[14]

Based Approach for Underrepresented Populations in High School Computer Science Education," *ACM Transactions on Computing Education (TOCE)* 16, no. 4 (2016): 2, https://doi.org/10.1145/2886418; Lee Cheng, "Teaching Live Coding of Electronic Dance Music: A Case Study," *Dancecult* 10, no. 1 (2018), https://doi.org/10.12801/1947-5403.2018.10.01.10; Christopher Petri, "Programming Music with Sonic Pi Promotes Positive Attitudes for Beginners," *Computers & Education* 179, no. March 2021 (2021): 1, https://doi.org/10.1016/j.compedu.2021.104409.

12. Magerko et. al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 2.

13. Michael S. Horn, Amartya Banerjee, and Melanie West, "Music and Coding as an Approach to a Broad-Based Computational Literacy," *Non-Formal and Informal Science Learning in the ICT Era*, edited by Michail Giannakos, 88 (Singapore: Springer Singapore, 2020), https://doi.org/10.1007/978-981-15-6747-6_5.

14. Bill Manaris, Blake Stevens, and Andrew R Brown, "JythonMusic: An Environment for Teaching Algorithmic Music Composition, Dynamic Coding and

In Sonic Pi, music and live coding is used to motivate learners, and while the original instructional focus of Sonic Pi lessons were related to computer science, Sonic Pi has facilitated opportunities for new music and new music learning practices.[15] Interesting trends have been observed of learners coding music with Sonic Pi, including strong correlations between coding and music, with increased engagement and enjoyment learning to code with music.[16] This presents an opportunity to investigate the ways in which making music with code engages students in music education. However, most research studies investigating making music with code spotlight the benefits to computer science education, often without investigating or mentioning any benefits to music education. I use the term *music for computer science transaction* to refer to the positioning of music educators and/or music education, whether intentional or unintentional, in service of computer science education through an interdisciplinary collaboration

---

Musical Performativity," *Journal of Music, Technology, and Education* 9, no. 1 (2016): 35. http://dx.doi.org/10.1386/jmte.9.1.33_1.

15. Samuel Aaron, Alan F Blackwell, and Pamela Burnard, "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," *Journal of Music, Technology, and Education* 9, no. 1 (2016): 86, 89. https://doi.org/10.1386/jmte.9.1.75_1.

16. Petri, "Programming Music with Sonic Pi Promotes Positive Attitudes for Beginners," 6.

that epitomizes academic inequality.[17] Music is intentionally integrated with coding to "alleviate the problem of beginner students viewing [coding] as difficult."[18] However, coding appears to be noticeably absent from music education, even though making music with code can "provide new pathways for young people into digital music."[19]

Several studies have demonstrated learners' engagement is heightened in computer science when making music with code, so it is unclear why "coding practices tend to be atypical forms of music engagement for most school music education curricula or programs."[20] One explanation may be that advancements in music technology now call into question the meaning of composing,

---

17. Jared O'Leary, "Intersections of Popular Musicianship and Computer Science Practices," *Journal of Popular Music Education* 4, no. 2 (2020): 169. https://doi.org/10.1386/jpme_00023_1.

18. Petri, "Programming Music with Sonic Pi Promotes Positive Attitudes for Beginners," 1.

19. Mark Gibbs, "Sonic Pi: Realtime Music Creation for the Raspberry Pi (and More)," *Network World Online* (2016): 2. https://www.networkworld.com/article/3053644/sonic-pi-realtime-music-creation-for-the-raspberry-pi-and-more.html

20. O'Leary, "Intersections of Popular Musicianship and Computer Science Practices," 158.

performing, and listening.[21] An example of this might include live coding, during which the musician writes and runs musical code for a live audience (both in-person and virtually) and alters the code live to perform a unique musical experience.[22] The concurrency of composing and performing is inconsistent with the traditional music paradigm. Typically, the process of composing and performing are sequential; first a piece of music is composed and then it is performed for an audience of listeners. In live coding, the coder composes and performs at the same time, while responding to listeners of their compositions. Another explanation may be that the composing, performing, and listening of classical music is commonly regarded as more beneficial to students' musical growth than popular music.[23] This notion is being challenged by proponents for teaching popular music asserting that popular music has greater potential to bridge connections to youth than traditional music offerings, and that popular music offers opportunities to introduce informal music making to the classroom

---

21. Tim Cain, "Theory, Technology and the Music Curriculum," *British Journal of Music Education* 21, no. 2 (2004): 217. https://doi.org/10.1017/S0265051704005650.

22. O'Leary, "Intersections of Popular Musicianship and Computer Science Practices," 166.

23. Deutsch, "Where was Technology and Music Education Twenty Years Ago?," 93.

environment.[24] These uncertainties about making music with code may contribute to the underutilization of making music with code in high school classrooms. This leads educators at the intersection of music and computer science to ask, if music is an engaging way to teach computer science, is computer science an engaging way to teach music?

Purpose Statement

The purpose of this study is to investigate perceptions of high school students who made music with code in Sonic Pi.

Research Questions

This study is guided by the following research questions:

1. What musical ideas, if any, do participants report learning or demonstrate through making music with code in Sonic Pi?
2. How does making music with code impact participants' perceptions of their music making?

24. Patricia Shehan Campbell, "Adolescents' Expressed Meanings of Music in and out of School," *Journal of Research in Music Education* 55, no. 3 (2007): 222. http://www.jstor.org/stable/4543122; Teresa R. Nielsen, "Teen Playlist: Music Discovery, Production, and Sharing Among a Group of High School Students," DMA diss., (Boston University, 2016): 40. http://hdl.handle.net/2144/19561.

3. How does making music with code impact participants' perceptions of their ability to learn to make music?
4. How does making music with code impact participants' interest in music courses?
5. How does making music with code impact participants' interest in computer science courses?

Scope of the Study

Study participants completed a series of Sonic Pi tutorials remotely and asynchronously in a Canvas learning management system (LMS) course on their personal devices. Prior to completing the tutorials, participants answered an entrance questionnaire asking about their experiences with making music, experiences with computer science, and demographic information. While progressing through the tutorials, participants submitted artifacts of their learning, including code samples, in Canvas. Participants concluded the study by submitting a final cumulative project in the form of a newly created piece of music and answered an exit survey. Participants were then invited to participate in an interview about their experience and perspectives on making music with code.

Sonic Pi

Several programming languages and environments exist for music. The exploration of computer programming as a compositional medium began in 1957 with MUSIC I.[25] The MUSIC-N family of programming languages influenced several subsequent languages, including: Csound, Max, Pure Data, SuperCollider, ChucK, and others. The Sonic Pi application was developed to address barriers related to readability and language syntax, and to quickly engage students in writing code through music.[26] The Sonic Pi application addresses this need as an integrated development environment (IDE) for music and audio software development using the Sonic Pi application programming interface (API), a library of routines for the Ruby programming language that controls the SuperCollider Server, Sonic Pi's audio engine. The advantage of Sonic Pi is its very readable, with high-level syntax, integrated tutorial materials, and reference documentation which are accessible alongside the IDE. In other words, the simplicity of the programming language and built-in support make it easy for

---

25. Wang, Ge. "The ChucK Audio Programming Language. 'A Strongly-Timed and On-the-Fly Environ/Mentality.'" PhD diss., (Princeton University, 2008) 11.

26. Aaron et. al., "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 79.

novice programmers to create music with code.

Sonic Pi was chosen for this study for its simplicity and ease. Sonic Pi's unified interface is all that is needed to start making music with code. The software is multiplatform and runs on Windows, macOS, and several distributions of Linux. Once installed, Sonic Pi assets are local and do not require an internet connection. If used with the Raspberry Pi, a tiny, affordable computer, the user will enjoy the added benefit of a low-cost music coding solution.

Computers and digital technologies can make music more accessible to learners by establishing a "pathway that did not previously exist," leveraging learners' diverse experiences, and presenting options for making music that embraces their musical preferences.[27] With code, making music is doable without learning music theory and without an understanding of Western staff notation, harmony, melody, or chord progressions.[28] Many people use digital devices every day for a variety of tasks. Musical code can transform these devices people are

---

27. Beckstead, "Will Technology Transform Music Education?," 47; Nielsen, "Teen Playlist: Music Discovery, Production, and Sharing Among a Group of High School Students," 39.

28. Beckstead, "Will Technology Transform Music Education?," 46; Magerko et. al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 14.

using daily into instruments and composing tools. With minimal effort, and with few lines of code, anyone can code music, especially with Sonic Pi.[29] There may even be technologies in the future that assist music makers, like generative artificial intelligence that use descriptive language in the form of prompts to make music. Participants in this study who have prior musical training may leverage their knowledge in unique meaningful ways that are atypical of traditional in-school experiences.

Remote Learning

Remote learning using Canvas LMS granted high school participants access to this study from any location. Furthermore, this approach allowed participants to work at their own pace and during times that were convenient for them without the limitations of synchronous sessions. Participants were given a six-month window to complete the research materials, however the approximate duration of active participation was approximately five hours for the majority of

---

29. Aaron et. al., "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 89; Sam Aaron, "Live Coding Education," *The MagPi Educator's Edition*, n.d., 47; "Sam Aaron's Sonic Pi Can Help You Make Music Through Code," GOTOpia, March 6, 2023, https://gotopia.tech/articles/220/Sam-Aaron-SonicPi-can-help-you-make-music-through-code; Sam Aaron, *Code Music with Sonic Pi: Live Code & Create Amazing Sound on Your Raspberry Pi*, v1 ed. (The MagPi, n.d.), 5.

participants. Participants represented varied levels of experience in music and computer science. This study demonstrated notable trends that will interest researchers and educators in music education and computer science education.

Concretizing Music

Creating music with code often occurs in informal settings and is a practice that developed outside the tradition of American K–12 music education.[30] The limited number of studies investigating computer science as a means to teach music are often framed in Papert's constructionism, and emphasize play with concrete examples of musical code that are provided by a curriculum, teacher, or facilitator.[31] In constructionism, learning takes place while creating something new (concretizing) from materials sourced from a surrounding

30. Campbell, "Adolescents' Expressed Meanings of Music in and out of School," 221; Bula, "Technology-Based Music Courses and Non-Traditional Music Students in Secondary Schools," 14; Jared O'Leary, "A Corpus-Assisted Discourse Analysis of Chiptune-Related Practices Discussed within Chipmusic.Org," PhD diss., (Arizona State University, 2019) 205. https://doi.org/10.1145/3287324.3293706.

31. Oliver Hancock, "Play-Based, Constructionist Learning of Pure Data: A Case Study," *Journal of Music, Technology, and Education* 7, no. 1 (2014): 94. https://doi.org/10.1386/jmte.7.1.93_1; Nick Collins, "Live Coding and Teaching SuperCollider," *Journal of Music, Technology, and Education* 9, no. 1 (2016): 7. https://doi.org/10.1386/jmte.9.1.5_1; Petri, "Programming Music with Sonic Pi Promotes Positive Attitudes for Beginners," 4.

environment.[32] Constructionism blurs the distinctions made in Piaget's constructivist concrete and formal operational thinking.[33] Mitchel Resnick clarifies this distinction in his foreword to *Mindstorms*:

> Piaget's great insight was that knowledge is not delivered from teacher to learner; rather, children are constantly constructing knowledge through their everyday interactions with people and objects around them. Seymour's constructionism theory adds a second type of construction, arguing that children construct knowledge most effectively when they are actively engaged in constructing things in the world. As children construct things in the world, they construct new ideas and theories in their minds, which motivates them to construct new things in the world, and on and on.[34]

Rather than transition from concrete operational thinking to formal operational thinking, as in constructivism, formal operational thinking is concretized in constructionism, meaning that the learner creates or makes objects to solve hypothetical or abstract problems and tasks.[35] In constructionism, the computer is noted as an important tool to concretize formal operational thinking, to

---

32. Hancock, "Play-Based, Constructionist Learning of Pure Data: A Case Study," 94; Seymour Papert, *Mindstorms: Children, Computers, and Powerful Ideas* (New York: Basic Books, 1980) 435, 636, Kindle.

33. Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 464.

34. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 77.

35. Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 464; Penny Thompson, "Cognitive Development: The Theory of Jean Piaget," in *Foundations of Educational Technology* (Oklahoma State University Libraries, 2019) np. https://doi.org/10.22488/okstate.19.000002.

facilitate the passage from child to adult thinking.[36] For these reasons, constructionism is used to frame the concretizing of musical engagement through computer science. In this study, participants engage in core musical activities of composition and listening using Sonic Pi.[37]

To concretize music with computer science, participants make music using Sonic Pi. Makers create new things using their hands and knowledge to solve problems or make tasks easier to do by sourcing materials available to them.[38] In an educational sense, making is an ideal way to concretize music. Participants in this study engaged in electronic learning resources and tutorials to make music with Sonic Pi. The emphasis of play with concrete examples in constructionism often includes the misconception that learning is spontaneous or that the learners

---

36. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 464.

37. Keith Swanwick, *A Basis for Music Education* (London: Routledge, 1979); Cain, "Theory, Technology and the Music Curriculum,", 217; Steven C Dillon, "The Student as Maker: An Examination of the Meaning of Music to Students in a School and the Ways in Which We Give Access to Meaningful Music Education." (Institute for Education LaTrobe University, 2001), https://eprints.qut.edu.au/17077/ 15; Steven Dillon, *Music, Meaning and Transformation: Meaningful Music Making for Life* (United Kingdom: Cambridge Scholars Publishing, 2007), https://eprints.qut.edu.au/6703/, 88-89, 96.

38. Alayna Hughes, "Maker Music: Incorporating the Maker and Hacker Community into Music Technology Education," *Journal of Music, Technology, and Education* 11, no. 3 (2018): 288. http://dx.doi.org/10.1386/jmte.11.3.287_1

"do whatever."[39] The tutorials and learning resources provided were intended to support participants' concretization of music that expressed their interests and perhaps, passion.[40]

Making music is multifaceted and "needs to be experienced and reflected upon through the musical processes of composition or creative making, performance and realisation, and active analysis of and listening to music."[41] Cain argued that digital computing makes these activity classifications problematic, as mentioned earlier, which is true if the computer is viewed as a device that simply puts learners through their paces.[42] However, by concretizing music using Sonic Pi, participants in this study engaged in these activities in novel ways, sometimes concurrently.

Composition or creative making in Sonic Pi is done using code. The

39. Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 636; Jared O'Leary, *#CSK8 Podcast*, 106, "Lifelong Kindergarten with Mitch Resnick," interview by Jared O'Leary, aired October 25, 2021, on Jared O'Leary //Multiplicity, https://jaredoleary.com/csk8feed/106.

40. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 636; O'Leary, "Lifelong Kindergarten with Mitch Resnick."

41. Dillon, "The Student as Maker: An Examination of the Meaning of Music to Students in a School and the Ways in Which We Give Access to Meaningful Music Education," 23.

42. Cain, "Theory, Technology and the Music Curriculum," 217; Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 587.

participants in this study used code as musical notation for their compositions/creations. Code is used to write algorithms, which make up the logical sequence of the music, and the final compiled programs are their musical compositions/creations. Performance and realization occur when the participants run their code. Participants in this study realized their music by design. Active analysis and listening are how the participant assessed their code/creation. In programming, active analysis includes debugging, listening particularly for logic errors (undesired results in a program). By concurrently engaging in core activities of making music using digital computing and code, participants concretized musical learning by creating in novel ways.

Positionality

My experiences with music growing up existed both inside school and outside of school. The music I learned in school included rote recitation/performance of folk and patriotic songs in elementary school, and later playing percussion in several performance ensembles throughout high school. I continued formal studies in music at the undergraduate and graduate levels. Outside of school, I was active in my church music ministry, led by my parents. I also studied piano and was interested in synthesizers and computers. I explored electronic music by attempting to program the infamous FM synthesis interface of my Yamaha DX100, sequencing MIDI patterns on my Yamaha QX21, and recording MIDI using Commodore 64, Macintosh, and Windows computers. Later, my interests in modular synthesizers fueled my curiosity of microcontrollers and embedded systems, and I then went on to explore do-it-yourself (DIY) making of circuits including Eurorack-format synthesizer modules. After developing a familiarity with Arduino programming syntax and control structures I moved on to the programming languages C/C++, and later to coding music with ChucK and Sonic Pi/Ruby to which I recognized the round-robin/superloop architecture from embedded systems. I also learned music programming with the graphical programming languages Max and Pure Data. These experiences profoundly shaped the educator I am today.

The majority of my career in education has been spent in non-performance-oriented programs at the middle school and high school levels as both a music teacher and computer science teacher. I have been fortunate to teach students who love music, and I have facilitated opportunities for them to create music using computers as their instruments. This included teaching electronic music production using MIDI and virtual instruments inside a digital audio workstation (DAW), musical outputs with microcontrollers (for example, a doorbell), algorithmic composition in EarSketch, live-coding dynamic compositions in Sonic Pi, and designing custom instruments in Pure Data. I have observed my own students enjoy making music with code, and for those students who were new to making music, code offered them the opportunity to be musically creative despite not having traditional Western musical training.

Having a breadth of classroom experience teaching music and computer science, I am positioned with a unique perspective to research making music with code. My observations of students making music with code is consistent with findings of studies engaging in the *music for computer science transaction.*[43] As a maker of music, I am concerned that music education in the United States is missing an incredible opportunity to introduce novel methods of composing, performing, and listening to music with code. I have been privileged to collaborate with in-service music educators and mentor pre-service educators on lesson development in music technology that incorporates making music with code. Findings from this study may inform the course of future lessons and curriculum development in music and computer science.

---

43. Magerko et. al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 2; Horn et. al., "Music and Coding as an Approach to a Broad-Based Computational Literacy," 88; Manaris et. al., "JythonMusic: An Environment for Teaching Algorithmic Music Composition, Dynamic Coding and Musical Performativity," 35; Aaron et. al., "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 86.

Roadmap

The chapters following this introduction further explore the discussed ideas and methods. Chapter 2 is a review of the literature relevant to the study, including: constructionism, algorithmic composition, and music and computer science education. Chapter 3 is an outline of the methods I used to investigate the problem, including modifications that were necessary to collect sufficient data. In Chapter 4, I summarize the procedures experienced by participants as they progressed through the study materials. In Chapter 5, I summarize the findings of the studies. Finally, in Chapter 6, I discuss the findings demonstrated by the study and implications for music education and computer science education pedagogy.

CHAPTER 2: REVIEW OF LITERATURE

In this review of literature, I explore the research on learning music and computer science with a focus on constructionism, algorithmic composition (specifically computer music and live coding), and music and computer science education. An exploration of Papert's constructionism provides an understanding of how learning is constructed by learners through concrete representations of knowledge. I selected literature that relates to the intersections of music and computer science, with a particular focus on the development of music coding technologies, their impact on music creativity, and potential for interdisciplinary learning.

The literature in this review provided context to my research and direction in my data analysis. The literature is organized into three sections, including: Constructionism, Algorithmic Composition, and Music and Computer Science Education. I conclude this chapter with a summary of issues and considerations.

Constructionism

Constructionism is a theory of education that learning takes place by constructing new knowledge through the creation of concrete things a learner finds in their environment.[44] Learning, as defined in constructionism, is facilitated

44. Mitchel Resnick, "Technologies for Lifelong Kindergarten," *Educational Technology Research and Development* 46, no. 4 (1998): 45,

by concretizing abstract ideas, often through the use of technology, especially the computer.[45] Technology is best utilized when it can provide a low floor (easy access to knowledge), high ceiling (high potential for growth), and wide walls (several pathways to grow).[46] Coding is often viewed as an effective way to concretize abstract ideas because it potentially has a low floor, high ceiling, and wide walls for learners to construct new knowledge.[47]

A Revolution of Ideas

Papert's conception of constructionism is an effort to develop a theory about education that builds upon Piaget's constructivism, which is a theory about learning.[48] Piaget points out that "knowledge is not delivered from teacher to learner," through what is often regarded as instruction; rather, learners construct knowledge "through their everyday interactions with people and

---

https://doi.org/10.1007/BF02299672.

45. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 241.

46. Mitchel Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play* (Cambridge, MA: The MIT Press, 2017), locs. 908–909.

47. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 461.

48. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 461.

objects."[49] Falbel stated, "[Papert] wanted to use what Piaget learned about children as a basis for rethinking education. He wanted to use Piaget's theory of knowledge to form a theory of education."[50] Papert argued that the role of educators is to facilitate, to provide opportunities for learners to construct knowledge by furnishing them with an environment filled with things to construct new things. He stated that "children appropriate to their own use materials they find about them, most saliently the models and metaphors suggested by the surrounding culture."[51] In Papert's view, the ideal classroom is a learning environment guided by a facilitator rather than an instructor. Resnick added to this the idea that classrooms should have multiple materials and resources for learners to construct knowledge, much like a kindergarten classroom. Resnick suggested, "Instead of making kindergarten like the rest of school, we need to make the rest of school (indeed, the rest of life) more like kindergarten."[52] For both Papert and Resnick, the computer and other digital devices were key

49. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 76–77.

50. Aaron Falbel, "Constructionism: Tools to Build (and Think) With," *LEGO Dacta*, n.d., 2.

51. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 435.

52. Mitchel Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 2007, 1, https://doi.org/10.1145/1254960.1254961.

resources for concretizing subject matter that has long been considered abstract. Computers are an important tool for concretizing the abstract, however they do not define constructionism. Rather, the most important concept within constructionism is that learners construct their knowledge as opposed to teachers instructing knowledge, to which Papert states, "the revolution I envision is of ideas, not of technology."[53]

Banking and Broadcasting (*A* Students)

The traditional system of education in the United States is focused on developing what Resnick refers to as *A* Students.[54] Resnick stated that "Most schools in most countries place a higher priority on teaching students to follow instructions and rules (becoming *A* students) than on helping students develop their own ideas, goals, and strategies (becoming *X* students)."[55] The pedagogies for developing *A* students have been characterized in a number of ways. Freire described this type of pedagogy as the banking concept, meant to oppress

---

53. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 2813.

54. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 89.

55. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 104.

learners.[56] Freire called for change, stating "They must abandon the educational goal of deposit-making and replace it with the posing of the problems of human beings in their relations with the world."[57] Resnick called it "a broadcast approach to education; that is, the teacher stood in front of the classroom and broadcast information."[58] In this analogy, learners are just receivers. Learners have an insignificant role in this type of education. They do not create or think creatively. Resnick stated, "Too often, schools focus on delivering instruction and information rather than supporting students in the creative learning process."[59] *A* students are constrained to classrooms in which a teacher, typically from the front of the room, broadcasts instructions in a procedural fashion ("do this first," "next do this," etc.) that are further constrained by a set of rules that students must operate within. Such an approach contrasts heavily with what Resnick described in *Lifelong Kindergarten*, "According to Piaget's constructivist theory of learning, children are active builders of knowledge, not passive recipients.

56. Paulo Freire, *Pedagogy of the Oppressed* (New York, NY: Bloomsbury, 2000), 60.

57. Freire, *Pedagogy of the Oppressed,* 1111.

58. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 158.

59. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 232.

Children don't get ideas, they make ideas."[60] When learning concretely, "Knowledge isn't poured into children, like water into a vase. Instead, children are constantly creating, revising, and testing their own theories about the world as they play with their toys and friends."[61] The iterative process of learning that Resnick describes may be a significant model for Papert's revolution of ideas.

Embracing Papert's revolution of ideas will challenge classroom teachers to shift their epistemologies and embrace their role as facilitator so that learners have the opportunities and resources to construct their knowledge. Falbel stated, "Better learning will not come from finding better ways for the teacher to instruct, but from giving the learner better opportunities to construct."[62] Falbel further argued that instruction (i.e., the banking concept or the broadcast approach) is problematic and "is like a strong medicine. If it comes at the right time and at the right dosage, then it can indeed be helpful. But if it's administered at the wrong time (against the learner's will) or at the wrong dosage (too much or too little), then it can be a hindrance or even intellectually

---

60. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 549.

61. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 547.

62. Falbel, "Constructionism: Tools to Build (and Think) With," 3.

poisonous!"[63] School curricula that favor instruction do not work well to provide opportunities for learners to construct knowledge with concrete representations and activities. Resnick stated, "In recent years, a growing number of researchers have argued that people form their strongest relationships with knowledge through "concrete" representations and activities--very different from the formal, abstract representations and approaches favored in traditional school curricula."[64] Constructing new knowledge requires one to think creatively with iterative processes for improvement. Students who learn this way are *X* students.

Creative Thinking (*X* Students)

While creative thinking may not frame traditional school curricula, Resnick reminds us that creative thinkers have emerged throughout history. He stated, "Risktakers. Doers. Makers of things. These are the *X* students, the creative thinkers. They've been the driving force for economic, technological, political, and cultural change throughout history."[65] It is important that educators facilitate learning opportunities for *X* students, as they will continue to

---

63. Falbel, "Constructionism: Tools to Build (and Think) With," 4.

64. Resnick, "Technologies for Lifelong Kindergarten," 46.

65. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 488.

drive economic, technological, political, and cultural change. Resnick stated, "In a society characterized by uncertainty and rapid change, the ability to think creatively is becoming the key to success and satisfaction, both professionally and personally."[66] He continued, "For today's children, nothing is more important than learning to think creatively – learning to come up with innovative solutions to the unexpected situations that will continually arise in their lives."[67] Growing creative thinkers means facilitating opportunities for learners to utilize materials and resources to solve relevant problems in the world. When these learners are thinking creatively and concretely using materials and resources to solve relevant problems, they construct new knowledge that will drive change.

The Creative Learning Spiral. The process of constructing knowledge by concrete learning through creative thinking is referred to by Resnick as the Creative Learning Spiral.[68] He stated, "The Creative Learning Spiral is the engine of creative thinking. As kindergarten children go through the spiral, they develop

---

66. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 1.

67. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 1.

68. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 227.

and refine their abilities as creative thinkers."[69] Resnick elaborated on the spiral, in which "children imagine what they want to do, create a project based on their ideas, play with their creations, share their ideas and creations with others, reflect on their experiences – all of which leads them to imagine new ideas and new projects."[70] He stated that "iteration is at the heart of the creative process" and that "The process of Imagine, Create, Play, Share, and Reflect inevitably leads to new ideas – leading back to Imagine and the beginning of a new cycle."[71] Without iterating this process, "they'll miss out on the most important part of the creative process," which are the opportunities to share and reflect.[72] Sharing and reflecting will often garner improvements or new ideas.

The design process. The Creative Learning Spiral is similar to the design process in that "Students rapidly build prototypes, play with them, share their prototypes with other students, and reflect on what they've learned. Then, it's

---

69. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 227.

70. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 2.

71. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 5.

72. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 5.

time to imagine the next version of the prototype, and they go through the spiral again — and again and again."[73] An idea for facilitating concrete learning for what are often considered abstract concepts is to engage learners in design activities. Children are active participants in design activities and have a greater sense of control over their learning, contrasting with traditional school activities in which new information is broadcasted to students.[74] In regard to content, "Design activities are often interdisciplinary, bringing together concepts from the arts, mathematics, and sciences."[75] Design processes often reveal multiple strategies to solve problems, or multiple solutions. Facilitating the design process in classrooms encourages the idea that multiple strategies and solutions are possible, as opposed to the right/wrong dichotomy widespread throughout most math and science classrooms in the United States.[76] Reflecting fuels thinking that leads to empathy. Resnick stated, "Design activities provide a context for reflection. A child's constructions serve as external shadows of the child's internal mental models--providing an opportunity for children to reflect on (and then

---

73. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 237.

74. Resnick, "Technologies for Lifelong Kindergarten," 44.

75. Resnick, "Technologies for Lifelong Kindergarten," 44.

76. Resnick, "Technologies for Lifelong Kindergarten," 44.

revise and extend) their internal models of the world."[77] He continued, "Design activities encourage children to put themselves in the minds of others, since they need to think through how other people will understand and use their constructions."[78] Design activities have the potential to concretize learning in multiple subject areas.

Fostering the growth of *X* students is rooted in creativity. Resnick stated,

> Creativity grows out of a certain type of hard work, combining curious exploration with playful experimentation and systematic investigation. New ideas and insights might seem like they come in a flash, but they usually happen after many cycles of imagining, creating, playing, sharing, and reflecting — that is, after many iterations through the Creative Learning Spiral.[79]

Electronic and digital materials, specifically computers, provide a broadly applicable medium for creativity. In constructionism, Papert, Resnick, and others view the computer, and other electronic and digital materials, as a means to concretize abstract ideas.

---

77. Resnick, "Technologies for Lifelong Kindergarten," 44.

78. Resnick, "Technologies for Lifelong Kindergarten," 44.

79. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 334.

Electronic & Digital Materials

Friedrich Froebel, the inventor of kindergarten, furnished his kindergarten classrooms with 20 gifts or manipulatives with which learners can create new things.[80] Froebel's gifts are the foundation for concrete learning in kindergarten classrooms. Resnick viewed Froebel's work as an early example of Papert's constructionist approach to education.[81] Resnick also viewed Froebel's gifts as something to expand upon. He stated, "With today's electronic and digital materials, we can create new types of construction kits, expanding Froebel's kindergarten approach to older students working on more advanced projects and learning more advanced ideas."[82] An example of this are the LEGO/Logo kits. Resnick stated, "Whereas traditional construction kits enable children to construct structures and mechanisms, LEGO/Logo goes further by enabling children to construct behaviors."[83] Children construct behaviors by programming them. Resnick continued, "LEGO/Logo users are not given ready-made

---

80. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 165.

81. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 3.

82. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 3.

83. Resnick, "Technologies for Lifelong Kindergarten," 46.

mechanical objects; they build their own machines before programming them. Elementary school students have used LEGO/Logo to build and program a wide assortment of creative machines."[84]

For Papert, the computer is an essential medium for more advanced projects and for learning more advanced ideas that are traditionally regarded as formal or abstract. Papert argued that "the computer can concretize (and personalize) the formal."[85] He continued, "Knowledge that was accessible only through formal processes can now be approached concretely."[86] In Piaget's view, formal/abstract processes were distinguished from concrete processes in that they lacked manipulatives to concretize learning. To Papert, the computer, in particular the ability to program a computer, is a manipulative to concretize formal/abstract processes. Papert stated, "If a person conceives of children's intellectual development (or, for that matter, moral or social development) as deriving chiefly from deliberate teaching, then such a person would be likely to underestimate the potential effect that a massive presence of computers and other

84. Resnick, "Technologies for Lifelong Kindergarten," 47.

85. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 461.

86. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 464.

interactive objects might have on children."[87] With 99% percent of schools having access to digital learning with broadband internet access, the computer has become an integral part of education culture.[88] However, in many cases computers are not used as Papert envisioned.

Computer-aided instruction. Computers are mostly used in classrooms to aid instruction, or as Papert saw it, the computer is used to "teach the child."[89] Resnick elaborated on this notion, and stated "Most applications of computers in education, for example, use computers in rather superficial ways. They take traditional classroom activities and simply reimplement them on the computer."[90] He stated later that "Too often, designers of educational materials and activities simply add a thin layer of technology and gaming over antiquated curriculum and pedagogy, somewhat like putting lipstick on a pig."[91] Papert summed up this

---

87. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 530.

88. Catherine Van Ness and Jake Varn, "Governors Leading On K-12 Digital Access State Successes In Bridging The Digital Divide For K-12 Students," National Governors Association, 2021, https://www.nga.org/news/commentary/governors-prioritize-expanding-internet-access-for-k-12-students/.

89. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 241.

90. Resnick, "Technologies for Lifelong Kindergarten," 45.

91. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 364.

idea and stated that "One might say the computer is being used to program the child. In [Papert's] vision, the child programs the computer and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building."[92] Papert's and Resnick's views situates learners as designers, in which the computer is expanded with new features, as opposed to operators, in which the computer remains limited with its existing capabilities.

Programming knowledge. The main issue with current applications of technology is constraint. Resnick stated that "Too often, educational technologies are overly constrained, such as tutoring software for teaching algebra, or simulation software for modeling planetary motion in the solar system. Our goal is to provide tools that can be used in multiple ways, leaving more room for children's imaginations."[93] By learning to program a computer, a learner can realize ideas in multiple ways. Resnick stated that "[Papert] viewed programming as a "universal language" for making new things with a computer and argued that

92. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 241.

93. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 2.

everyone should learn programming."[94] Resnick ties programming knowledge to fluency, which in turns fosters creative thinking in addition to one's voice and identity. He stated, "Most people won't grow up to become professional programmers or computer scientists, but learning to code fluently is valuable for everyone. Becoming fluent, whether with writing or coding, helps you to develop your thinking, develop your voice, and develop your identity."[95] The ability to program will bridge learners' creative thinking to relevant problems in meaningful ways. Knowledge of programming permits the learner to create more and removes constraints.

Meaningful Connections

Creative thinking is often facilitated by providing choice.[96] When learners choose what is important, their learning is more meaningful. Papert stated, "But if we can find an honest place for [creative] thinking in activities that the child feels are important and personal, we shall open the doors to a more coherent,

---

94. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 570.

95. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 708.

96. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 914.

syntonic pattern of learning."[97] Picard added to this, having stated, "In order for the learning to become truly rooted, a person has to have a deep emotional attachment to the subject area."[98] Resnick echoed this statement, "When learners have more choice and control, they can build on their interests and passions, and learning becomes more personal, more motivating, more meaningful."[99] Resnick later added the idea that meaningful connections do not just connect the learning to something relevant or important, but also previous ideas, experiences, and thinking. He later stated, "Learners also make deeper cognitive connections when they follow their interests. When activities involve objects and actions that are familiar and relevant, learners can leverage their previous knowledge, connecting new ideas to previously constructed mental models."[100] Falbel added that "elements of personal connection and care will serve to make the learning experience deep, meaningful, and long lasting."[101] Learners will dive deeper into

97. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 1381.

98. Rosalind W. Picard et al., "Affective Learning - A Manifesto," *BT Technology Journal* 22, no. 4 (2004): 264, https://doi.org/10.1023/B:BTTJ.0000047603.37042.33.

99. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 1096.

100. Resnick, "Technologies for Lifelong Kindergarten," 46.

101. Falbel, "Constructionism: Tools to Build (and Think) With," 6.

content because they care about what they are learning.

Resnick points out that intrinsic value motivates students. He stated, "When people work on projects they care about, they're willing to work longer and harder."[102] This can negate the need for extrinsic rewards. Resnick adds, "Rewards can deliver a short-term boost — just as a jolt of caffeine can keep you cranking for a few more hours. But the effect wears off — and, worse, can reduce a person's longer-term motivation to continue the project."[103] Resnick summarizes the value of intrinsic motivation, having stated, "But if your goal is to help people develop as creative thinkers and lifelong learners, then different strategies are needed. Rather than offering extrinsic rewards, it's better to draw upon people's intrinsic motivation — that is, their desire to work on problems and projects that they find interesting and satisfying."[104] Providing choices to learners to make meaningful connections facilitates multiple pathways to construct new knowledge.

---

102. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 280.

103. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 1036.

104. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 1054.

Wide Walls

Creative thinking is best facilitated by providing accessible entries to learning and is most sustainable when the learner is not hurdled by constraints or boundaries to their learning. Resnick recalled that Papert often emphasized the importance of "low floors" and "high ceilings."[105] Papert stated that for a technology to be effective, it should provide easy access for novices to start (low floors) but also allow ways for them to work on increasingly sophisticated projects over time (high ceilings).[106] Falbel points out that "one person cannot dictate what is to be personally meaningful for another person. This is where choice enters the picture."[107] Resnick refers to providing choices as "wide walls." He stated, "we try to design technologies that support and suggest a wide range of different types of projects."[108] He continued, "It's not enough to provide a single path from a low floor to a high ceiling; it's important to provide multiple

---

105. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 909.

106. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 908–909.

107. Falbel, "Constructionism: Tools to Build (and Think) With," 5.

108. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 913.

pathways."[109] Providing multiple pathways increases opportunities to make meaningful connections and for collaboration.

Collaboration

A learning environment with low floors, high ceilings, and wide walls may present opportunities for collaboration.[110] Falbel points out that a good learning environment will try to maximize choice, diversity, and congeniality.[111] Collaborative environments provide choices and are strengthened by diversity and congeniality. Falbel emphasized the value of diversity, having stated, "In a more diverse setting, people with less experience can learn much from freely associating with others who display a level of skill slightly above their own. People with more experience refine their skill and knowledge through helping and explaining things to others. And the diversity of artifacts created fuels everyone's creative imaginations. Ideas are borrowed and embellished in an exciting, vibrant cross-fertilization of knowledge."[112]

---

109. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 914.

110. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play,* 1283–1284.

111. Falbel, "Constructionism: Tools to Build (and Think) With," 5.

112. Falbel, "Constructionism: Tools to Build (and Think) With," 7.

Furthermore, in diverse environments, learners have the opportunity to establish meaningful relationships. Falbel stated that, "a good learning environment provides learners with time and space not only to do certain types of constructive work, but also to meet and form relationships with other people who are similarly interested in doing such work."[113] This is important because the building of relationships fosters empathy which enhances teamwork, to which Resnick stated, "teamwork is more important in today's workplace than ever before."[114]

Finally, collaboration and teamwork fuel creativity. Resnick stated, "children become more engaged in the construction process when they are able to share their constructions with others in a community, and children become more engaged with communities when they are able to share constructions (not just chat) with others within those communities."[115] Collaboration is enhanced by online communities where "people actively create and share ideas and media with one another on blogs and collaborative websites like Flickr (for photographs) and

113. Falbel, "Constructionism: Tools to Build (and Think) With," 8.

114. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 4.

115. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 5.

YouTube (for videos)."[116] Collaboration, including opportunities found in online communities such as those mentioned above fuel the Creative Learning Spiral where learners use creative thinking to solve relevant problems and construct new knowledge.

Critiques

There are few critiques of Papert's constructionism. The most prominent critic is Beynon and he primarily focuses on the role of computers and programming as a means to concretize learning. Beynon stated that "[Papert] is posing a challenging fundamental question about the relationship between computer-based activities and learning."[117] This is absolutely true in that Papert proposed a paradigm shift in how computers should be used in education, since computers are mostly "used either as a versatile video game or as a "teaching machine" programmed to put children through their paces in arithmetic or spelling."[118] In Papert's view, the learner "teaches" the computer using

116. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," 4.

117. Meurig Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," in *Educational Robotics in the Makers Era*, vol. 560, 2017, 4, https://doi.org/10.1007/978-3-319-55553-9.

118. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 587.

programming to construct new knowledge.[119] The output of the program is then an artifact of learning.

Beynon's perspective of computers in education is presented in a way that stresses the limitation of computers. He stated that "[Papert] proposes an engagement between a person and a computer resembling that between a scientist and musician and their instrument in which behaviours are crafted responsively moment-by-moment."[120] Beynon views this perspective as problematic since it does not align with conventional views of computers and computer science. Beynon further stated, "The world-wide dissemination of the SCRATCH culture might be taken as vindication of the idea that learners can use the computer to share and express their ideas as envisaged in Mindstorms."[121] Scratch does provide users the ability to share and express their ideas. More importantly, however, it is a means for learners to construct new things in the world, rather than just be passive recipients of information.[122]

---

119. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 936.

120. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 5.

121. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 7.

122. Resnick, *Lifelong Kindergarten: Cultivating Creativity through*

Beynon's perspective initially suggests that learning computer programming is skill oriented rather than a tool for creative thinking and problem solving. He stated, "Papert's lack of interest in starting from the kind of comprehensive understanding of the agency in a situation that is presumed when framing a computational process is what distinguishes his perspective from that of a professional programmer."[123] Papert does argue that everyone should learn computer programming.[124] Beynon later acknowledged that "it is helpful to bear in mind that Papert was primarily interested in how the computer could transform the experience of the learner, and was not narrowly committed to any particular technical approach to computing per se."[125] Although he stated later, "Programming is typically promoted as a discipline that teaches children the need for absolute precision in thought and expression, since there can be no

---

Projects, Passions, Peers, and Play, 566.

123. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 9.

124. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 903; Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 570.

125. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 10.

negotiation of meaning with a computer."[126] The "absolute precision in thought and expression" is the means by which a user *teaches* the computer. This is akin to what Papert describes as "descriptive language."[127]

Beynon continued with discussing construals. He stated, "Informally, 'making a construal' means figuring out how you think something works. The activity plays a fundamental role in everyday life, especially when we encounter unfamiliar situations, as in trying to make sense of a new place, culture or language."[128] Beynon used this to illustrate Papert's view of Piaget's constructivism, having stated, "A construal of the most basic concepts of linear algebra serves as a simple illustration of Papert's constructivist vision: it models the transition from concrete empirical observation of the canvas to the abstract formal structure of a 2-dimensional linear space."[129] Beynon recognized the value of construals, and their correlation to concretizing learning. He stated, "The most significant potential impact of introducing this new perspective is to give clear

---

126. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 11.

127. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 1404.

128. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 12.

129. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 13.

expression to the great varieties and subtleties of meaning that inhabit the space in which construction occurs."[130] This is an important aspect for educators to consider when they use formative assessments. Beynon stated, "We should not judge the quality of a construal by how easily it can be constructed by a learner, though we are concerned with how readily the learner can make insightful connections with other experience of the learning domain."[131] These "insightful connections" are ways in which educators can determine the meaningfulness of the learning occurring in their classrooms. Additionally, the insightful connections learners make provide educators a means in which to assess the quality of their instruction and facilitation.

Beynon uses music composition to evaluate the quality of construals, however, he omits a key participant in music composition — the composer. He stated, "the quality of a musical composition such as a concerto is evaluated in terms of the degree of satisfaction it gives to all who participate in its rehearsal and — the audience members, the soloist, the instrumentalists, the conductor,

---

130. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 16.

131. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 17.

the critics."[132] He continued, "A successful musical composition is appreciated by all participants in different ways and degrees according to their level of familiarity and expertise."[133] This perception perpetuates the idea of music as a performance discipline, negating the creative thinking processes of composers, producers, and songwriters. A composition or song may be the direct result of a construal by its maker. Evaluation of a construal may come from an external source; however, it must first come from its originator to determine whether the construal accurately reflects their understanding.

Algorithmic Composition

Algorithmic composition refers to the guidance of rules to compose a piece of music. The term can broadly be applied to many genres of music; however, it is most commonly used in reference to electronic music, often through programmability. Wang stated, "As electronic music evolved, analog synthesizers gained popularity (around the 1960s). They supported interconnecting and interchangeable sound processing modules. There is a level of programmability involved, and this block-based paradigm influenced later design of digital

132. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 17.

133. Beynon, "Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy," 17.

synthesis systems."[134] The block-based paradigm that Wang referred to resembles the block diagrams used to model signal flow in an electronic circuit. For this review, I will address literature that utilizes programmability in what Puckette generalizes as *computer music*.[135] It is important to make this distinction, because, as Ariza pointed out, "Labels such as algorithmic composition, automatic (or automated) composition, composition pre-processing, computer-aided (or -assisted) composition (CAC), computer composing, computer music, procedural composition, programmed music, and score synthesis have all been used to describe overlapping, or sometimes identical, projects in this field."[136] In regards to the specific use of "algorithmic composition," Ariza also stated, "The label "algorithmic composition" is likewise too broad, particularly in that it does not specify the use of a computer."[137]

Algorithmic composition, including computer music, has caused

---

134. Wang, "The ChucK Audio Programming Language. 'A Strongly-Timed and On-the-Fly Environ/Mentality'," 9.

135. Miller Puckette, The Theory and Technique of Electronic Music (Hackensack, NJ: World Scientific Publishing Co., 2007), xi.

136. Christopher Ariza, "An Open Design for Computer-Aided Algorithmic Music Composition: AthenaCL," PhD diss. (ProQuest Dissertations Publishing, 2005), 4.

137. Ariza, "An Open Design for Computer-Aided Algorithmic Music Composition: AthenaCL," 5.

controversy by raising questions about how music is composed. In response to Subotnick's *Wild Bull*, Dockstader asked, "how much is the Machine, how much is the Man?"[138] He continued and asserted, "This is a case of the machine influencing the music -- a continuing problem in electronic music, which depends so heavily on machinery for its composition and performance."[139] The instrument one composes for has always influenced composition and performance. For example, the modern piano is capable of greater amplitude, dynamic contrast, and articulations than its predecessor, the piano forte of the early nineteenth century. Compositions and performance spaces reflect these differences. Dockstader pointed out that spaces have significant influence on performance, and that, "In orchestral records, this is incidental to the work; in electronic music, it is a central part: the composer has to create his own space as well as his own sounds."[140] Computer music is dramatically influenced by space, but also by advances in technology.

138. Tod Dockstader, "Morton Subotnick: The Wild Bull a Composition for Electronic-Music Synthesizer," *The Musical Quarterly* 55, no. 1 (1969): 136.

139. Dockstader, "Morton Subotnick: The Wild Bull a Composition for Electronic-Music Synthesizer," 136.

140. Dockstader, "Morton Subotnick: The Wild Bull a Composition for Electronic-Music Synthesizer," 137.

Computer Music

Ada Lovelace is often credited for her foresight in computer music. Wang stated, "Ada Lovelace, while working with Charles Babbage, wrote about the applications of the theoretical Analytical Engine, the successor to Babbage's famous Difference Engine."[141] The Analytical Engine never came to fruition, however the idea of such a machine led Ada Lovelace to speculate on the capabilities of a computer, particularly in regard to composing music. When these capabilities were finally in their infancy, Hiller and Isaacson explored the capabilities of music composition with an early digital computer. They stated, "Computers are ideal instruments for purely abstract and unbiased study of musical concepts since the control over the musical output is limited solely by the input instructions."[142] Their efforts culminated into what they titled the *Illiac Suite*, published May 1957, by New Music Edition, New York.[143] They stated, "the musical content of each movement was obtained by means of unbiased

---

141. Wang, "The ChucK Audio Programming Language. 'A Strongly-Timed and On-the-Fly Environ/Mentality,'" 8.

142. L. A. Hiller and L. M. Isaacson, "Musical Composition with a High-Speed Digital Computer," *Journal of the Audio Engineering Society* 6, no. 3 (1958): 10.

143. Hiller and Isaacson, "Musical Composition with a High-Speed Digital Computer," 10.

sampling procedures so that the most representative sets of experimental results were included in the suite."[144] The unbiased sampling procedures were the algorithmic basis of the composition.

Computers, in general, were later recognized for other potential benefits. Puckette stated, "The computer, however, is relatively cheap and the results of using one are easy to document and re-create. In these respects at least, the computer makes the ideal electronic music instrument—it is hard to see what future technology could displace it."[145] Rambarran viewed the computer, specifically the laptop computer for its portability and unified construction as a "disruptive technology" because "the computer removes the need to use anything other than itself ."[146] In Rambarren's view, music made with laptop computers disrupts and challenges the classical tradition of Western music. Computers are prominent in the production of most modern music. Lyon stated, "The prevalence of the use of computers in today's music demands another distinction; at its

144. Hiller and Isaacson, "Musical Composition with a High-Speed Digital Computer," 10.

145. Puckette, *The Theory and Technique of Electronic Music*, xi.

146. Shara Rambarran, "'DJ Hit That Button' Amateur Laptop Musicians in Contemporary Music and Society," in *The Oxford Handbook of Music Making and Leisure*, ed. Roger Mantie and Gareth Dylan Smith (Oxford University Press, Incorporated, 2017), 591, http://ebookcentral.proquest.com/lib/bu/detail.action?docID=4745366.

outset computer music meant experimental music, carried out in laboratories and universities. This experimental work continues here and at many other institutions, but most of today's computer music is created in the field of entertainment, whether film music or the various technology-drenched genres of rap, techno, rock, and pop."[147] Much of early computer music, however, was composed for the video game industry. The computer as a tool for music production, as mentioned above, took place after the music industry's adoption of the Digital Audio Workstation (DAW).

Programmable Sound Generators

The early days of the video game industry utilized 8-bit programmable sound generators (PSGs), including: Atari VCS, Nintendo Entertainment System (NES), Commodore 64, and coin-operated arcade machines.[148] PSGs are synthesizer voice integrated circuits that generate waveforms with functional limitations.[149] Commodore 64's Sound Interface Device (SID) is a famous

---

147. Eric Lyon, "Dartmouth Symposium on the Future of Computer Music Software: A Panel Discussion," *Computer Music Journal* 26, no. 4 (2002): 13, https://doi.org/10.1162/014892602320991347.

148. Karen Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," *Twentieth-Century Music* 4, no. 2 (2007): 210, https://doi.org/10.1017/S1478572208000510.

149. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game

example, consisting of a "three-plus-one generator chip" developed by Bob Yannes.[150] Collins stated, "Each tone on the chip could be selected from a range of waveforms – sawtooth, triangle, variable pulse (square wave), and noise. An independent ADSR envelope generator for each channel enabled the SID to imitate traditional instruments more accurately than did previous chips."[151] Programmers of these chips needed functional knowledge of sound synthesis to achieve their desired sound.

The first composers for PSGs were programmers. Collins stated, "Early sound programmers and musicians needed to understand assembly language to engage the chips, which meant that most of the early composers for games were, in fact, programmers working on other aspects of a game."[152] Programmers were often uncomfortable composing new music for the games they created.[153] Rather,

---

Audio," 213.

150. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," 214–15.

151. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," 215.

152. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," 215.

153. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," 216.

they created more sophisticated sound effects and borrowed melodies from classical and popular music (e.g., rock and roll).[154] The NES system became more recognizable than the C64 for its game music, due to two major factors as Collins stated, "the music on the Commodore was coded in assembly language, which was harder to program than NES's BASIC-based language, and the availability of memory."[155] Collins noted however, that despite the challenges composing in assembly language, "Some Commodore 64 composers were also adventurous with their coding, including the use of random number generators into the code to select from a group of loop options."[156]

Languages for Sound & Music

Perhaps the greatest challenge composing with code is that most programming languages are not intended for musical composition, including languages intended for sound. An audience member at the Dartmouth Symposium stated,

---

154. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," 216.

155. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," 217.

156. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," 219.

> Fundamentally, they are not composition languages: they are instrument design languages. They're languages that give you a way to design different kinds of instruments and play them in various different ways. They do not have anything to do with music at the fundamental level, just in the same way that putting together some strings and bits of wood and various things has nothing intrinsically to do with music. If you are making a violin, that is not the music, the music is what you do with it afterwards.[157]

The need for a computer music medium, a composition language, fueled the development of several new languages.

The development of music composition languages has been challenging. Furthermore, the development of a music composition language that is user friendly has been even more challenging. This sentiment was reiterated by Puckette at the Dartmouth Symposium when he stated, "How on earth are we going to make it so that people can just download the program that really does just give you a blank sheet of paper to work on and that allows you to put your own will on it?"[158] Even with the development of language, learning the language is still a challenge for users. Mathews stated at the Dartmouth Symposium, "I think the major cost of a lot of software is not the dollars that you have to spend for it, but the number of hours or weeks that you have to spend learning how to

---

157. Lyon, "Dartmouth Symposium on the Future of Computer Music Software: A Panel Discussion," 20.

158. Lyon, "Dartmouth Symposium on the Future of Computer Music Software: A Panel Discussion," 24–25.

use it."[159] Historically, this has been the case for several music programming environments. Even with such difficulty present, learning a music programming language is a prerequisite for code-based electronic music. In *Theory and Technique of Electronic Music*, Puckette illustrated examples in Pd, but encouraged readers to use the same techniques in other environments such as Csound or Max/MSP.[160] For Puckette, developing an understanding of computer music as you learn the language is a beneficial approach. Aaron embraced this mindset in the development of Sonic Pi, adding the benefit of meaningful connections. He stated, "Sonic Pi takes a different approach. It turns code into a powerful new kind of musical instrument with a focus on fast feedback and iterative learning. It enables students to code the kinds of music they're typically used to listening to."[161] Modern music programming environments, especially Sonic Pi, have made learning to code music more user friendly, offering a low floor to users. As Puckette stated, "You don't need much background in music as it is taught in the West; in particular, Western written music notation is not

---

159. Lyon, "Dartmouth Symposium on the Future of Computer Music Software: A Panel Discussion," 25.

160. Puckette, *The Theory and Technique of Electronic Music*, xi.

161. Sam Aaron, "Live Coding Education," *The MagPi Educator's Edition*, Education Special Issue 2, n.d., 44.

needed."[162] Computer music removes learning an acoustic instrument and reading Western staff notation as a barrier.

Live Coding

Live coding has emerged as a popular medium for computer music. Aaron et al. stated, "Live coding is a growing international phenomenon that brings together the creative skills of music making and computer programming; it is a mode of digital creativity that considers coding as a performance. Live coding extends the exciting notion of 'liveness' (Tanimoto 1990; Collins et al. 2003; Church 75 76 et al. 2010) with composition happening in the immediacy of improvised performance."[163] Goldman added, "live coding can generally be understood as a practice in which performers create music by writing algorithmic instructions for computers in real-time performance."[164] Often, the live coder's screen is a feature of the performance. Bovermann and Griffiths stated, "When live coding performers take to the stage and project their screens, they invite us

---

162. Puckette, *The Theory and Technique of Electronic Music*, xii.

163. Aaron et. al., "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 75–76.

164. Andrew Goldman, "Live Coding Helps to Distinguish between Embodied and Propositional Improvisation," *Journal of New Music Research* 48, no. 3 (2019): 281, https://doi.org/10.1080/09298215.2019.1604762.

to join them in attempting to understand this intricate dance between human and machine."[165] Most importantly though, is that live coding has developed into a music composition and performance medium for computer music. Cardenas stated, "Advances in computation mean advances in the capacity of translation of my musical thought and its mutations."[166] She continued, "As composer and performer, the flexibility of computer-based music allows me to do both at the same time."[167] Cardenas views advances in computation as means to realize composition, including changes to the composition, and performance at the same time. This advancement in technology is unaligned with Western traditions in which the composer, performer, and listener are different roles.

Disruptive Technology

Live coding is disrupting traditional Western music.[168] It blurs the lines between composer, performer, and listener, and the people who are stereotyped in

---

165. Till Bovermann and Dave Griffiths, "Computation as Material in Live Coding," *Computer Music Journal* 38, no. 1 (2014): 40, https://doi.org/10.1162/COMJ.

166. Cardenas, "Live Coding: A New Approach to Musical Composition."

167. Cardenas, "Live Coding: A New Approach to Musical Composition."

168. Joanne Armitage and Helen Thornham, "Don't Touch My MIDI Cables: Gender, Technology and Sound in Live Coding," *Feminist Review* 127, no. 1 (2021): 91, https://doi.org/10.1177/0141778920973221; Rambarran, "'DJ

these roles. However, despite this disruption, these distinctions still exist. Armitage and Thornham stated, “live coding is understood on the one hand as the ‘offspring’ of formal music composition that has always been technologised (e.g., scores), and as the live ‘performance’ of those technologies (complete with capacities for disruption and nuance) on the other. This introduces notions of play, disruption and generative capacities through live ‘performance’, but such generative capacities are realised through an affective response to the ‘performance’, and the live coder remains, as Thor Magnusson argues, ‘primarily a composer’.”[169] While these distinctions remain, technology enables us to see these distinctions concurrently. Cremata stated, “Those 3 divisions, composing, performing and listening, still to this day there are composers who compose, performers who perform and the masses who listen. But this technology can potentially soften those boundaries to the point that a person can be all three at the same time through these technologies.”[170]

Hit That Button’ Amateur Laptop Musicians in Contemporary Music and Society,” 591.

169. Armitage and Thornham, “Don’t Touch My MIDI Cables: Gender, Technology and Sound in Live Coding,” 92.

170. Radio Cremata, “The Use of Music Technology Across the Curriculum in Music Education Settings: Case Studies of Two Universities” (2010), 209.

Disembodied Technique

Live coding concerts facilitate the coexistence of “3 divisions.” Goldman stated, “In live coding concerts, performers often set up computational processes that generate an ongoing stream of musical output allowing the performer to turn their attention to writing or modifying processes.”[171] What this means is that live coding differs from traditional instrumental performance in that temporal synchronization of physical movements and sounds are not necessary. Goldman stated, “Certainly note-for-note sensorimotor engagement is not a necessary feature for something to qualify as a musical performance (improvisatory or not); however, this lack of sensorimotor engagement is notably different from musical performance using acoustic and electro acoustic instruments, as those practices do have sensorimotor engagement.”[172] He continued, “it is the code that determines the characteristics of the sound, not the specific qualities of the human movements used to write the code. By contrast, this is not true for a pianist or a wind player whose motor executions (finger movements, breath) do change the

171. Andrew R. Brown and Andrew Sorensen, “Interacting with Generative Music through Live Coding,” *Contemporary Music Review* 28, no. 1 (2009): 20, https://doi.org/10.1080/07494460802663991.

172. Goldman, “Live Coding Helps to Distinguish between Embodied and Propositional Improvisation,” 282.

sound by virtue of an analogue interface between movement and sound."[173]

In addition to the performing aspect, the listening aspect of the live coder is also separate. Goldman called the live coder's technique "disembodied," and stated, "The nonlinear temporal aspect of live coding practice also divorces physical movement from sensory feedback, and furthers my characterisation of its technique as disembodied."[174] Goldman clarified this by having stated, "If the coder makes a syntax error or other mistake, they can correct it before they execute the code, employing a temporal resource that is unavailable to instrumentalists that use embodied technique where there is no erasing the effect of an unintended finger movement."[175] Goldman acknowledged that live coding can incorporate both embodied and disembodied elements.[176] Embodied elements likely include the act of composing as it occurs in synchronization with performing (writing code) and listening (responding).

---

173. Goldman, "Live Coding Helps to Distinguish between Embodied and Propositional Improvisation," 282.

174. Goldman, "Live Coding Helps to Distinguish between Embodied and Propositional Improvisation," 283.

175. Goldman, "Live Coding Helps to Distinguish between Embodied and Propositional Improvisation," 283.

176. Goldman, "Live Coding Helps to Distinguish between Embodied and Propositional Improvisation," 283.

Spaces & Aesthetics

Live coding concerts have paved the way for new spaces and new aesthetics. Cardenas stated that, "[the] Algorave (a portmanteau of Algorithmic Rave) is a scene that promotes the creation of (often live coded) music for the dance floor, putting together the apparent disparate disciplines of computer programming, dance music, live veejaying, and musical composition in real-time."[177] She continued, "One of the practices of the algorave is to project the screen of the performer, revealing the code that is producing the music. This breaks down the mystique of the artist, revealing the process of musical creation and the possibility of appropriation by all."[178] The projection of the live coder's screen adds visual aesthetics to the performance. This addition enhances the audience's appreciation of the live coder's on-the-fly composing and performing. Wang and Cook stated that, "The on-the-fly programming aesthetic can be seen as one approach to address [technical and aesthetic intentions that are often difficult to discern], for it provides a channel for the audience to see both the intention and the results. Additionally, the appreciation of this aesthetic approach does not necessarily hinge on the audience understanding the code

177. Cardenas, "Live Coding: A New Approach to Musical Composition."

178. Cardenas, "Live Coding: A New Approach to Musical Composition."

(though this can certainly open new dimensions); it is more fundamentally about the act (and art) of construction, via coding."[179] Wang and Cook continued, and stated that "On-the-fly programming supplies a platform where the performer is able to render various types of mastery and creativity that can be immediately appreciated, or at least perceived. While typing speed may not inspire, the general expressive power of programming languages opens unlimited possibilities for clever approaches and beautiful design."[180] Collins exclaimed that, "A new area of performance is being opened up by laptop musicians attempting to work with scripting languages in live concerts."[181] Collins echoed Cardenas' assertion that live coding is disruptive to traditional Western music. In addition to blurring what Cremata calls the "3 divisions," Collins added that a programming mindset is also present. He stated, "Yet we do not wish to be restricted by existing instrumental practice, but to make a true computer music that exalts the

---

179. Ge Wang and Perry R Cook, "2004: On-the-Fly Programming: Using Code as an Expressive Musical Instrument," in *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*, ed. Alexander Refsum Jensenius and Michael J Lyons (Cham: Springer International Publishing, 2017), 206, https://doi.org/10.1007/978-3-319-47214-0_13.

180. Wang and Cook, "2004: On-the-Fly Programming: Using Code as an Expressive Musical Instrument," 206.

181. Nick Collins et al., "Live Coding in Laptop Performance," *Organised Sound: An International Journal of Music Technology* 8, no. 3 (2003): 329.

position of the programming language, that exults in the act of programming as an expressive force for music closer to the potential of the machine – live coding experiments with written communication and the programming mindset to find new musical transformations in the sweep of code."[182] He elaborated on the programming mindset, and stated that "Behind the Slub interfaces lie the 'compositional' or 'musical' processes – many separate pieces of code written as explorations of musical ideas. Each piece of code describes an experiment in such areas as combinatorial mathematics, chordal progressions, sonified models of dancing people, morphing metres, algorithmic breakbeats, and so on. In any case, the code engine is creating patterns, melodies and stranger musical components."[183] The live coder is doing more than just composing, performing, and listening. They are writing software.

Notation Versus Score

Live coding is inherently interdisciplinary, combining music composition, performance, and listening with computer science. Brown and Sorensen stated that, "writing software becomes part of the performance. Code becomes the musical score; a score that is written, modified and executed as part of the

182. Collins et al., "Live Coding in Laptop Performance," 322.

183. Collins et al., "Live Coding in Laptop Performance," 323.

performance. In this way live coding practices combine aspects of composition, arranging, improvisation and performance."[184] Magnusson challenged that code is the musical score since modifications are represented in the log as opposed to the code.[185] Brown and Sorensen continued, "The code acts as an intermediary between imagination and sound and is elaborated and transformed in performance within a tight feedback loop."[186] The feedback loop necessitates a listener. The listener may be the coder responding to their own music; however, listeners often make up an audience to which the coder responds as they compose. Like code for other programming applications, Brown and Sorensen stated that, "Notated in code, the music is available for reflection, reuse and modification. It can be saved, replayed, shared with others or analysed."[187] Here they refer to code as notation rather than score. They continued and solidified that assertion that "Computer programming code is the music notation of live

---

184. Brown and Sorensen, "Interacting with Generative Music through Live Coding," 20–21.

185. Thor Magnusson, "Scoring with Code: Composing with Algorithmic Notation," *Organised Sound: An International Journal of Music Technology* 19, no. 3 (2014): 275, https://doi.org/10.1017/S1355771814000259.

186. Brown and Sorensen, "Interacting with Generative Music through Live Coding," 20–21.

187. Brown and Sorensen, "Interacting with Generative Music through Live Coding," 18.

coding performances."[188]

Code may be the notation of live coding; however, is code also the musical score of live coding? Magnusson stated that, "Musical notation has typically followed the tradition of defining events on a linear timeline. This representation of time on a traditional stave is not perfectly isomorphic, as bars with notes of long duration are typically shorter on the paper than bars with many short notes. Such compression and expansion of 'spacetime' shows that the language of the score is event-based as opposed to time-based."[189] Magnusson pointed out that Western staff notation is represented as a timeline; however, the flexibility of spacetime in Western staff notation suggests that a musical score is event-based, rather than time-based. He continued that, "[Traditional staff] notation is a systematic format of instructions for the performance of musical events."[190] In that regard, Western staff notation shares a commonality with code - both are event-based. If we view Western staff notation as code, we may notice, as Magnusson pointed out that, "As a coherent code, [Western staff] notation is a

188. Brown and Sorensen, "Interacting with Generative Music through Live Coding," 18.

189. Magnusson, "Scoring with Code: Composing with Algorithmic Notation," 268.

190. Magnusson, "Scoring with Code: Composing with Algorithmic Notation," 269.

system of abstractions, affording certain types of expression but excluding others. Any system of abstractions has involved a *design process* of deciding what to include and exclude, judged from a perspective of what is important to express"[191] While Western staff notation may be viewed as a score and as code, the same is not entirely true for code used in live coding.

Code is the live coder's musical notation; however, for a complete picture of musical events in a performance, since code is added and edited throughout performance, more is needed to constitute a music score or transcription. Brown and Sorensen stated that, "For a full transcription of a live coding performance all of the changes would need to be logged."[192] This is problematic for some platforms used in live coding. Magnusson pointed out that live coding is essentially an improvisational practice and that live coding systems generally do not implement scoring mechanisms besides the code itself.[193] Magnusson has attempted to remedy this problem by developing an interactive musical score system for code that also represents time-based events called the Threnoscope. He

---

191. Magnusson, "Scoring with Code: Composing with Algorithmic Notation," 269.

192. Brown and Sorensen, "Interacting with Generative Music through Live Coding," 18.

193. Magnusson, "Scoring with Code: Composing with Algorithmic Notation," 274.

stated, "The code score in the Threnoscope is proposed as a solution to the problem of how code can be presented on a timeline axis, given that code is itself its best representation."[194] At present, more systems used in live coding, like Sonic Pi and ChucK, include logs for tracking actions.

Live coding has the potential to transform both music and computer science education. However, Armitage and Thornham asserted that its potential might be limiting and stated that "Hacking code and software means that live coding has the potential to be more esoteric and collaborative, offering specific avenues of intense subcultural deviation. It also means that we need to think of code, software and live coding as always in process, always being generated and produced, never fixed."[195] They also emphasized that live coding includes a process, which is also iterative, aligning well with a constructionist mindset, suggesting a place for live coding music and computer science education.

Music and Computer Science Education

Both music and computer science educators are advocating for new learners. Music education programs are shrinking and are often viewed as

194. Magnusson, "Scoring with Code: Composing with Algorithmic Notation," 275.

195. Armitage and Thornham, "Don't Touch My MIDI Cables: Gender, Technology and Sound in Live Coding," 98.

irrelevant to learners.[196] Many music educators grasp onto performance-based programs, rather than embracing emerging technologies that facilitate music composition and creativity.[197] Computer science programs perpetuate the rhetoric that we are falling short preparing our youth for future jobs and the economy.[198] In response to this rhetoric, many computer science programs integrate music without consideration to its impact on music education. There are examples, however, of interdisciplinary collaborations of music and computer science.

---

196. Nielsen, "Teen Playlist: Music Discovery, Production, and Sharing Among a Group of High School Students" 3; Adam J. Kruse, "'They Wasn't Makin' My Kinda Music': A Hip-Hop Musician's Perspective on School, Schooling, and School Music," *Music Education Research* 18, no. 3 (2016): 250, https://doi.org/10.1080/14613808.2015.1060954.

197. Beckstead, "Will Technology Transform Music Education?," 45; Bula, "Technology-Based Music Courses and Non-Traditional Music Students in Secondary Schools" 58; Aaron et. al., "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 89; Cathy Benedict and Jared O'Leary, "Reconceptualizing 'Music Making:' Music Technology and Freedom in the Age of Neoliberalism," *Action, Criticism, and Theory for Music Education* 18, no. 1 (2019): 28, https://doi.org/10.22176/act18.1.26.

198. Brooks and Lindgren, "Responding to the Coding Crisis: From Code Year to Computational Literacy," 1.

The Music for Computer Science Transaction

Code is often identified as the new requisite skill for future jobs.[199] To engage learners in code, educational programs leverage the holding power of music to hook learners. This transaction often benefits computer science education, and coding in particular, at the cost of music.

The Demand for Computer Skills

Coding knowledge has become a priority in schools, and with it, capitalism of computational literacy. Brooks and Lindgren are concerned that "the call to increase literacy standards is primarily about increasing individuals' credentials and private interests rather than the public good, and that individuals who do not attain these higher credentials fail because of personal shortcomings."[200] The rationale for learning to code is an exaggerated claim about America's economic deficits.[201] Code is positioned as a functional literacy necessary for meeting

---

199. Brooks and Lindgren, "Responding to the Coding Crisis: From Code Year to Computational Literacy," 2; Benedict and O'Leary, "Reconceptualizing 'Music Making:' Music Technology and Freedom in the Age of Neoliberalism," 27.

200. Brooks and Lindgren, "Responding to the Coding Crisis: From Code Year to Computational Literacy," 3.

201. Brooks and Lindgren, "Responding to the Coding Crisis: From Code Year to Computational Literacy," 4.

societal demands.[202] To recruit learners, individual credentialing is promoted to sell skills.[203] Coding is being marketed as a skill that can combat economic decline within the United States.

Numerous companies and public figures have expressed their support for public coding education campaigns. Vee noted that these companies and public figures are the primary advocates for the "code for everyone" agenda.[204] The "code for everyone" agenda is a campaign to bring computer science education to all schools in the United States. Although American capitalism was the motivation for the "code for everyone" agenda, Vee stated that improved access to technology and learning resources for coding have made learning coding easier.[205]Additionally, Vee pointed out that coding is applicable to a broader range of ideas, and stated, "the ability to read, modify, and write code can be useful not just for high-profile creative or business applications but also for

---

202. Benedict and O'Leary, "Reconceptualizing 'Music Making:' Music Technology and Freedom in the Age of Neoliberalism," 27.

203. Brooks and Lindgren, "Responding to the Coding Crisis: From Code Year to Computational Literacy," 5.

204. Vee, *Coding Literacy: How Computer Programming Is Changing Writing*, 1.

205. Vee, *Coding Literacy: How Computer Programming Is Changing Writing*, 5.

organizing personal information, analyzing literature, publishing creative projects, interfacing with government data, and even simply participating in society."[206] The "code for everyone" agenda has brought awareness to the virtues of coding as a skill. However, coding is more than a skill. It is a means to realize ideas, whether they are musical ideas or ideas that would have otherwise remained abstract. Papert's belief that everyone should learn coding was not for an in-demand skill. Rather, it was so anyone could realize ideas in multiple ways.[207]

Computer Science Benefits from Music

Papert developed the Logo programming language so that young learners could realize their ideas. In Papert's view, Logo was a way to concretize the abstract.[208] Limited support for music was added to Logo for learners to realize musical ideas when Jeanne Bamberger collaborated with Seymour Papert at the MIT AI Lab to adapt the Turtle Graphics programming language Logo to support music.[209] Since the adaptation of music to Logo, several other

---

206. Vee, *Coding Literacy: How Computer Programming Is Changing Writing*, 5.

207. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 570.

208. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 461.

209. William Payne and S. Alex Ruthmann, "Music Making in Scratch:

programming languages have been updated to support music. Music is engaging for many young learners, and computer science curriculum developers quickly found music to be an effective hook to capture learners' interests in computer science.[210] Burg, Romney, and Schwartz stated, "Students' natural interest in sound and music gives us an entrée through which to introduce STEM concepts."[211] Using music to capture computer science learners's attention can be enhanced by engaging with students' cultures, demonstrated by Krug et al. and the Code Beats project. Code Beats is a virtual camp event in which middle school "youth of color" learn computer science concepts by coding hip hop music in Sonic Pi.[212] They stated, "One alternative way of attracting students is to

---

High Floors, Low Ceilings, and Narrow Walls?," *The Journal of Interactive Technology & Pedagogy*, no. 15 (May 16, 2019), https://jitp.commons.gc.cuny.edu/music-making-in-scratch-high-floors-low-ceilings-and-narrow-walls/; Frankie Schembri, "Ones and Zeroes, Notes and Tunes," MIT Technology Review, February 21, 2018, https://www.technologyreview.com/2018/02/21/145234/ones-and-zeroes-notes-and-tunes/.

210. McCoid et al., "EarSketch: An Integrated Approach to Teaching Introductory Computer Music," 147; Magerko et al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 2; Horn et. al., "Music and Coding as an Approach to a Broad-Based Computational Literacy," 84.

211. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 663.

212. Lusa Douglas Krug et al., "Code Beats: A Virtual Camp for Middle

create an approach that engages with their culture, which has been shown to be effective."[213] This inclusion of music is often perceived as an exciting way to engage learners in computer science; however, in most cases, it is only meant to benefit computer science education, not music education.

Several emerging programs intentionally align learning to code music with computer science curricula and standards. In regards to Sonic Pi, Aaron stated, "Considerable care and attention has been placed to ensure that Sonic Pi allows educators to deliver all of the core concepts in the UK's new computing curriculum."[214] He continued, "You can easily use Sonic Pi to teach basic computing concepts such as sequencing, iteration, selection, functions, data structures, and algorithms."[215] Burg, Romney, and Schwartz noted the connection between sound, music, and STEM disciplines, and stated, "Sound and music applications are excellent for introducing STEM subjects, given that they

---

Schoolers Coding Hip Hop," in *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Association for Computing Machinery, Inc, 2021), 397–403, https://doi.org/10.1145/3408877.3432424.

213. Krug et al, "Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop."

214. Aaron, "Live Coding Education," 45.

215. Aaron, "Live Coding Education," 45.

combine mathematics, physics, electronics/engineering, CS, and programming."[216]

One could also argue that coding sound and music are beneficial beyond a mere introduction. Burg, Romney, and Schwartz also call the use of sound and music with STEM as "interdisciplinary," and stated, "The interdisciplinary collaboration was intended to make learning more engaging while staying true to the fundamental purpose, which is to present CS and mathematical concepts and skills."[217] Their statement that the fundamental purpose is to "present CS and mathematical concepts and skills" does not represent "interdisciplinary collaboration." Rather, it is a transaction that benefits Computer Science "Big Ideas" and mathematics at the cost of sound and music. The aim for Burg, Romney, and Schwartz was to meet the Learning Objectives (LOs) of the Advanced Placement (AP) Computer Science Principles (CSP) course. They stated, "Creative sound processing environments range from high level applications like Audacity and Sonar to programming languages like Java, Processing C/C++, or MAX. These environments provide endless opportunities to explore the LOs under creativity (e.g., LO2: collaborate in the creation of

---

216. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 667.

217. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 663.

computational artifacts, and LO5: use programming as a creative tool)."[218] Similarly, Krug et al. shared a similar approach, and stated that, "Code Beats uses a musical approach to teaching CS concepts. Its lessons provide graded scaffolding that allows beginners to thrive, intermediate students to grow, and experts to go deep. It uses hip hop to increase student engagement and cultural relevance."[219] Krug et al. are also addressing LOs for the Advanced Placement Computer Science Principles course (AP CSP course). They stated that, "Code Beats enables teaching a variety of CS concepts, including: Sequencing, Parameters, Loops, Data Structures, Functions, Variables, and Conditionals, covering the concepts in Units 4, 5, and 7 in the code.org CS Principles Curriculum Guide."[220] Freeman and Magerko targeted similar objectives with EarSketch, a coding platform for music, and stated, "We designed EarSketch to increase and broaden participation in computer science at the high school level, addressing critical challenges to engaging underrepresented groups in

---

218. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 666.

219. Krug et al., "Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop."

220. Krug et al, "Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop."

computing."[221] Their results echoed results from the Code Beats study in which "Based on survey's results and beats submitted during Code Beats and for the contest we can say that the use of hip hop to engage students in computational thinking is promising."[222] Freeman and Magerko also noted demonstrative success from a 2013 pilot study with teenage students enrolled in an introductory computer science course improving student engagement and content knowledge in computing.[223] Additionally, they stated that, "EarSketch students have thus far shown highly significant increases in intent to persist and content knowledge in computing."[224] Their efforts with EarSketch have yielded surprising results regarding music and code, most notably that, "Although the initial design hypothesis of the work was that this work would first appeal in general to most students but more so toward ethnic minorities (who may be more likely to have unfavorable views toward computing as a field), female students have been the

---

221. Jason Freeman and Brian Magerko, "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch," *Journal of Music, Technology, and Education* 9, no. 1 (2016): 61.

222. Krug et al., "Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop."

223. Freeman and Magerko, "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch," 62.

224. Magerko et al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 20.

most successful population group in terms of changing attitudes toward computing."[225] Females are significantly underrepresented in STEM career fields.[226]

In a separate study of EarSketch, Magerko et al. stated that, "An interesting finding from the focus groups is that EarSketch feels more authentic to students from a musical perspective rather than a programming perspective. Students recognize that they are learning to program, but they tend to see the transfer of programming into the music industry more strongly than the transfer into computing fields."[227] Magerko's findings in this study are compelling for music education. This finding may suggest that interdisciplinary studies of music and computer science could be mutually beneficial for both disciplines.

---

225. Magerko et al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 20.

226. Anthony Martinez and Cheridan Christnact, "Women Are Nearly Half of U.S. Workforce but Only 27% of STEM Workers," Census.gov, accessed April 2, 2023, https://www.census.gov/library/stories/2021/01/women-making-gains-in-stem-occupations-but-still-underrepresented.html.

227. Magerko et al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 20.

Critiques

Most critiques of educational programs using sound and music to engage learners in computer science tend to focus on the shortcomings of the programming languages rather than the transactional nature of sound and music for computer science.[228] Scratch, the popular blocks-based programming language has been specifically targeted for its poor implementation of music. Payne and Ruthmann stated three fundamental problems with Scratch's implementation of music. They stated that, "First, the functionality of music blocks is immediately accessible, but designed to play sounds at the level of "musical smalls" (i.e., the "atoms" or "phonemes" of music, such as an individual note, pitch, or duration) vs. "musical simples" (i.e., the "molecules" or "morphemes" of music, such as motives and phrases)."[229] The presentation of "musical smalls" is problematic because musical elements are taken out of context. The learner must understand how these blocks will work with one another before being able to construct a sequence that reflects a musical idea. They continued that, "Second, arising from Scratch's bottom-up programming style (i.e., building music code block by block,

228. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"; Beckstead, "Will Technology Transform Music Education?," 48.

229. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"

and note by note from a blank screen), the act of realizing musical sequences is tedious requiring a deep mathematical understanding of music theory to (en)code music."[230] Scratch parameters for sound are numerical, so the learner must define numerical relations of pitch or other parameters in order to construct musical ideas. Lastly, Payne and Ruthmann stated that, "Third, and perhaps most challenging to the end user for music, is a timing mechanism designed to privilege animation frame rates over audio sample-level accuracy. As illustrated by numerous Scratch projects and our own task breakdown, a user must take extra, often unintuitive steps to achieve adequate musical timing to implement basic musical tasks such as drums grooving together in time, or melodies in synchronization with harmony."[231] What this means is that Scratch lacks the timing mechanisms necessary to realize musical ideas requiring synchronization.

Payne and Ruthmann noted several additional reasons why Scratch may not be an appropriate coding environment for music. They stated, "Unfortunately, the three types of audio playback in Scratch lack consistency."[232]

---

230. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"

231. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"

232. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"

Consistency is an important characteristic of any programming language. An example of where this is problematic in Scratch, as Payne and Ruthmann stated is, "Setting a tempo affects pitched instruments and drums, but not audio files (e.g., sounds)."[233] Payne and Ruthmann view music composition in Scratch as cumbersome in general, and stated that, "One might argue that hard coding all of the pitches, like one might in a DAW, defeats the purpose of using a programming environment to represent music."[234] They continued, "in Scratch, beginning music coders must slog through dragging blocks and blocks of code before realizing even a simple melody, let alone a multi-part composition."[235]

When learning to compose music in the traditional Western sense, a learner is often expected to demonstrate proficiency in a musical instrument and Western staff notation as a prerequisite.[236] Learning to compose music in Scratch is similar in the sense that the user not only needs to understand basic music theory, but Scratch language syntax and mathematics as well before being able to

---

233. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"

234. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"

235. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"

236. Beckstead, "Will Technology Transform Music Education?," 45.

compose music. Payne and Ruthmann stated that, “Conventional instruction asks us to begin our educational work with the smallest musical components, often taken out of context, rather than those that our prior experience and developed intuitions with music may have provided.”[237] One aim of the music coding environments mentioned above is a low floor, an entry into music composition in which the learner does not need prior musical experience. As long as the music coding environment has high ceilings and wide walls in addition to low floors, they can often be leveraged to mutually benefit the learning of both computer science and music.

Mutually Beneficial Music and Computer Science Learning

Music has often been used as a hook to engage students in computer science learning. However, interdisciplinary music and computer science concepts can be taught in a way that views computer science as a means to facilitate new music composition. Computer science concepts and code in particular have been presented as the medium by which novel music may be composed. These new interdisciplinary methods encourage learners to identify as musicians as well as coders and to create new musics for new spaces.

---

237. Payne and Ruthmann, “Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?”

The majority of music classes at the high-school level are aimed at training performers.[238] Performance oriented music programs may have impacted the meaning of the word "musician," perhaps now projecting the perception that performance is a strong aspect of being a musician. Rambarran stated that, "music making is widely pursued and enjoyed as a leisure activity, individually and socially, as indeed it has been for centuries. Due to the developments and applications of digital technology, one could point to a shifting understanding of the term 'musician,' perhaps alongside a related debate around what is considered to be 'music.'"[239] The term 'musician' may now have less association with leisure activities in music. It is important that music educators guide learners in their identity development as musicians, especially as they engage in composing and listening, in addition to performing.

All learners can compose music without prior knowledge of an instrument or Western staff notation with musical code. Freeman and Magerko stated that, "At this level, even students with no background in music composition, theory or performance can rapidly create music that they feel is personally expressive and

---

238. Williams, "Reaching the 'Other 80%:' Using Technology to Engage 'Non-Traditional Music Students' in Creative Activities"; Deutsch, "Where Was Technology and Music Education Twenty Years Ago?," 93.

239. Rambarran, "'DJ Hit That Button' Amateur Laptop Musicians in Contemporary Music and Society," 586.

that they wish to share with others."[240] This is especially possible with Sonic Pi, which was intentionally created to engage young learners in music creation. Aaron stated that, "One of the principle design decisions continually re-taken with the evolving architecture is to ensure that every feature is simple enough to teach to a 10 year old child. This limitation means Sonic Pi does not contain as many features as other live coding systems - however, the features it does contain are often easier to learn, understand and master."[241] Although Sonic Pi was funded by the Raspberry Pi Foundation to introduce learners to computer science, its developer, Sam Aaron has discussed its implications on music education.[242] He stated that, "The practice of live coding has the potential to transform music education by exploring the creative potential of responsive programming languages and environments to provide new pathways for young

---

240. Freeman and Magerko, "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch," 64.

241. Samuel Aaron, "Sonic Pi - Reliable Randomisation for Performances," *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC* 2016-Novem (2016): 242, https://doi.org/10.1109/VLHCC.2016.7739697.

242. Aaron, Blackwell, and Burnard, "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 79.

people into digital arts."[243] However, live coding and its attribution to performance may be a turn-off for some learners.

Making music with code is not limited to live coding. Rather, a non-performance approach (as opposed to performance attributed to live coding) has been suggested as an appropriate starting point for young learners to begin composing.[244] Petrie stated that, "However, negative reports from beginner programmers were voiced in the Burnard et al. (2016) study when live coding in front of peers, which have been reflected in EarSketch by Freeman and Magerko (2016). As a result, music composition — where students prepare code instead of performing it — is more appropriate for beginners at a school level and the aims of the present study."[245] Specifically, Freeman and Magerko stated that, "The students were universally focused on creating 'MP3s', not performances, for two reasons. First, they wanted something they could put onto their phone and share with their friends through social media. Second, they were more interested in

243. Aaron, Blackwell, and Burnard, "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 89.

244. Freeman and Magerko, "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch," 24.

245. Petrie, "Programming Music with Sonic Pi Promotes Positive Attitudes for Beginners," 3.

learning about DAW production, song structure and making music from loops than in learning about interactivity and performance."[246] Freeman and Magerko pointed out that sharing and social media were significant factors — learners want a community in which they can share their music. Also, since EarSketch utilizes digital audio workstation (DAW) elements in its graphical user interface (GUI), learners are making connections to DAW-based music production.

Most music production facilities rely on a DAW (or DAWs) as part of their workflow. This leads many learners interested in music production to explore a DAW at some time in their education. Hancock stated that, "Students on the Production pathway are comfortable working with technology, but their practice relies heavily on digital audio workstation (DAW) software. They have a firm grounding in synthesis and sampling but little or no experience of programming languages."[247] The reliance on DAW workflows have prevented learners from exploring the benefits of musical coding, which allows more granular control over sound and a window to view application development for music production. Hancock continued, and stated that, "Specific comments

246. Freeman and Magerko, "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch," 64.

247. Hancock, "Play-Based, Constructionist Learning of Pure Data: A Case Study," 97.

indicated that students found the programming environment difficult in comparison to the DAWs on which most of them rely. Some students recognized Pd as potentially useful, although many students questioned its relevance to their practice."[248] Learning a programming language often presents challenges to learners. The language and the pedagogical practice of the educator will influence the learner's ability to construct knowledge with code.

Collins argued that learning code is easier when learners can manipulate concrete examples. He stated that, "Programming languages are typically easier to work with for most learners if concrete examples are provided, with students constructing their own generalizations, rather than dry and abstract outlining of technical concepts."[249] Collins modeled ways to manipulate examples. He stated, "Teaching is led by examples, and students are encouraged to modify the examples to pursue musically interesting variations, obtaining feedback on the success of variation of the underlying programming code simultaneously."[250] He continued, "The only reasonable solution I have found is to use live coding as an instructor, to build the program in front of the students, explaining each step of

---

248. Hancock, "Play-Based, Constructionist Learning of Pure Data: A Case Study," 102.

249. Collins, "Live Coding and Teaching SuperCollider," 7.

250. Collins, "Live Coding and Teaching SuperCollider," 7.

the way and its musical justification."[251] With musical code, examples are more engaging when they produce a musical, or at least a sonic output. Collins stated that, "Audio-visual outputs tend to hold greater motivation than printing text or writing to databases."[252] Additionally, learners make meaningful connections to their learning when, as Aaron, Blackwell, and Burnard stated, "It was found that lessons where there was not a clear musical goal, but that were constructed to communicate a specific computer science concept, were far less effective."[253] Aaron, Blackwell, and Burnard's findings support the idea that music is engaging and meaningful to many learners. The means by which music coding is presented and facilitated may impact learners' engagement.

When learners are permitted to construct their own knowledge with the freedom of exploration, their experiences are often more positive and they become invested.[254] Hancock experienced this when learners were allowed to play with Pd. He noted that play-based aspects received nine positive comments, one of

---

251. Collins, "Live Coding and Teaching SuperCollider," 8–9.

252. Collins, "Live Coding and Teaching SuperCollider," 7.

253. Aaron, Blackwell, and Burnard, "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 86.

254. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 1772.

which was from a student who previously commented negatively on an encounter with Max/MSP.[255] Hancock stated that, "Students confidently explored [Pd] by customizing patches. One typed new values into the objects of the ambulance patch to change the character of the LFO-driven siren, epistemically seeking to differentiate their functions. One added multiple `osc~` objects to the basic test tone patch to form a chord, demonstrating integration. Another asked how he could modify the exterminate audio effect patch, possibly motivated by its strong affordance of ludic play."[256] Collins shared that learners will continue to construct knowledge in pursuit of a personal goal, and stated that, "Sometimes, students are sufficiently addicted by this point that they will overcome all manner of obstacles in the pursuit of their personalized drum machine project; for others, core principles have not adhered."[257] There could be several reasons why core principles were not constructed by learners. The floor for learning Pd is not as low as other music programming languages, for example Sonic Pi. The facilitator may also find that learners need additional support grasping specific concepts.

---

255. Hancock, "Play-Based, Constructionist Learning of Pure Data: A Case Study," 102.

256. Hancock, "Play-Based, Constructionist Learning of Pure Data: A Case Study," 101.

257. Collins, "Live Coding and Teaching SuperCollider," 7.

For learners, there can be obstacles when learning how to make music with code. Collins stated that, “There can be a tension where the difficulty of more advanced programming work intervenes before the real potential of computer programming in music.”[258] He also stated that, “While simple examples are accessible enough, larger projects and more involved programming structures and principles can lead to trouble.”[259] This is alleviated to some degree when learners have prior experiences with code. Petrie stated that, “prior programming experience positively affects attitudes more than those without.”[260] Code needs to be written in accordance with the syntax rules specific to that language. Collins stated that, “To complicate matters, many computer science concepts simply have a particular corresponding [SuperCollider] syntax, since [SuperCollider] is itself a full programming language, and certain students may fixate on the pure programming side above of musical examples.”[261] He continued, “Code a student has spent time typing can provide a sense of personal investment, though programming’s dire requirement for perfect syntax can still obstruct getting to

258. Collins, “Live Coding and Teaching SuperCollider,” 8.

259. Collins, “Live Coding and Teaching SuperCollider,” 7.

260. Petrie, “Programming Music with Sonic Pi Promotes Positive Attitudes for Beginners,” 7.

261. Collins, “Live Coding and Teaching SuperCollider,” 10.

the musical outcome and frustrate the student left behind."[262] Collins encouraged the students to make mistakes in an effort to demonstrate that the consequences of syntax errors are minor.[263] This mindset aligns with Papert's idea to question whether something is fixable rather than succumb to the dichotomy common in traditional classrooms of whether it is right or wrong.[264] Collins utilized the classroom community to assist with this, stating that, "I find myself eager for students to call out requests, and challenge the musical direction of the class; a huge amount can be learnt working together with students through musical problems, jointly considering possible SC solutions."[265] This is also a method to collect formative assessments of learners' knowledge.

Assessing learners' knowledge is necessary to develop meaningful opportunities that guide learning. Collins provided examples of different tasks he assessed, and stated that, "Assessment involved creative musical programming tasks, such as creating a sound synthesizer with user interface, or an algorithmic

262. Collins, "Live Coding and Teaching SuperCollider," 9.

263. Collins, "Live Coding and Teaching SuperCollider," 9.

264. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 495.

265. Collins, "Live Coding and Teaching SuperCollider," 9.

composition, contextualized with respect to computer music precedents."[266] For Collins, assessments included feedback from collaborations, facilitated by group work and pair programming. He stated that, "By actively involving students in group work, rather than the traditional model of lone programming, it provides a potential boosting mechanism to facilitate overall student learning."[267] He continued, and stated that, "The experience of pair programming was helpful to the students in general, and welcomed by them as facilitating learning, though there were a few individual concerns on the mismatch of ability level."[268] Ongoing assessments, by the educator and learners' peers, will facilitate construction of knowledge and reveal opportunities for growth.

A constructionist approach to learning music composition with code can give learners the opportunity to realize musical ideas in a low floor, high ceiling, and wide walls environment. Learners are encouraged to draw from their knowledge and experiences to create. Educators who embrace and nurture participation witness growth. Aaron, Blackwell, and Burnard stated that, "Nurture the emerging 'participatory culture' within each classroom by enabling

---

266. Collins, "Live Coding and Teaching SuperCollider," 11.

267. Collins, "Live Coding and Teaching SuperCollider," 11.

268. Collins, "Live Coding and Teaching SuperCollider," 12.

students to recognize their own musical versatility and creativities through play, experimentation, risk taking, and free interplay between old and new elements across diverse musical styles and genres."[269] Collins added that as growth occurs, new ideas emerge, and stated that, "The demands of novel computer music lead students into novel programming tasks."[270] Novel programming tasks will encourage music composition only possible with technology. Han stated, "In particular, some of the compositional processes were only possible with technology, which allowed composers to create unique sounds like those of musical instruments, to compose unorthodox melodies, and to instantaneously experiment various instrumentations and arranging. Thus, the researcher concluded that implementing music technology in the music creation process improved the results of composition in terms of efficiency and musicality."[271]

While some novel music compositions may only be realized with technology, Han found that composing music this way had a positive impact on

---

269. Aaron, Blackwell, and Burnard, "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 89.

270. Collins, "Live Coding and Teaching SuperCollider," 14.

271. Jinseung Han, "Digitally Processed Music Creation (DPMC): Music Composition Approach Utilizing Music Technology" (ProQuest Dissertations Publishing, 2011), 204.

musical knowledge. Han stated that, "A conclusion drawn from this finding is that employing music technology in music composition pedagogy can significantly improve the results, in particular, the effects shown in basic musicianship, music analysis and composition technique."[272] Music composition with code has the potential to impact music education in novel ways. Music composition with code may encourage music composition as a leisure activity, and perhaps lead more learners to identify as musicians. It may also facilitate more problem-solving mindsets, and movement away from the right/wrong dichotomy. Music composition with code may also lead to novel musics, composed by people who may have otherwise lost an opportunity to create music that was driven by a performance mindset.

Critiques

Approaching musical code with a mutually beneficial interdisciplinary mindset has drawn critique, most notably that inevitably interdisciplinary integration will favor a primary discipline. Burg, Romney, and Schwartz stated that, "Often an "interdisciplinary" program ends up being administered directed

272. Han, "Digitally Processed Music Creation (DPMC): Music Composition Approach Utilizing Music Technology," 205.

solely from one department, whether that be art, music, or computer science."[273] An additional factor is that learning objectives often favor one discipline over another. Burg, Romney, and Schwartz also stated that, "It's difficult for music, art, and computer science departments to agree on learning objectives, given that the disciplines are so different."[274] Burg, Romney, and Schwartz also pointed out that the educator's mindset may also present a barrier. They stated that, "In interdisciplinary courses, we can think of creative applications that require computers and programming for their solution, but we may not be self-aware enough in the problem-solving process."[275]

Music educators will drive the direction of music education. Likewise, computer science educators will drive the direction of computer science education. Interdisciplinary integration of music and computer science necessitates a mutual respect for both disciplines and their benefits to learners. Burg, Romney, and Schwartz gave this example, and stated that, "it is not enough [for learners] to have a foundation course in computer science at the introductory level and not

---

273. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 313.

274. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 313.

275. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 313.

follow up with more advanced courses that synthesize computer science and digital art in terms of a shared language and shared application."[276] Cremata added that learners' engagement with knowledge is also a factor, and stated that, "A potential exists to discover more about the phenomenon of street learned skills as compared with traditional classroom learning. Music technology is an excellent avenue to pursue this line of research, since it appears popular to learn about music technology through both approaches."[277] Educators and curriculum writers need to have an understanding of how learners engage with materials inside of the classroom and outside of the classroom.

Summary

In this review of literature, I explored constructionism as a framework for facilitating music creativity with a specific focus on traditional models (*A* student) versus a constructionist model (*X* student). I also explored algorithmic composition in the contexts of computer music, specifically for research and PSGs as a medium, and live coding as a convergence of performance, composition, and listening. Finally, I investigated music and computer science education through

---

276. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 313.

277. Cremata, "The Use of Music Technology Across the Curriculum in Music Education Settings: Case Studies of Two Universities," 226.

the lens of transactionary and interdisciplinary models. Computers used for research in music composition paved the way for educational models for musical coding. An early example of this was the addition of music capabilities to the Logo programming language.[278] Early programming of PSGs demonstrated the need for music composers to become proficient in code.[279] Live coding emerged illustrating the laptop computer as a "disruptive technology."[280] Live coding also brought into question the roles of performer, composer, and listener.[281] Computer science education, at the expense of music, contributed to my understanding of interdisciplinary collaborations of music and computer science and informed my use of the term *music for computer science transaction*. This led to greater considerations of how music and computer science can be mutually beneficial in an educational context. The literature in this review informed the questions and the direction of this research study. In chapters 5 and 6, I unpack the findings

---

278. Seymour Papert and Cynthia Solomon, "Twenty Things to Do with a Computer" (A.I. Laboratory, Massachusetts Institute of Technology, June 1971), 24.

279. Collins, "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio," 224.

280. Armitage and Thornham, "Don't Touch My MIDI Cables: Gender, Technology and Sound in Live Coding," 92.

281. Goldman, "Live Coding Helps to Distinguish between Embodied and Propositional Improvisation," 283.

from this study and discuss the implications of this study as it relates to music and computer science education.

CHAPTER 3: METHODOLOGY

The purpose of this study was to investigate perceptions of high school students who study music making with code in Sonic Pi. I am interested in knowing what musical ideas, if any, participants report learning through coding music with Sonic Pi and if coding music impacts participants' views of their ability to learn music. I am also interested in knowing if music making with code impacts students' interest in continued studies in music and/or computer science. To investigate these perceptions, I addressed this inquiry as a qualitative multiple case study.[282]

Qualitative Multiple Case Study

In a qualitative case study, the researcher chooses what (case) is to be studied.[283] The cases I chose to study are individuals in an extracurricular club at a public charter high school who have volunteered to participate on-site and remotely asynchronously via Canvas LMS. The research study was also made available to high school students nationwide to participate remotely and

---

282. John W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed-Methods Approaches,* (Thousand Oaks: SAGE Publications Inc., 2009), Kindle edition, 4022.

283. Robert E. Stake, "Qualitative Case Studies," in *The Sage Handbook of Qualitative Research*, edited by N. K. Denzin and Y. S. Lincoln (Sage Publications Ltd., 2007), 443.

asynchronously via Canvas LMS. I sought to understand the individual perceptions of each participant and how their perceptions orient with their making of music.[284] To do so, I concentrated on the experiential knowledge (i.e., making music with code) of each participant by paying meticulous attention to their making music.[285]

A qualitative multiple case study extends instrumental interest to multiple participants.[286] Instrumental interest is the focus on an item, for which decisions and choices are made by the participant that will support the researcher's inquiry.[287] My instrumental interest is the making of music and perceptions of making music by each participant in the study. By acquiring an understanding of code-based music making in each case, I may be able to better understand the implications of code-based music making for educators.

Qualitative Methods

As a case study researcher, I investigated what is common and particular about code-based music making for each participant.[288] During this investigation

284. Stake, "Qualitative Case Studies," 443.

285. Stake, "Qualitative Case Studies," 444.

286. Stake, "Qualitative Case Studies," 445–446.

287. Stake, "Qualitative Case Studies," 445.

288. Stake, "Qualitative Case Studies," 447.

I considered these items outlined by Stake: "the nature of the case, particularly its activity and functioning; its historical background; its physical setting; other contexts, such as economic, political, legal, and aesthetic; other cases through which this case is recognized; and those informants through whom the case can be known."[289] For the participants in this study, I focused on the following items: participants' process (activity) of making music and the resulting product (functioning) of their making music; the participant's prior knowledge and experiences (historical background); the impact of on-site learning versus remote, asynchronous learning (setting); and the aesthetics of each participant's making music.[290] After examining participants' data and considering the items mentioned above, I cross examined commonalities and particulars with the other participants in the study.[291]

Selecting Cases

For this study, I attempted to draw a purposeful sample of high school participants representing diverse demographics remotely from throughout the country.[292] To recruit remote participants, I distributed the research study

---

289. Stake, "Qualitative Case Studies," 447.

290. Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 461.

291. Stake, "Qualitative Case Studies," 450.

292. Stake, "Qualitative Case Studies," 451.

information to educators via discussion forums and social media groups identified as music education, music technology education, and computer science education forums or groups. I sent the research study information via email to educational administrators and requested they forward the study information to their educator networks. I asked educators to distribute research study information to high-school students, ages 13–18, and to encourage the students to participate. While a small number of remote individuals ($n = 20$) viewed the study, only a few ($n = 4$) engaged with the study and provided consent/assent, and none of these individuals completed the study.

In order to gather more participants, I chose to recruit high school students who were more accessible.[293] I distributed research study information on-site to students participating in an extracurricular computer club at a public charter high school. Interested participants self-enrolled in the study ($n = 10$), provided assent (if under the age of 18) and consent (if the participant is 18 or if a parent/guardian), and confirmed they met the inclusion criteria. Of the on-site participants ($n = 10$), less than a third ($n = 3$) completed the study. The data from participants completing the study materials informed this multiple case study.

293. Stake, “Qualitative Case Studies,” 451.

Delimitations

Participants in this study were delimited to include high school students, some of whom had studied music before and read Western staff notation in a formal environment. Others in the study had musical experiences but had limited or no studies of music in school. Participants also had a range of coding experience. Some participants had little prior experience writing code while others had experience in text-based, object-oriented coding environments. Any person not enrolled as a high-school student was excluded from this study. Participants ages 18 years or older provided consent to participate in the study. Participants under the age of 18 provided assent, with consent provided by their legal guardian to participate in the study.

Setting

Participants completed a series of Sonic Pi tutorials on-site and remotely asynchronously via a Canvas LMS course.[294] This approach permitted the recruiting and participation of cases from a body of interested students. Furthermore, this approach allowed participants to work at their own pace and during times that were convenient for them without the limitations of

---

294. Specific details on how and why the Sonic Pi curriculum was truncated can be found in CHAPTER 4: PROCEDURE OF THE STUDY.

synchronous sessions.[295] Outside of the on-site location, participants chose the best physical environment and the best computer environment to facilitate their participation. Participants had approximately three months to complete the research materials, however the duration of their active participation was less than five hours on average.

Data Collection

The study design consisted of four data collection methods frequently utilized in qualitative research: documents, multimedia artifacts, observations, and interviews.[296] Documents were in the form of questionnaires, surveys, notes from participants' comments, and analytics collected using Canvas LMS. Multimedia artifacts included the source code and exported audio of participants' music making. For observations, unstructured and semi-structured field notes were taken of participants' behaviors and activities at the on-site location.[297] Interviews were conducted virtually and recorded using Zoom, an online video conferencing application. Interviews lasted approximately 30 minutes, during which time the participants described details of their experiences during the case

---

295. Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 464.

296. Creswell, *Research Design*, 3946.

297. Creswell, *Research Design*, 3952.

study. Data collected during the active portion of the study was coded to identify emergent commonalities and particulars. The emerged commonalities and particulars informed low-level interview questions.[298] I functioned as the key instrument in developing the questionnaire, survey, and interview questions, available in Appendices 7, 8, and 9.[299]

Canvas LMS easily facilitated the distribution and collection of questionnaires, surveys, and lesson instructions, in addition to providing analytics displaying participants' views of pages and the elapsed time of their engagement. Questionnaires and surveys were each distributed as a "quiz," for which grading indicators were set to zero points. When completed and submitted, questionnaires and quizzes were simply marked "complete." Page views were a numerical tabulation of the different areas visited by participants within the study. Some participants explored several pages when initially enrolling, while others only engaged with the sequential lessons. In both cases, the page views counter incremented as the participants worked through the study materials. The study materials had completion rules in place that only permitted participants to work in sequential order. Elapsed time recorded participants' active time in study

---

298. Johnny Saldaña, *The Coding Manual for Qualitative Researchers*, 4th ed, (London, England: Sage Publications Ltd., 2021) 168.

299. Creswell, *Research Design*, 3854.

materials. Canvas LMS was new and unfamiliar to all participants; however, each participant was able to learn and engage with the interface quickly.

The lessons of the study were each distributed in Canvas LMS as "assignments," for which grading indicators were set to zero points. When completed and submitted, lessons were simply marked "complete." Upon completion of a lesson with Sonic Pi, participants uploaded their code, either as a text file or by copying and pasting their code into the Canvas editor as an artifact of their learning. Artifacts of participants' learning were collected and examined to reveal commonalities and particulars, with the first coding cycle focusing on individual works and the second coding cycle focusing on the whole portfolio, which is an approach informed by Cayari's dissertation.[300] The collection and analysis of artifacts was ongoing throughout the active period of the study. These commonalities and particulars helped inform questions for participants' exit interviews.

300. Christopher Cayari, "Virtual Vocal Ensembles and the Mediation of Performance on YouTube," PhD diss., (University of Illinois at Urbana-Champaign, 2016): 84; Saldaña, *The Coding Manual for Qualitative Researchers*, 333.

Observations of research study progressions and group interactions were ongoing during on-site meetings.[301] Field notes were recorded to capture the behaviors and activities of the participants.[302] An observational protocol was used to separate descriptive notes from reflective notes.[303] Descriptive notes primarily reconstructed dialogue between participants, accounts of specific occurrences, and timelines of the activities.[304] Reflective notes recorded my thoughts about my observations, including: my impressions, ideas, concerns, and questions.[305]

Exit interviews were conducted via the Zoom video conferencing application following the active period of the study to further explore participants' views of music making during the study. The purpose of the interviews was to collect insightful information about the participants' experiences making music with code. Interview questions inquired about notable

---

301. Ellen Perecman and Sara R. Curran, *A Handbook for Social Science Field Research Essays & Bibliographic Sources on Research Design and Methods,* (Thousand Oaks: Sage Publications, 2006) 108, Kindle.

302. Creswell, *Research Design*, 3952.

303. Creswell, *Research Design*, 3974.

304. Creswell, *Research Design*, 3974.

305. Creswell, *Research Design*, 3974.

particulars gathered from Canvas documents and participants' artifacts.[306] Interviews were recorded and automatically transcribed by Zoom.[307]

Data Analysis

To analyze the collected data, three analysis methods were used over the course of two analysis cycles. For the first cycle, a combination of descriptive coding and values coding were used. Descriptive coding is a broadly applicable method that is well-suited for coding documents, artifacts, and field notes during the active portion of the study; however, it is not the best coding method for case study interviews.[308] Since I was interested in the perceptions of participants about making music with code, values coding was selected to code interviews after the active portion of the study.[309] For the second cycle, I used longitudinal coding to identify changes that occurred in participants' thinking and perceptions, and how they constructed knowledge over the duration of the study starting from the entrance questionnaire through the interview.[310] This method is

---

306. Stake, "Qualitative Case Studies," 447.

307. High-level interview questions can be viewed in Appendix 9.

308. Saldaña, *The Coding Manual for Qualitative Researchers*, 134.

309. Saldaña, *The Coding Manual for Qualitative Researchers*, 167.

310. Saldaña, *The Coding Manual for Qualitative Researchers*, 321 and 331; Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 77.

highly appropriate for the constructionist framing of this study because participants' construction of knowledge by concretizing music took place over a span of time.[311]

First Cycle

To identify emergent commonalities and particulars in documents, multimedia artifacts, and field notes during the first cycle, descriptive coding was used. I coded any pattern occurring more than twice, similar to Williams.[312] Descriptive coding is also referred to as "topic coding," "topic tagging," or "index coding."[313] Descriptive coding is the assignment of basic labels to data (i.e. tagging) in order to inventory emergent topics.[314] Descriptive coding was chosen to codify documents, multimedia artifacts, and field notes because it may be applied to a breadth of data types.[315] Additionally, the primary goal of

---

311. Saldaña, *The Coding Manual for Qualitative Researchers*, 331; Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 77.

312. Roger Neil Williams, "Investigating Culturally Responsive Teaching in the Jamaican Secondary Music Classroom: A Multiple Case Study," PhD diss., (Boston University, 2022) 65.

313. Saldaña, *The Coding Manual for Qualitative Researchers*, 133.

314. Saldaña, *The Coding Manual for Qualitative Researchers*, 129.

315. Saldaña, *The Coding Manual for Qualitative Researchers*, 134.

descriptions is to assist readers to see the cases as I do.[316] Hashtags (#) were used to identify and link comparable topics.[317]

To collect participants' perceptions about their experiences making music with code during the first cycle, values coding was used. Values coding is the codification of qualitative data reflecting the values, attitudes, and beliefs of participants, representing their perspectives or worldview.[318] In this study, value represents the *importance* each participant attributed to making music with code.[319] Attitude represents the way each participant thinks and feels about making music with code.[320] Participants' beliefs embody their values and attitudes about making music with code, but are also informed by their knowledge, experience, perceptions, opinions, and biases.[321] Through the interviews, participants elaborated on their values, attitudes, and beliefs,

---

316. Saldaña, *The Coding Manual for Qualitative Researchers*, 134; Stake, "Qualitative Case Studies," 454.

317. Saldaña, *The Coding Manual for Qualitative Researchers*, 134.

318. Saldaña, *The Coding Manual for Qualitative Researchers*, 167.

319. Saldaña, *The Coding Manual for Qualitative Researchers*, 167; Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 1381.

320. Saldaña, *The Coding Manual for Qualitative Researchers*, 167; Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 233.

321. Saldaña, *The Coding Manual for Qualitative Researchers*, 168; Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 809.

contextualizing their perceptions about making music with code.

Second Cycle

The second coding cycle was aimed at identifying common and particular themes and categories as they emerged and progressed throughout the study.[322] Longitudinal coding was used to review data commonalities and particulars by category and theme comparatively over the duration of the entire study.[323] A longitudinal qualitative data summary matrix was utilized to categorically organize the data collection.[324] Saldaña's seven longitudinal coding categories were used, including: (1) increase and emerge, documenting change that takes place gradually; (2) cumulative, documenting the overall effect of several successive actions; (3) surges, epiphanies, and turning points, documenting a change of sufficient magnitude; (4) decrease and cease, documenting decreasing changes and cessation; (5) constant and consistent; documenting regularized activity over time; (6) idiosyncratic, documenting the unpredictable; and (7) missing, documenting the possibly and plausibly absent.[325] By utilizing longitudinal coding, I was able to identify fluid movements throughout the study.

---

322. Stake, "Qualitative Case Studies," 447.

323. Saldaña, *The Coding Manual for Qualitative Researchers*, 332.

324. Saldaña, *The Coding Manual for Qualitative Researchers*, 332.

325. Saldaña, *The Coding Manual for Qualitative Researchers*, 333.

Participants' Confidentiality and Privacy

Participants registered in Canvas LMS and were assigned a unique SIS (Student Information System) identification number following registration in the research study. Identifiable information was maintained in the "People" area of Canvas LMS. Canvas LMS allowed the course administrator to control access to virtual learning areas. Access to this information was restricted to the research team only to maintain confidentiality.

All participant data was protected by password and stored using the Canvas LMS cloud server. Backups of the cloud server were kept on a local storage device and secured under password protection. Participants did not have access to other participants' data because viewing restrictions for all assignment and assessment types in Canvas were limited to myself. No peer assignments were used in Canvas for this research study (for example: group projects or discussion boards). Participants did not have access to other participants' information within Canvas because the "People" tab was disabled. Participants had the opportunity to share their code with other participants and collaborate while meeting on-site; however, each participant submitted their code individually in Canvas. Identifiable information is not accessible to any persons outside the study.

Credibility & Triangulation

Throughout the study I utilized data collection and codification procedures that check findings for accuracy to safeguard the validity of the study.[326] By using Canvas LMS as the medium to distribute materials and collect data in a consistent manner, I was able to achieve data collection and procedural redundancy.[327] For observations, procedural redundancy was achieved by following an observation protocol.[328] For interviews, procedural redundancy was achieved by following an interview protocol.[329] Data collection varied, which was expected with qualitative observations and interviews, though the data enhanced the perspective of each participant.

In an effort to reduce opportunities for misinterpretation, I utilized multiple ways of collecting data, including: documents, artifacts, observations, and interviews, to reach triangulation.[330] To facilitate readers' interpretation of the data, and ensure validity of the study, I have described the design of the

---

326. Creswell, *Research Design,* 4173.

327. Stake, "Qualitative Case Studies," 454.

328. Creswell, *Research Design*, 3974.

329. Creswell, *Research Design,* 3984.

330. Stake, "Qualitative Case Studies," 454.

study separately from the interpretations and evaluations of the data collected.[331] Details of the study, including its design in chapters 3 and 4, and a thorough discussion of my interpretations and evaluations in chapter 6 are outlined for readers to determine its credibility.[332] The discussion in chapter 6 is intended to contribute to the study's external validity, so readers interested in transferability will have a framework for comparison.[333] I acknowledge that my interpretations and evaluations are not perfectly repeatable, so through triangulation, the reader can identify the ways in which I see each participant.[334]

---

331. Stake, "Qualitative Case Studies," 453.

332. Creswell, *Research Design,* 2218.

333. Creswell, *Research Design,* 4389.

334. Stake, "Qualitative Case Studies," 454.

CHAPTER 4: PROCEDURE OF THE STUDY

Participants created music using Sonic Pi, an application developed by Sam Aaron for coding music in the Ruby programming language.[335] Readers of this dissertation are encouraged to download and install Sonic Pi, available at https://sonic-pi.net, in order to experience the learning materials and participants' artifacts in a concrete way. Coding is a different way of creating music, just as Western staff notation and jazz improvisation are different from one another. Western staff notation is a written way of creating music. Jazz improvisation is a spontaneous way of creating music. Coding is a computational way of creating music.[336]

Participants completed a series of tutorials for the Sonic Pi application that I assembled during engagement on-site at extracurricular club meetings and remotely through asynchronous engagement in a Canvas LMS course. Tutorial materials were available to participants from the Sonic Pi website (https://sonic-pi.net/tutorial.html) or from within the Sonic Pi application (figure 1). In this chapter, I illustrate and discuss key examples from the tutorials to provide

335. Aaron et al., "The Development of Sonic Pi and Its Use in Educational Partnerships," 80.

336. This analogy is inspired by Papert's analogy of Turtle geometry to other styles of geometry. For more, see: Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 1022.

context for the data analysis from the study.

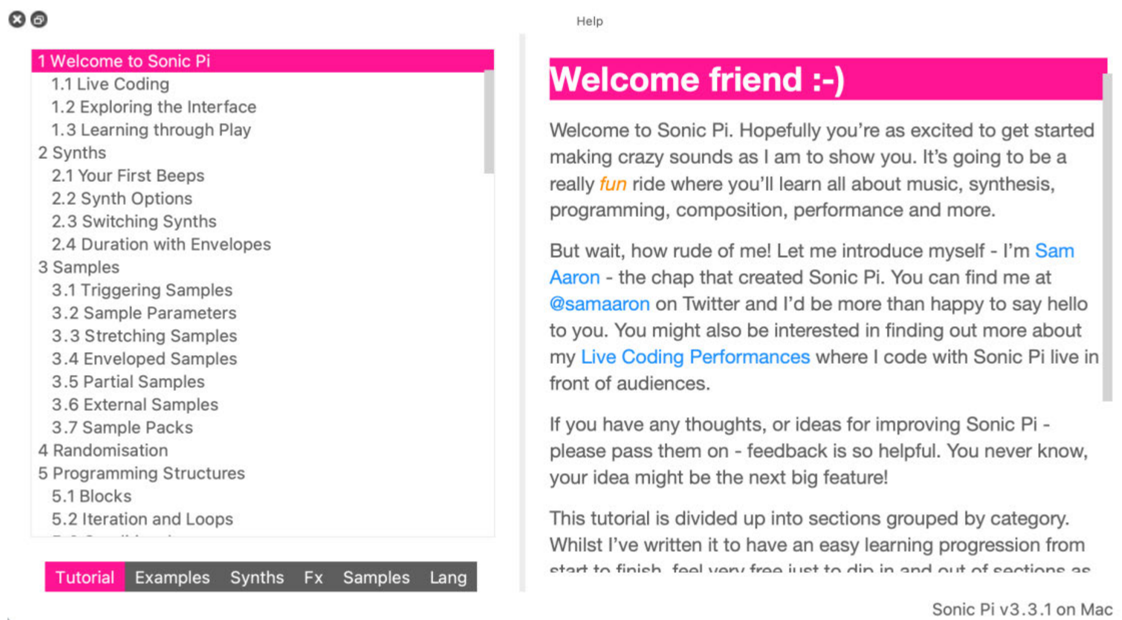


Figure 1. The Sonic Pi in-app tutorial

Participants registered for the research study using an invitation link that was provided electronically. Once enrolled in the research study and logged into Canvas, the Canvas dashboard was visible to participants and included a hyperlink to the research study. The research study was separated into "modules," which are linkable folders in Canvas. Participants were welcomed in module 1, "Statement of Consent/Assent," by a pre-recorded video and asked to complete the consent/assent form.[337] Participants submitted the completed consent/assent form by either uploading a signed digital copy or by capturing the form with

337. The Letter of Consent/Assent is viewable in Appendix 5.

their computer's webcam.

After submitting the consent/assent form, participants gained access to the second module, "Entrance Questionnaire," and answered questions about their experiences with music and computer science. The questionnaire included an optional section on participants' demographics. The third module, "Installing Sonic Pi," provided participants instructions for downloading and installing Sonic Pi on their personal computer. Sonic Pi is multi-platform and may be installed on a Windows, macOS, or Linux machine. At the time of this study, it was not compatible with Chromebook or mobile devices. Once participants installed Sonic Pi, they were able to proceed to the tutorial.

The complete Sonic Pi tutorial consists of 14 sections, including: "1 Welcome to Sonic Pi," "2 Synths," "3 Samples," "4 Randomisation," "5 Programming Structures," "6 FX," "7 Controls," "8 Data Structures," "9 Live Coding," "10 Time State," "11 MIDI," "12 OSC," "13 Multichannel Audio," and "14 Conclusions."[338] For the purposes of the study, I chose to truncate the complete tutorial to the following sections: "1 Welcome to Sonic Pi," "2 Synths," "3 Samples," "4 Randomisation," and "6 FX." I omitted sections that dealt with

338. Sonic Pi, "Welcome to Sonic Pi," Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial.html.

programming and control structures, as well as technical music topics.[339] Focusing primarily on sections with clear musical goals were demonstrated by Aaron and colleagues to be more effective than focusing on lessons prioritizing computer science goals.[340] Moving forward, I refer to the sections within the tutorials as "lessons" and number them sequentially because the sections chosen from the complete tutorial for the study were not in sequential order. The tutorials emphasize musical experimentation and play while utilizing easy-to-understand programming concepts. With this approach, participants did not need knowledge of Western staff notation, a musical instrument, or coding to participate in the study.[341]

While progressing through the lessons, participants submitted artifacts of their learning in Canvas. Artifacts included their code and recordings of their music. Similarly to Collins, I assessed the development of musical ideas and only assessed programming constructs when they were necessary to meet a musical goal.[342] Participants submitted a final cumulative project demonstrating

---

339. Sonic Pi, "Welcome to Sonic Pi."

340. Aaron et. al., "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 86.

341. Puckette, *The Theory and Technique of Electronic Music,* xii.

342. Collins, "Live Coding and Teaching SuperCollider," 11.

meaningful aspects of their learning and answered an exit survey. Following the exit survey, I interviewed participants about their experiences.

Lesson 1: Welcome & Live Coding

Participants began by copying and pasting a code example from the tutorial that utilizes a continuous loop, `live_loop`, a regularly or semi-regularly repeating sequence of sounds.[343] The `live_loop` is a unique programming construct in Sonic Pi that repeats indefinitely until it is stopped intentionally by the coder.[344] The `live_loop` can be modified and reevaluated "on-the-fly."[345] That is, the `live_loop` can be edited while it is running and updated seamlessly without interruption to the music. Participants' first live loop looked like the example in figure 2:

```
live_loop :flibble do
 sample :bd_haus, rate: 1
 sleep 0.5
end
```

Figure 2. Participants' first live loop

In figure 2, `live_loop :flibble do` creates a `live_loop` object called `:flibble`, followed by `do` which starts the code block. Nested within, `sample`

343. Puckette, *The Theory and Technique of Electronic Music,* 71.

344. Aaron, "Live Coding Education," 46.

345. Aaron, "Live Coding Education," 47.

`:bd_haus, rate: 1` loads a sample object called `:bd_haus` with the option `rate:` setting pitch and duration parameters to `1`. The next function, `sleep 0.5`, sounds the sample for half a beat (default 60 bpm) before looping the code. Finally, `end` finishes the code block. Participants were not expected to understand the functional elements of the code, but they were given instructions for which parameters to change and reevaluate/re-run.[346] This is similar to the methods used by Nick Collins to teach SuperCollider, another music programming language.[347] This was done because the goal was to focus on how to navigate Sonic Pi, not the formal rules about how the code was written.[348] For the code in figure 2, participants followed these directions from the tutorial:

1. Make sure the bass drum sound is still running
2. Change the sleep value from 0.5 to something higher like 1.
3. Press the Run button again
4. Notice how the drum speed has changed.
5. Finally, remember this moment, this is the first time you've live coded with Sonic Pi and it's unlikely to be your last...[349]

Instructions like these were present throughout the tutorial. They were intended to give participants safe ways of interacting with their code and to challenge

346. Sonic Pi, "Welcome to Sonic Pi."

347. Collins, "Live Coding and Teaching SuperCollider," 7.

348. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 1063.

349. Sonic Pi, "Welcome to Sonic Pi."

them to connect the changes they make in code to the music they hear.[350]As participants played and experimented, they witnessed the effect their changes had on their music and improved their familiarity with the Sonic Pi user interface.

Lesson 2: Synths

Like Papert's "Turtle Talk" discussed in Chapter 2, participants used commands to make their own sounds.[351] The `play` function sounds monophonic or polyphonic notes and the `sleep` function gives notes a time duration. This was the first opportunity participants had to start creating their own, personalized code. In order to tell Sonic Pi what note to play or how long the duration of the note should be, participants wrote parameters for the functions. The `play` and `sleep` functions each take a single parameter. The `play` function accepts either note name or MIDI number to assign a note to the function.[352] When sounded, the `play` function uses the default `:beep` synthesizer sound. Take notice of the `:` preceding `beep`. Code beginning with `:` are objects, referred

350. Freeman and Magerko, "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch," 68.

351. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 1037.

352. Sonic Pi, "Synths," Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial#section-2.

to as symbols in Ruby, and are built into Sonic Pi.[353] The `sleep` function accepts a time parameter measured in seconds, and written as a float (decimal) value.[354] Figure 3 is an example of the `play` and `sleep` functions:

```
play :C3
sleep 0.5
play :D3
sleep 0.5
play :E4
```

Figure 3. `play` and `sleep` functions

In figure 3, `sleep 0.5` sounds the prior `play` function using the default `:beep` synth for half a beat. Since the last `play` is not followed by `sleep`, it plays for the default note length of one second.

At this point in the tutorial, participants have gained knowledge to sound notes, chords, and sequences in Sonic Pi. Next, participants learned steps to control the amplitude and panning (the stereo position) of their sounds. Controlling amplitude is the "most frequently used operation on electronic sounds."[355] In Sonic Pi, these controls are called options or "opts."[356] Options are

---

353. Objects/symbols are virtual devices in Sonic Pi that users can interact with. Objects may represent something small, like a note name in place of MIDI or frequency value. Objects may also represent larger things, like instruments.

354. Sonic Pi, "Synths."

355. Puckette, *The Theory and Technique of Electronic Music*, 6.

356. Sonic Pi, "Synths."

recognized by having the `:` follow the option name. They are written in line with the `play` function. The amplitude option, `amp:`, accepts a float number between 0.0 (silence) and 1.0 (full volume); however, values beyond 0.5 may cause distortion.[357] The panning option, `pan:`, accepts positive and negative floats between -1.0 (sound is fully in the left speaker), 0.0 (sound is centered between speakers), and 1.0 (sound is fully in the right speaker).

As participants continued, they explored examples of synthesizer objects in Sonic Pi, including: `:prophet`, `:dsaw`, `:fm`, `:tb303`, and `:pulse`, which were selected by writing them as a parameter with the `use_synth` function. Participants generated envelopes for the synthesizer sounds. Envelope generators were used to control the articulation of notes by defining the speed at which a note sounds, its amplitude as it is sustained, and the length of its release. This is the electronic equivalent to the ways in which wind instrument players control their sound with air speed and embouchure, or the ways in which violin players control their sound with bow speed and pressure. This idea is an extension of a larger idea that controls, or parameters, are used in electronic music to communicate our gestures.[358] Participants used a standard ADSR, an acronym

357. Sonic Pi, "Synths."

358. Todd Barton, "6 — Gestures: Sculpting Energy," Buchla Blog, n.d., https://toddbarton.com/2014/03/6-gestures-sculpting-energy/.

for "attack, decay, sustain, and release," which is a 4 stage envelope generator that controls the dynamic shape of the sound.[359] In Sonic Pi, each envelope segment are opts, written as: `attack:`, `decay:`, `sustain:`, `release:`, with each using a float parameter to define the time duration of the stage.[360] Figure 4 shows a visualization of the ADSR envelope.

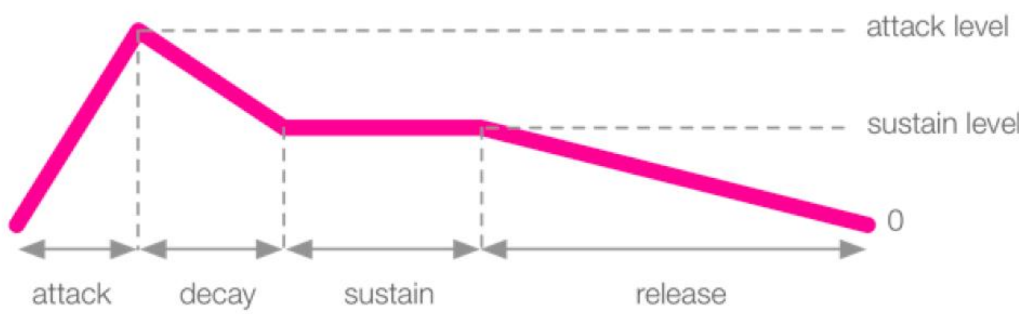


Figure 4. Graphical representation of the ADSR envelope

The amplitude level of the `attack:`, `decay:`, and `sustain:` stages are controllable using the opts, `attack_level:`, `decay_level:`, and `sustain_level:`, respectively.[361] Each option accepts float numbers beginning with 0.0, similar to the `amp:` option. When applied, it looks like figure 5:

```
play 60, attack: 0.1, attack_level: 1, decay: 0.2, sustain_level: 0.4,
sustain: 1, release: 0.5
```

Figure 5. ADSR envelope in Sonic Pi code

359. Puckette, *The Theory and Technique of Electronic Music*, 89.

360. Sonic Pi, "Synths."

361. Sonic Pi, "Synths."

Lesson 3: Samples

Following lesson 2, participants explored samples and creatively applied options to samples in their code. Participants were encouraged to use samples of audio that aligned with their interests, musical or otherwise. Samples are triggered using the function, `sample`, followed by the name of the sample, which could be either a Sonic Pi object preceded with `:` or as a file path to their own custom sample on their computer.[362] Samples accept the same `amp:` and `pan:` options with parameters as synthesizers, along with envelope generator options.[363]

Additionally, participants used the `rate:` option to apply time-stretching like augmentation or diminution to a sample.[364] The `rate:` option accepts a float parameter that works as a ratio to the sample's original value. Using "1" as the parameter: `sample :ambi_choir, rate: 1` will result in unchanged playback of the sample.[365] However, in this example: `sample :ambi_choir, rate: 0.5` the sample playback is half the speed of the original, and sounds an

362. Sonic Pi, "Samples," Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial#section-3.

363. Sonic Pi, "Samples."

364. Puckette, *The Theory and Technique of Electronic Music,* 181-184.

365. Sonic Pi, "Samples."

octave lower.[366] To clarify, `rate:` controls two parameters concurrently, playback speed and pitch.[367] Using a negative value for `rate:` results in a reverse playback of the sample.[368]

Finally, participants were able to playback segments of samples using the options `start:` and `finish:`. Both options use float number values in seconds as a parameter.[369] For example, if given `start: 0.5`, the playback will begin at the halfway point of the sample.[370] In turn, if given `finish: 0.5`, the playback will end at the halfway point of the sample.[371] If the `start:` parameter position is after the `finish:` parameter position, the sample is played in reverse.[372] By utilizing samples, participants were able to remix music aligned with their interests. Participants explored Sonic Pi's built-in sample library, auditioned samples, and remixed them. Participants were encouraged to use external samples and found that Sonic Pi enabled them to remix their favorite sounds to

---

366. Sonic Pi, "Samples."
367. Sonic Pi, "Samples."
368. Sonic Pi, "Samples."
369. Sonic Pi, "Samples."
370. Sonic Pi, "Samples."
371. Sonic Pi, "Samples."
372. Sonic Pi, "Samples."

make their own.

Lesson 4: Randomization

After remixing samples, participants used randomization (specifically pseudo-randomization) for creative effect or to simulate humanness.[373] This concept is rooted in computer science; however, when applied to music, can produce outcomes uncommon in music production and live performances of traditional instruments. The function `rrand`, meaning ranged-random, uses 2 float parameters, which indicate the beginning and end range points.[374] The function `rrand` can be used as an option parameter to choose a random pitch, randomize rhythms, or for randomizing envelope articulations.[375]

In addition to `rrand`, Sonic Pi has other functions to create randomness. First is the function `choose`, which selects a random element within an array (collection of information, for example, note names). This is particularly useful when a programmer wants only random choices from a predetermined collection of values, for example, a scale. Next, `rrand_i` works similarly to `rrand`, however it uses 2 parameter values as whole number integers (not decimals) to

373. Puckette, *The Theory and Technique of Electronic Music,* 27.

374. Sonic Pi, "Randomisation," Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial#section-4.

375. Sonic Pi, "Randomisation."

determine range. This is useful for randomizing parameters that require whole numbers, for example, MIDI notes. Next, `rand` returns a random float value between 0 and 1, which is useful in randomizing float parameters that require a 0 to 1 numerical range like `amp:` levels and ADSR levels. If the starting point in the range is always 0, `rand_i` returns a random whole-number integer using a single parameter for the ending point of the range because 0 is always the starting point.

Additional functions for randomness allowed participants to use familiar applications of chance in their music. A more specialized version of `rand_i` is the function `dice`, which simulates a dice throw by changing the starting number of the range to 1. Finally, `one_in` returns true or false based on the given probability.[376] With the availability of so many functions for randomness, participants could push the boundaries of electronic music by permitting the computer to make musical determinations that we cannot accomplish with traditional Western staff notation and acoustic instruments. Randomization enabled participants to create varieties of drum loops almost instantly, and aleatoric melodies based on ranges of numbers or programmed collections of notes.

376. Sonic Pi, "Randomisation."

Lesson 5: Effects

In the final lesson, participants applied effects to their code. Sonic Pi has several built-in effects which are applied using the `with_fx` function, with the effect name as a parameter. The primary effects participants used in the tutorial are useful for simulating listening environments, for example `:reverb`, and `:echo`. Since there are varieties of listening environments, participants used the options `decay:`, the length of time that it takes for the effect to fade away, and `phase:` to specify the time passed between sounding echoes. Participants nested effects to create effect chains, much like a guitarist's pedalboard.

Both `:reverb` and `:echo` are examples of delay-based effects.[377] Delay-based effects require significant computing power because the algorithms for these effects use several instances of multiplication and addition to repeat duplications of the original sound.[378] With this in mind, participants were guided to use these effects in optimized ways, so that computing power is preserved. Consider the example in figure 6:

---

377. Zhengting He, "How to Create Delay-Based Audio Effects on the TMS320C672x DSP," Dallas, 2005: 9-10. https://www.ti.com/lit/an/spraaa5/spraaa5.pdf.

378. Sonic Pi, "Randomisation"; Zhengting He, "How to Create Delay-Based Audio Effects on the TMS320C672x DSP," 9-10.

```
loop do
  with_fx :reverb do
    play 60, release: 0.1
    sleep 0.125
  end
end
```

Figure 6. Optimized use of effects

In this example, `with_fx :reverb` is written inside the loop, so every time the loop repeats, a new instance of `:reverb` is made. In the figure 7, rather than nest `with_fx :reverb` inside the loop, `loop` is nested within `with_fx :reverb`, creating only a single instance of `:reverb` that is sounded with each repetition of the loop.

```
with_fx :reverb do
  loop do
    play 60, release: 0.1
    sleep 0.125
  end
end
```

Figure 7. A single instance of `:reverb`

Effects give music dimension and transport participants' music from the programming environment to a virtual listing space. Participants determined the size and materials of that space to curate a venue for their music.

Capstone - Your Musical Creation

Once the participants completed the series of tutorials, they were asked to leverage their new knowledge for the creation of new original music that was submitted as a capstone project. Participants were encouraged to use any of the concepts they learned. They were not given specific criteria to follow. Rather, they were encouraged to make use of techniques and procedures that resonated with their creative processes. By learning to make music with Sonic Pi, participants used the lessons learned throughout the tutorial as building blocks to create hierarchies of knowledge which are then present in their capstone projects.[379]

Concluding the Study

Once completing the series of tutorials and capstone project, participants completed a Likert-type exit survey on which they rated their perceptions of the given statements. The statements on the exit survey were informed by the questions on the entrance questionnaire and were used to determine changes, if any, in participants' perceptions. Data from the questionnaire and survey, along with participants' artifacts, were used to guide interviews of the participants. Participants were asked to elaborate on their experiences with Sonic Pi and in some cases, to provide deeper explanations of their answers to the questionnaire and survey. The interviews were the last step in data collection and focused on participants' relationship with music through Sonic Pi, as opposed to learning the

---

379. Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 1099.

technical aspects of Sonic Pi.[380]

380. Papert, *Mindstorms: Children, Computers, and Powerful Ideas,* 1176.

CHAPTER 5: FINDINGS

The purpose of this study was to investigate perceptions of high school students who made music with code in Sonic Pi. Participants completed research study materials, including a series of tutorials for Sonic Pi, on-site at extracurricular club meetings and remotely through a Canvas LMS course. During the course of the study, I collected answers to questionnaires and surveys, multimedia artifacts including the source code, exported audio of participants' music making, and interviewed participants. I then codified and analyzed the data I collected in two cycles, utilizing descriptive coding, values coding, and longitudinal coding.[381] I am interested in knowing which, if any, musical ideas participants reported learning or demonstrated through coding music with Sonic Pi and if coding music impacted participants' views of their ability to learn music. I am also interested in knowing if music making with code impacted students' interest in continued studies in music and/or computer science.

Participants in this study were three students who were members of an extracurricular computer club at a public charter high school. Participants were given the research study materials outlined in CHAPTER 4: PROCEDURE OF THE STUDY via Canvas LMS. Participants were encouraged to progress

---

381. Saldaña, *The Coding Manual for Qualitative Researchers*, 134, 167, 332.

through the research study materials and to create new (concretize) musical artifacts from the research study materials.[382] At the conclusion of the study, all participants created musical artifacts, and offered their perceptions of making music with Sonic Pi by participating in a questionnaire, survey, and interview.

Research Question 1: What musical ideas, if any, do participants report learning or demonstrate through making music with code in Sonic Pi?

Participants completed five Sonic Pi lessons on the following topics: Welcome & Live Coding, Synths, Samples, Randomization, and Effects. In addition, participants were encouraged to explore further learning resources if they desired. After the completion of these five lessons, participants were able to create their own songs in Sonic Pi. Figure 8 demonstrates a simple melody that participants might create in Sonic Pi, "Au claire de la lune."

---

382. Hancock, "Play-Based, Constructionist Learning of Pure Data: A Case Study," 94; Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 435, 636.

```
1   use_bpm 100
2   use_synth :piano
3   with_fx :reverb do
4     2.times do
5       3.times do
6         play :C4, amp: rrand(0.7, 1.0)
7         sleep 1
8       end
9       play :D4, amp: rrand(0.7, 1.0)
10      sleep 1
11      play :E4, amp: rrand(0.7, 1.0)
12      sleep 2
13      play :D4, amp: rrand(0.7, 1.0)
14      sleep 2
15      play :C4, amp: rrand(0.7, 1.0)
16      sleep 1
17      play :E4, amp: rrand(0.7, 1.0)
18      sleep 1
19      play :D4, amp: rrand(0.7, 1.0)
20      sleep 1
21      play :D4, amp: rrand(0.7, 1.0)
22      sleep 1
23      play :C4, amp: rrand(0.7, 1.0)
24      sleep 4
25    end
26  end
```

Figure 8. "Au claire de la lune" in Sonic Pi

For the reader's benefit of understanding the case findings, I will draw correlations and comparisons between figure 8, written in Sonic Pi, and figure 9, written in Western staff notation. Sonic Pi, like many programming languages and environments, reads through code from left to right, top to bottom to create music or sound as instructed in the algorithms provided by a user. In figure 8, line 1 establishes the tempo of the program by setting the beats per minute to 100. This means that each of the following lines containing the `sleep` function with a duration of 1 is equal to a quarter note at 100 beats per minute. Similarly, the Western staff notation in figure 9 also establishes the tempo above the staff at a speed of 100 quarter notes per minute. Line 2 in figure 8 establishes the instrument that will play the notes written below it, the piano. The same task is accomplished in figure 9 by preceding the staff with the name of the instrument, piano. Beginning on line 4 in figure 8, the sequence of notes is surrounded by a loop that will repeat the entire melodic sequence twice, working in the same manner as the repeat symbol in the Western staff notation in figure 9. Each note in figure 8 is sounded with the `play` function, followed by the `sleep` function on the next line to set the length of time until the next note is sounded. In the Western staff notation of figure 9, the melodic sequence is written on a staff, with lines and spaces specifying which notes to play for a given clef, such as the treble clef noted in the figure below. Note durations are indicated by the type of notehead written on the staff. Figure 8 concludes with commands ending sections of code: Line 25 in figure 8 uses `end` to conclude the `2.times do` section beginning on line 4 and line 26 uses `end` to conclude the `with_fx :reverb do` section beginning on line 3. In figure 9, the Western staff notation example ends after the second repetition.

While several correlations exist, there are some notable differences. Line 3 in figure 8 demonstrates the use of reverb in Sonic Pi. In Sonic Pi, reverb is customizable to simulate a variety of listening environments. There is not an indicator in Western staff notation for reverb because it is dependent on the performer, instrument, and performance venue. In figure 8, the ranged-random function, `rrand()`, is used to randomize the amplitude of the notes called by the `play` function; this simulates a performer adjusting the volume of each note by randomly playing a note between 70% (0.7) and 100% (1.0) of maximum volume. Although a human performing this melody could also randomize how loud or soft they perform each note, Sonic Pi can follow these specific rules with a level of precision and consistency that could not be replicated by humans. In figure 9, clef and time signature are written to indicate the performance range and meter of the song, respectively. In figure 8, there are no codes to indicate clef or time signature, and the notes themselves are called by either MIDI number or note name and octave, negating the need for a clef or meter. While Sonic Pi can replicate the information provided through Western staff notation, it affords a level of specificity and granular control that rivals what producers have through Digital Audio Workstations (DAWs) like Avid Pro Tools or Ableton Live.

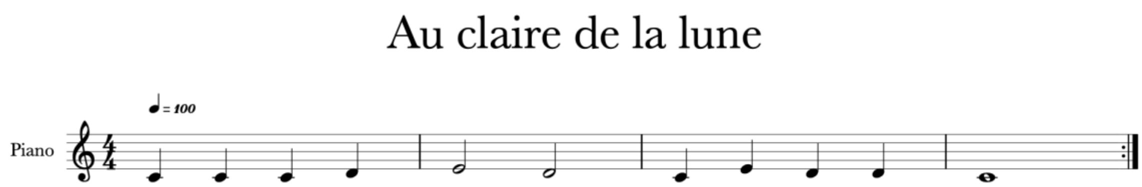


Figure 9. "Au claire de la lune" in Western staff notation

The Four Properties of Sound

Findings that developed in relation to the first research question closely aligned with the "four properties of sound."[383] The four properties of sound are: pitch, duration, intensity/amplitude, and timbre. While the four properties of sound serve as an organizational framework for presenting findings from this study, additional findings emerged, including form, non-traditional notation, and randomization. Like Collins, my assessments and codification focused on the development of musical ideas as participants progressed through lessons of the study with focus given to programming constructs when they were necessary to meet a musical goal.[384]

---

383. Bruce Benward and Marilyn Saker, *Music in Theory and Practice* (Vol. 1), McGraw-Hill Higher Education (New York, NY, 2009), xiv-xvi.

384. Collins, "Live Coding and Teaching SuperCollider," 11.

Pitch

Pitch is the degree to which a sound or a note's frequency is audibly high or low.[385] Pitch using Sonic Pi synths are represented by either note name, represented as a symbol (Ruby language object) preceded by `:`, or as a MIDI note number as defined by the MIDI Standard. For example, the middle C on a piano could be written as `60` or `:C4` in Sonic Pi. The participants in this study made extensive use of note names and MIDI note numbers as musical notation.

```
play 60
sleep 0.5
play 70
sleep 0.5
play 80
sleep 0.5
play 70
sleep 0.5
```

Figure 10. Excerpt from *Case 1, Capstone Artifact* (see Appendix 16: Case 1, Capstone Artifact for complete example)

```
play 72
sleep 0.5
play 75
sleep 0.5
play 79
sleep 0.5
```

Figure 11. Excerpt from *Case 3, Lesson 2 Artifact* (see Appendix 25: Case 3, Lesson 2 Artifact for complete example)

Figure 10 and figure 11 are examples of note sequences written as MIDI note numbers with the `sleep` function following each note to express note

385. Benward and Saker, *Music in Theory and Practice*, xiv.

duration of half a count (i.e., an 8th note). Figure 12 is a transcription of the example in Western staff notation. Line 26 in figure 10 calls `play 60`, in which `play` means to sound a note and 60 is the MIDI note number for "middle C" on a piano keyboard. The remaining notes in the excerpt are called on lines 28, 30, and 32, calling A#/Bb, G#/Ab, and A#/Bb respectively.

Figure 12. Excerpt from *Case 1, Capstone Artifact* in Western staff notation

In figure 13, the participant makes use of a programming construct, called an array, to organize notes into a group, much like a repeating ostinato pattern. In this example, the participant identifies the notes on line 15 by MIDI note names by using a `:` followed by the note name (sharp notes are indicated by a lowercase "s" while flat notes are indicated by a lowercase "b"), followed finally by the octave of the note. For example, the notation `:Fs5` represents the F# above "middle C" on a piano keyboard. This group of notes is stored into a variable so it may be called or performed later in the program, much like a composer may use and reuse a leitmotif or ostinato pattern throughout a piece. Line 16 contains a group of durations that when called concurrently with the variable on line 15, will sound each note with the corresponding durations. If a composer using Western staff notation software decided to modify a repeated pattern used throughout a composition, they would have to find and manually adjust every instance of that phrase; however, the use of arrays in Sonic Pi as seen in figure 13 makes it so the user only has to adjust one line of code to modify every use of these variables.

```
b_background_music = [:A4, :D5, :A5, :A4, :D5, :A5, :Fs5]
b_background_timing = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]
```

Figure 13. Excerpt from *Case 2, Capstone Artifact Second Submission* (see Appendix 23: Case 2, Capstone Artifact Second Submission for complete example)

Figure 14 is an example of figure 13 in Western staff notation. In Western staff notation, the pitch of the note and duration are indicated with a single note placed on the staff.

Figure 14. Excerpt from *Case 2, Capstone Artifact Second Submission* in Western staff notation

Participants demonstrated sequential and concurrent performance of note names and MIDI note numbers, resulting in the use of pitch to create a variety of melodies and harmonies. Participant 1 indicated in their interview that their knowledge of pitch was improved from this experience. Participant 2 expressed in their interview that their ability to read Western musical notation was improved from this experience. Participant 3, in their interview, was excited to see graphical representations of pitch, which they related to their knowledge of mathematics. In addition to demonstrating knowledge of pitch, participants expressed their knowledge in their interviews and for participants 2 and 3, connected this new knowledge to prior knowledge.

Duration

Duration is the length of time that a sound is audible or the silence between sounds.[386] In lesson 1, participants first learned how to perform sound or silence duration in Sonic Pi by calling the `sleep` function. Users also had granular control over the shape of each sound's duration through an envelope generator; for example, an ADSR, an acronym for attack, decay, sustain, and release. Envelope generators are used to articulate the duration or length of a note with varied amplitudes; it is a way to code precise articulations. In Sonic Pi, participants regularly utilized the `sleep` function and envelope generators to adjust the duration of, and between, each sound.

386. Benward and Saker, *Music in Theory and Practice*, xiv.

```
1 live_loop :foo do
2   play 60, amp: 2
3   play 70, amp: 0.5
4   sleep 0.5
5   play 50, amp: 3
6   sleep 0.25
7   play 60, attack: 0.5, attack_level: 1, decay: 1, sustain_level:
  0.4, sustain: 2, release: 0.5
8   sleep 1
9 end
```

Figure 15. *Case 1, Lesson 2 Artifact*

In figure 15, the note sequence is wrapped within a `live_loop`, beginning on line 1 with `do` and ending on line 9 with `end`, repeating the sequence indefinitely, until the program is stopped. Lines 2 and 3 in figure 15 sound two notes simultaneously, each with their own `amp:` that will be discussed in the next subsection. On line 4, the `sleep` function sets a time duration of 0.5 or half a count, transcribed as an eighth note in figure 16, until the `play` function on line 5 is sounded. On line 6, the `sleep` function sets a time duration of 0.25 or quarter of a count, transcribed as a sixteenth note in figure 16, until line 7 is sounded. Line 7 is written with an envelope generator option. The options on line 7 together create an ADSR envelope. The attack (A) and sustain (S) stages of the envelope include options for amplitude (written with `_level` after the stage name), which will be discussed in greater detail in the following subsection. The amplitude (`_level`) options indicate the loudness of the given stage. Each

ADSR stage has a parameter for time duration before progressing to the next stage or option. The total time elapsed for the entire ADSR envelope is four counts. This means that line `7` in figure 14 will continue to sound after the `sleep` functions progress the program, creating a modulating drone effect. This kind of notation is difficult to recreate with Western staff notation. In figure 16, the final note duration is equal to a whole note; however, rhythmically it is only equal to a quarter note before repeating the sequence of notes.

Figure 16. *Case 1, Lesson 2 Artifact* in Western staff notation

In figure 17, an attack release (AR) envelope is used. The `attack` stage of this envelope sets the amplitude to `2`. The release stage of the envelope is timed and releases the note at `0.2` of 1 count. This note will be released before the `sleep` function progresses the program.

```
use_synth :saw
play 72, attack: 2, release: 0.2
sleep 1
play 75
sleep 1
play 79
sleep 1
play 82
```

Figure 17. Excerpt from *Case 2, Lesson 2 Artifact*

Meter

Meter describes the regularly recurring pulses of sound in equal duration.[387] The meter of the program or composition is determined by iterative patterns and the length of sound and silence collections. In Sonic Pi, meter is not defined explicitly. Rather, it is implicitly defined by iterative patterns, as with figure 15 using the `live_loop`. In figure 18, the time signature is added to the Western staff notation. The addition of a time signature in Western staff notation often implies rhythmic groupings. Note the beaming of the first two notes in the figure below. Beaming notes together into groups places an emphasis on certain beats. In this figure, emphasis is placed on beats 1 and 4.

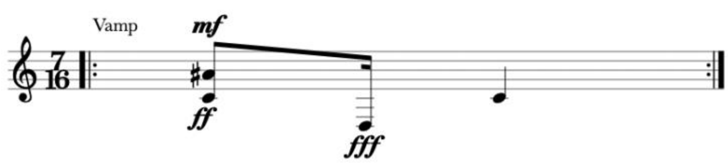


Figure 18. *Case 1, Lesson 2 Artifact* in Western staff notation including the 7/16 time signature

Another means of implying meter in Sonic Pi is by defining variables that hold groupings of notes, like the arrays in figure 13 from participant 2. Two arrays are defined, each containing the same number of items, called indexes. When these arrays are called concurrently in a regular recurring pattern, the

387. Benward and Saker, *Music in Theory and Practice*, xiv.

length of indexes may reveal an intended meter.

Probability is another means of implying meter. In figure 19 samples are triggered based on probability. Lines 7–10 call samples of instruments within a drum set. They will only sound when the probability statement of an `if` condition is true. For example, line 7 contains the probability statement `one_in(6)`, meaning that this statement will return true with a probability of 1/6 and return false with a probability of 5/6. On line 7, the `sample :elec_hi_snare` will only sound when the probability statement is true.

```
1  live_loop :multi_beat do
2    use_random_seed 2000
3    8.times do
4      c = rand(70, 130)
5      n = (scale :e1, :minor_pentatonic).take(3).choose
6      synth :tb303, note: n, release: 0.1, cutoff: c if rand < 0.9
7      sample :elec_hi_snare if one_in(6)
8      sample :drum_cymbal_closed if one_in(2)
9      sample :drum_cymbal_pedal if one_in(3)
10     sample :bd_haus, amp: 1.5 if one_in(4)
11     sleep 0.125
12   end
13 end
```

Figure 19. *Case 3, Lesson 1 Artifact*

Each of the participants demonstrated the use of duration in their artifacts, including the `sleep` function to pass time and envelope generators to code precise articulations. Extended techniques, including arrays and probability were also used. Their use of duration was frequent throughout the course of study.

Intensity/Amplitude

Intensity, or amplitude, is the audible loudness of a sound or a note.[388] In Sonic Pi, amplitude is specified on a scale of 0.0, or silence, to 1.0, or unity, the loudest a signal can be before distortion. Sonic Pi refers to unity level as "normal volume."[389] It is possible to specify amplitudes greater than 1.0, though it should be done with the understanding that the sound may distort. To avoid distortion, Sonic Pi automatically applies compression to amplitude levels above 1.0.[390] Compression is the reduction of an audio signal above a given threshold.

Participants used amplitude to control the dynamic contrast of their music or to create articulations by controlling the shape of envelopes. In figure 15 above, `amp:` options were used to define amplitude levels for the notes sounded on lines 2, 3, and 5. On line 2 of figure 15, the amplitude level is `2`, exceeding unity level meaning that it is possible this note will distort. It is interpreted at *ff* (*fortissimo* - very loud) in figures 17 and 18. This interpretation is generalized, and as typical with Western staff notation, is subject to ensemble instrumentation and performance venue. In Sonic Pi, the amplitude level is

---

388. Benward and Saker, *Music in Theory and Practice*, xiv.

389. Sonic Pi, "Synths."

390. Sonic Pi, "Synths."

absolute, and is interpreted the same way each time a program is run, regardless of the system with which it is performed. On line 2 of figure 15, the amplitude level is `0.5`, interpreted as *mf* (*mezzo-forte* - medium-loud) in figures 17 and 18 which is below unity level. On line 5 of figure 15, the amplitude level is `3`, interpreted as *fff* (*fortississimo* - very very loud) in figures 17 and 18, which is significantly above unity level. Participants' artifacts do not distort because Sonic Pi automatically applies compression to amplitude levels above unity.

As mentioned in the prior subsection, amplitude (`_level`) options indicate the loudness of given stages of an envelope. Also mentioned were the attack (A) and sustain (S) stages of the envelope on line 7 of figure 15. The options `attack_level` and `sustain_level` specify amplitude levels of `1` and `0.4`, respectively. Figure 20 is a graphical representation of the envelope generator represented on line 7 of figure 15.

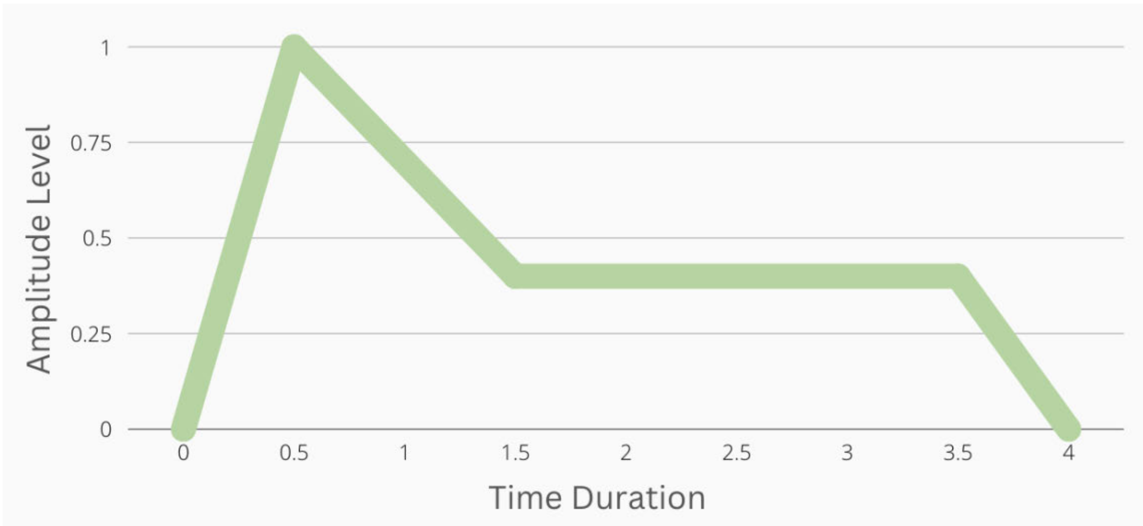


Figure 20. *Graphical Representation of the ADSR Envelope,* Case 1, Lesson 2 Artifact, Line 7

Each participant wrote code to edit amplitude levels, both for dynamics and envelopes. Participant 3 noted in their interview that they identified these concepts as programming concepts but related them to music when they achieved a "spacey" effect with them. Some participants' artifacts demonstrated amplitude levels above unity, which were automatically compressed by Sonic Pi. In these instances, participants may have had difficulty discerning the differences in amplitude levels above unity. For amplitude levels between 0.0 and 1.0 in Sonic Pi, the differences are audible.

Timbre

Timbre is the audible character or color of a sound; it is the property of sound that allows us to distinguish between different sound sources or instruments.[391] In Sonic Pi, virtual instruments, called "synths" can be layered with other virtual instruments or samples of recorded sound. Sonic Pi contains several virtual instruments and samples in its built-in library for the user to choose from.

Throughout the duration of the study, each participant demonstrated unique and expressive layering of virtual instruments and samples, creating new timbres. In the following paragraphs are three different examples of timbre combinations.

```
1  live_loop :foo do
2    sample :ambi_choir, rate: 0.5
3    sample :perc_impact1, rate: 2
4    sleep 1
5  end
6
7  live_loop :lol do
8    sample :bd_haus, rate: 1
9    sleep 2
10 end
```

Figure 21. *Case 1, Lesson 1 Artifact*

In figure 21, two `live_loop`s will play back concurrently. On lines 2, 3, and 8, `sample` objects are called. Since the `live_loop`s play back concurrently, we will hear all of the samples sounding at the same time. Each `sample` is given a `rate:` option. On line 2, the `rate:` is `0.5`. This will play back the sample at

391. Benward and Saker, *Music in Theory and Practice*, xiv.

half its recorded speed. On line 3, the `rate:` is `2`. This will play back the sample at twice its recorded speed. The sample on line 8 is given a `rate:` of `1`. This will play back the sample at its original recorded speed. Each of these options adjusts not only the speed, but the timbre of the sounds being played by altering the frequency of the sample. When combined, this creates an analog-like effect in which the frequency changes with speed. When the sample is sped up, it will also sound higher in frequency. When the sample is slowed down, it will sound lower in frequency.

In figure 22, the `in_thread` function on line 30 is used to play the `play_pattern_timed` functions on lines 31 and 34 concurrently, so that the `music` and `background` melodies play at the same time. The `play_section` function that wraps this section of code is later called on line 42. These melodies playing at the same time creates harmony.

```
define :play_section do |music, timing, background, background_timing, repeat|
  in_thread do
    play_pattern_timed music, timing
  end
  repeat.times do
    play_pattern_timed background, background_timing
  end
end

define :haggstrom do
  2.times do
    intro
  end
  play_section a_music, a_timing, a_background_music, a_background_timing, 4
  /play_section b_music, b_timing, b_background_music, b_background_timing, 1/
end

use_synth :kalimba
haggstrom
```

Figure 22. Excerpt from *Case 2, Capstone Artifact Second Submission* (see Appendix 23: Case 2, Capstone Artifact Second Submission for complete example)

In figure 23, a `:tb303` virtual instrument is layered with four samples, `:elec_hi_snare`, `:drum_cymbal_closed`, `:drum_cymbal_pedal`, `:bd_haus` on lines 6-10. These lines will all play concurrently at a `sleep` time duration of `0.125` given on line 11 and repeat indefinitely until the program is stopped because it is wrapped with a `live_loop`. I will explore figure 23 further in the subsection Randomization to unpack the randomization and conditional statements.

```
live_loop :multi_beat do
  use_random_seed 2000
  8.times do
    c = rrand(70, 130)
    n = (scale :e1, :minor_pentatonic).take(3).choose
    synth :tb303, note: n, release: 0.1, cutoff: c if rand < 0.9
    sample :elec_hi_snare if one_in(6)
    sample :drum_cymbal_closed if one_in(2)
    sample :drum_cymbal_pedal if one_in(3)
    sample :bd_haus, amp: 1.5 if one_in(4)
    sleep 0.125
  end
end
```

Figure 23. *Case 3, Lesson 1 Artifact*

Effects have a considerable impact on timbre. Effects may be used to simulate an acoustic environment, or they may be used to creatively alter timbre. All participants utilized effects such as `:reverb` and `:wobble` to accomplish both of these approaches.

In figure 24, effects were used both to simulate an acoustic environment and for creativity. The entire body of code is wrapped with the `:reverb` object, which applies reverb to the entire program. Reverb simulates a listening environment, like a room or an outdoor setting. Nested within the `:reverb` object is the `:wobble` object, which is a modulating filter frequency cutoff that uses the `phase:` option.

```
with_fx :reverb do
  with_fx :wobble, phase: 0.5, decay: 8 do
    play 60
    sleep 0.5
    sample :ambi_drone
    sleep 0.5
    play 62
    sleep 0.5
    sample :ambi_piano
    sleep 0.5
    play 60, rate: -0.5
  end
end
```

Figure 24. *Case 3, Lesson 5 Artifact*

In Sonic Pi, participants demonstrated control of timbre, rather than timbre being an artifact of an instrument, performer, or performance space. With this level of granular control, timbre can be composed using code.

Form

Form is the organization of music into discernable parts. Form is also a mechanism to engage listeners by introducing related, yet different sections of music that may have changes to one or several of the following: melody, harmony, timbre, rhythm, meter, etc. In Sonic Pi, form can be arranged by using conventional programming structures.

Participants used conventional programming control structures, including: loops/iteration, function definitions, and function calls to create musical forms. In figure 22 above, the function `haggstrom` is called on line 47. This function runs the entire program and is defined beginning on line 38. Within this function, the overall form of the song is given. We will first hear the intro played twice because it is wrapped inside the for loop, 2.times do on lines 39-41. Line 42 calls the function play_section, which is defined on line 29 of the program. This function will play the remainder of the song. Line 43 will not play because it has been commented-out with a /, which is a way for users to indicate to Sonic Pi that a line of code should be ignored. This feature is especially useful during a live performance with Sonic Pi, as it allows users to prevent a line of code from running without having to delete the line of code.

The play_section function passes 5 parameters, specified between the pipe characters (|) on line 29. These 5 parameters each have specific instructions within the function. When the function is called, variables are given for the 5 parameters. On line 49, the variables a_music, a_timing, a_background_music, a_background_timing, 4 are given. To understand how these parameters are used, we need to look at the procedure (specific instructions) within the play_section function definition beginning on line 29. On line 30, the in_thread function is called. This will play back the code wrapped within the in_thread concurrently with the following code inside the play_section function definition, up to line 36. On line 31, play_pattern_timed (defined earlier in the program) is called to play music and `timing`. The parameters on this line of code are passed from the variables given on line 42. Similarly, the remaining parameters from line 42 are passed to lines 33-34. Here, a value of `4` is passed to `repeat` on line 33. On line 34, `a_background_music` and `a_background_timing` are passed to `background` and `background_timing` respectively.

Using control structures in programming to create musical form makes sense. Control structures are used to define specific sets of instructions in a particular order. In this regard, control structures are similar to Western staff notations that indicate the sequence of musical parts. Each participant demonstrated musical form in their artifacts by using control structures to specify instructions in an intentional order.

Non-Traditional Music Notation

In many schools of music in the United States, Western staff notation is the standard for notating acoustic musical instruments.[392] During the 20th-century, with the emergence of serialism, minimalism, experimental music, and electronic music, Western staff notation was expanded to accommodate the varieties of new compositional techniques and tools. By the mid-20th century,

---

392. John Kratus, "A Return to Amateurism in Music Education," *Music Educators Journal* 106, no. 1 (2019): 33, https://doi.org/10.1177/0027432119856870.

computers were being tasked by avant-garde composers with producing audible sound. Computers require musical input (notation) that begins with high-level, human-readable code, and is then compiled into machine code, and finally binary data that can be run by a computer. Fundamentally, this process has not changed in the 21st century, though it has become much more streamlined. For this study, we used Sonic Pi as the musical input. Sonic Pi's high-level, human readable code was well-suited for the participants.

At the conclusion of the study, one participant reported in their interview that they were able to change sound using math in Sonic Pi. This participant described math as a way to visualize sound, and that using math to make music was unexpected, stating, "the crossover of math and music is what I remember." The code in Sonic Pi, and other music programming languages, including mathematics to a degree, are examples of non-traditional music notation. To notate music in Sonic Pi is accessible to this participant, they stated: "I know how to code [music] and I can just code a little music, you know that's awesome!" This participant related mathematical concepts to engineering, which for them was a relevant and meaningful connection to music.

Randomization

Randomization is the unpredictable arrangement of given musical elements. In Sonic Pi, there are several ways in which one can randomize notes (to create melodic sequences), envelopes (both amplitude of stages and stage duration), or effects. Randomization is problematic for human performance. While humans can attempt to randomize an arrangement of things, this is a process much better suited for a computer. Computers can handle randomization with significantly greater efficiency and consistency than a human because they are not hindered by rhythmicity or other limitations.[393]

Participants made use of Sonic Pi's randomization functions to create melodic sequences, create variable envelope stages, and to randomize effects, specifically their amplitude envelopes. In figure 25, a short melodic sequence is played using a variety of randomization functions. Rather than play a specific note on line 2, the note parameter is replaced by the `rrand()` (ranged random) function. The `rrand()` function is given parameters of `30` and `90`, meaning that it will randomize note selection between MIDI notes 30 and 90. On line 4, the `play` parameter is replaced by the `choose()` function. The `choose()` function

393. Michael D. Greenfield et al., "Synchrony and Rhythm Interaction: From the Brain to Behavioural Ecology," *The Royal Society*, 2021, 1, https://doi.org/10.1098/rstb.2020.0324.

is given a list of MIDI notes to choose from as its parameter: `20`, `50`, and `80`. The `play` function on line 4 will only sound one of these three notes during each iteration of the `loop` that wraps the program. Finally on line 6, the `play` parameter is replaced by the `rand()` function, which with no parameter given, will sound a completely random MIDI note, sometimes outside the audible range of human hearing.

```
1  loop do
2    play rrand(30, 90)
3    sleep 0.5
4    play choose([20, 50, 80])
5    sleep 0.5
6    play rand
7    sleep 1
8  end
```

Figure 25. *Case 1, Lesson 4 Artifact*

Finally, I will revisit figure 23 above because of the extensive use of randomization in the program. Inside the `live_loop` on line 2, the function `use_random_seed` is called with a parameter of `2000`. This function replaces the random number generator to use a specified seed, which has been specified as 2000. With a specific seed, the randomized note sequence can be reproduced or changed with the same or different number. In other words, Sonic Pi will play the randomized results the same way each time the seed number is set to 2000; however, it will play a different outcome of consistent, randomized parameters if the seed number is set to 1999. On line 4, variable `c` stores a ranged random

MIDI note between the values of `70` and `130`. This value will be selected from the seed 2000. On line 5, variable `n` stores an array of 3 indexes with the `take()` function from a minor pentatonic scale starting on E. The `choose` function is used to randomly choose a value from the 3 indexes. On line 6, the `synth :tb303` is sounded with the value of variable `n` and cuts off with the ranged random number stored in variable `c` so long as the random value following is less than `0.9`, which it will always be with the given seed of 2000. On lines 7-10, each `sample` is followed by an `if` condition, then the probability statement, `one_in(`*`given_number`*`)`. With each iteration, the `one_in()` function will return the same probability statement with the given seed of 2000. If you, the reader, are listening to these examples in Sonic Pi, try commenting-out line 2 by preceding the statement with `#`. Notice the significant change without using a seed.

Participants made extensive use of randomization. Participant 3 exclaimed in their interview how they were able randomize volume which was “really cool.” Randomization is an easy, yet powerful way to create variations in music. Participants were able to demonstrate variations in many aspects of their music, including: melodies, amplitude, and effects.

Research Question 1: Summary

Participants reported and demonstrated a variety of musical ideas throughout the course of the study. Many of the ideas reported and demonstrated extended well beyond the four properties of sound, and well beyond standard conventions in Western staff notation. Please note that I did not provide Western staff notation interpretations for several of the figures above. This is because standardized writing conventions for these examples, particularly for notating samples or randomization, do not exist in Western staff notation. Participants' use of melodies, made up of notes at varieties of lengths and amplitudes, along with envelopes, samples, and randomization, was written in code as a novel, non-traditional notation for music.

Research Question 2: How does making music with code impact participants' perceptions of their music making?

Sonic Pi is an integrated software instrument and programming environment for creating music. Sonic Pi was developed for the Raspberry Pi Foundation to be an educational platform for learning to code within a coding environment, in which the user performs and composes concurrently. Participants created artifacts that explored coding while learning the Sonic Pi/Ruby programming language. At the conclusion of the study, the participants were asked if coding music in Sonic Pi impacted their perceptions of their music making.

Participants Made Music with Code

All participants reported on their exit surveys that they either strongly agreed or somewhat agreed that the artifacts they created during this study were music. Participant 1, while having stated on their exit survey that it was somewhat untrue they were a composer, reported having composed music in Sonic Pi in their interview. They stated, "at that point I felt I had composed music in Sonic Pi, so it could really no longer be entirely untrue." Participant 3 reported in their interview that it is easy for anybody to make music using Sonic Pi. This participant stated that, "I don't know how to read sheet music, and I'm not very musically inclined. I thought [Sonic Pi] would be hard, but it wasn't because I didn't need to know music to do it." For this participant, Sonic Pi was a medium that permitted them to create music, without the prior knowledge of Western staff notation or performance skills with a musical instrument.

In order to make music with code, all three participants connected prior knowledge, from either music or computer science, to create music through code within Sonic Pi. Participants 1 and 2 indicated on their entrance questionnaires that they read Western staff notation and participated in school music, which may have informed their music making with Sonic Pi. Evidence of this was demonstrated by participant 2 by their use of MIDI note names in figure 13. Additionally, participants 1 and 2 have studied computer science, which may have also informed their music making. For example, having a stronger foundation in computer science concepts and practices may have enabled both participants to focus on learning music concepts and practices that were unfamiliar to them. Participant 3 indicated on their entrance questionnaire that they did not read Western staff notation; however, they had studied computer science and were familiar with several programming languages. This participant may have depended on their knowledge of computer science to concretize their music. In their interview, they referenced measuring sound waves mathematically and graphing them, and by this it was possible to encode music.

Musicianship is Multifaceted

Of the three participants, participants 1 and 2 stated on their exit surveys it was somewhat true they considered themselves musicians and elaborated on this notion in their interviews. Participant 1 stated that musicians do more than perform, they create. They stated, “I interpret musicians as less of somebody who like casually as a hobby plays an instrument, and more does it more like with more focus and creates. I feel like a musician incorporates more of the creation like composing, and something similar to that. I lack those qualities.” Participant

2 agreed that being a musician was more than engaging in music as a hobby, stating that for them, "[music is] more of a hobby that's cool." Participant 3 stated on their exit survey that it was untrue that they considered themself a musician. To them, they "weren't really a music person," although they enjoyed the experience and making music.

Both participants 1 and 2 felt that musicianship is not a casual endeavor. Musicianship to them involves dedicated studies in music and creative output. For participant 3, musicianship is more tied to one's perception of themself. One could be a musician if they identified themself as a "music person."

Composing Music Doesn't Necessarily Make You a Composer

Despite having composed music, participant 1 did not consider themself a composer. To this participant, composing is something that is done regularly and with great interest. They stated, "So I have composed technically but I don't consider myself a composer because I don't do it consistently...It just has to be done with more interest than with more intensity and on a more regular basis than I had done." Participant 2 stated in their exit survey it was somewhat true they considered themself a composer; however, they also noted that regularity was a trait of a composer, and additionally, a desire to grow and learn. They stated, "I [experimented] with various music software for example, like I'm pretty sure one is called Reaper another one's called, well, Garage Band. I've been just making music using that. It's sorta an on and off thing as mentioned previously...but like just a hobbyist, like I'm not actively trying to learn, it's just that I open it up sometimes and then just experiment." Participant 3 indicated on their exit survey that it is untrue that they are a composer and stated that a passion for music is requisite of a composer. They stated that their passion was not music, but rather engineering and coding. They said, "another participant who's really into music, but doesn't really play any instruments, might have viewed himself as a composer afterwards." Participant 3 also stated during their interview that making music with Sonic Pi was a fun experience, and that while they do not consider themself a composer, they enjoyed making music.

To the participants, composing is a part of musicianship that is tied to musical growth. It is a desire to pursue and grow as a musician that leads to composing. Participants also appear to describe their perceptions of making music in stages. First, there is the hobbyist, a person who is interested in music. Second, one may be a musician with dedication and an affirmed self-identity as a musician. Lastly, composing is intentionally pursued by musicians to further their musical art.

Research Question 2: Summary

The participants' perceptions of their music making were highly influenced by their prior experiences and perceptions of musicianship and composing. All three participants agreed that the artifacts they created for this study were music, even if they did not consider themselves a musician or composer. Interestingly, through participants' perceptions of themselves, one does not need to identify as a musician or a composer to make music.

Research Question 3: How does making music with code impact participants' perceptions of their ability to learn to make music?

Learning to make music may be associated with performance ensembles or with musically-talented individuals. However, for this study, I approach learning to make music as inclusive of all learners, especially those prejudged as "untalented."[394] Participants' perceptions of their abilities to learn music making was directly impacted by their experiences with Sonic Pi. Regardless of their musical experiences or knowledge, participants were able to make music with Sonic Pi. Participants with prior musical knowledge reported improving their musical skills with Sonic Pi. Participants also reported their perceptions of music making changed throughout this study. Some participants reported that

---

394. Deutsch, "Where Was Technology and Music Education Twenty Years Ago?," 93.

knowledge of Western staff notation was a perceived barrier to music making, however they were able to work around that barrier when creating music with Sonic Pi.

Learning and Reinforcing Musical Understandings through Sonic Pi

All three participants reported in their exit surveys and interviews that they learned and improved musical skills through Sonic Pi. Participant 1 reported in their interview that they have a better understanding of pitch and intervals, "it increased my knowledge about pitch somewhat more, ... how distant they are from each other." Participant 2 reported in their interview that their process of transcribing music improved their ability to read music, especially in treble clef and bass clef, and that their piano skills improved (through transcription). Participant 2 chose to transcribe Western staff sheet music into MIDI notation to recreate a song in Sonic Pi. As a result of this process, the participant exclaimed, "I've actually got better reading treble class and bass clef!" By engaging in Sonic Pi syntax, participants 1 and 2 were able to relate musical ideas in Sonic Pi with musical ideas they learned prior to this study.

With Sonic Pi, a user can create music resembling what one may learn in a typical music classroom in the United States. Participant 3 reported in their interview that while Sonic Pi is useful for creating electronic music, it can be useful for replicating acoustic music too, stating, “there's a difference between traditional music and electronic music, but even with Sonic Pi, there are notes and sounds from traditional instruments. So you could [write code] and make it sound like somebody is physically playing a traditional instrument.” Participant 3 highlighted that Sonic Pi has capability as both a compositional medium and performance instrument.

Western Staff Notation Isn't Required

Western staff notation is the primary medium for music notation in most music classrooms in the United States. Participants 1 and 2 reported on their entrance questionnaire that they are currently studying a musical instrument in school and can read Western staff notation. All participants reported on their exit surveys that they either disagree ($n = 2$) or strongly disagree ($n = 1$) that you have to read Western staff notation to create music, which is a change from their entrance questionnaire, in which participants 2 and 3 reported that they somewhat agree that you have to read Western staff notation to create music. The findings suggest that participants’ experience coding music in Sonic Pi may have changed their perspectives on whether someone needs to read Western staff notation to create music.

Participant 3, who played an instrument prior to this study but does not read Western staff notation, was elated by the accessibility to create music with code. Participant 3 stated, “That's like the joy of it, it's like you're able to make music, and it completely busts open that whole stereotype that you have to know an instrument or you have to read music in order to do musical things.” They continued, “I just thought I couldn't make music at all, or like you know, like with an instrument, because that's how I mainly thought music was made.” The ability to read Western staff notation was no longer perceived as a barrier for music making when coding music with Sonic Pi.

Research Question 3: Summary

The participants agreed that Sonic Pi is a tool to learn musical skills, concepts, and understandings. Its benefit to the individual participants varied; however, every participant acknowledged having learned or improved their understanding of, and ability to make, music. The perceptions of the participants regarding Western staff notation changed during the course of the study. Prior to working in Sonic Pi, some participants agreed that knowledge of Western staff notation was needed to make music. At the conclusion of the study, all of the participants agreed that Western staff notation was not necessary to make music. Each participant’s perception of their ability to learn music making was enhanced with the availability of Sonic Pi as an additional tool for learning.

Research Question 4: How does making music with code impact participants' interest in music courses?

Williams reported in 2011 that music courses in high schools across the United States were suffering from decreased enrollment.[395] Coding can be a novel way to pursue musical interests without the need for reading Western staff notation or an instrument other than a computing device. Participants were asked if their experience with Sonic Pi impacted their interests in further musical studies. All participants reported in their exit surveys that they were somewhat interested in continuing to make music. However, participants' interests in continuing musical studies in a formal setting were also impacted by personal interests and post-secondary plans.

Interest in Continuing Music Making, but not Necessarily with Code

Participant 1 reported on their entrance questionnaire that, prior to this experience, they made music using GarageBand. Although they reported enjoying creating music with code, this participant also noted in their exit interview that there are simpler ways to make music than to write it with code. They stated, "I believe that making the music specifically with code was interesting, and was

---

395. David A. Williams, "The Elephant in the Room," *Music Educators Journal* 98, no. 1 (2011): 51, https://doi.org/10.1177/0027432111415538.

somewhat fun to like dive into and try to like explore, ... I feel like I don't need such intricacy and that, like I can find more simple ways to do it." The granular level of detail that code permits may have inhibited the creative process for Participant 1.

Similarly, participant 2 enjoyed using Sonic Pi to realize transcriptions of video game music; however, they noted that "I really enjoyed having this experience. I mean, I'm not gonna actually use it to make music, but like it's just something to do for fun." Participant 2 also expressed in their interview that Sonic Pi was a "fun tool," and making music for them was something that was "on and off" as a hobby that for them was done prior in the DAWs GarageBand and Reaper.

However, participant 3 viewed music making with Sonic Pi as something to pursue during their free time and perhaps during college. Participant 3 stated, "yeah definitely I feel like this is something I would probably do in my free like off time, and just make little sounds for fun, or like, maybe in the future in college? Who knows I'll have a project? and it's like a video...that needs background sound. Oh, I know how to code that and I can just code a little music, you know that's awesome." Participant 3 viewed music making with code as a form of casual or project-based leisure and an opportunity to enhance their productivity in other areas by making cross content connections.[396]

For participants 1 and 2, prior experiences in music, whether with reading Western staff notation or with digital audio workstations, appeared to impact their view of music making prior to the study. For these participants, they had experience making music in other ways that they perceived as simpler or easier. For participant 3 however, music making was a barrier until they were introduced to Sonic Pi. Their experience with Sonic Pi removed the music making barrier so now they can envision themselves making music in creative ways.

All three participants expressed their post-secondary plans; none of which included formal studies in music. Participants 1 and 2 expressed in their interviews that they will pursue majors in computer science in college. Participant 3 expressed in their interview that they will pursue a major in biomedical engineering in college. All three participants recognized that their future majors included required course requirements that would likely restrict options for formal continued musical studies.

---

396. Robert A Stebbins, "The Serious Leisure Perspective," in *The Idea of Leisure: First Principles* (Taylor & Francis Group, 2012), 67.

Research Question 4: Summary

Participants' interests in future music courses were varied. Each participant drew on prior experiences throughout the study and related their coding experiences during the study to their prior experiences. Prior knowledge of DAWs negatively impacted some participants' perspectives of making music with Sonic Pi. For participant 3, however, Sonic Pi enabled them to make music, which was previously a barrier. While each participant found the experience of coding music interesting, continued studies in music were impacted additionally by participants' career interests and post-secondary plans.

Research Question 5: How does making music with code impact participants' interest in computer science courses?

All participants reported on their entrance questionnaire that they had experience coding prior to the study. Additionally, all participants reported on their exit surveys that they were somewhat interested, interested, or very interested in continuing coding, however not necessarily related to coding music. Participants identified that coding was important for their future careers and that Sonic Pi is helpful for individuals who want to learn coding.

Coding is Important for Career

Coding is a valuable skill that is useful in a number of careers, but more importantly, can foster creativity. Participants 2 and 3 expressed in their interviews that coding was a valuable skill and that growing one's coding skills is beneficial for a variety of careers. Participant 2 stated, "I recommend [musical coding] if you're trying to be like a music major, or if you're like a programming major or like a computer science major. I feel like this is a fun experiment for you to basically test your programming skills as well as develop your music skills." For participant 2, there was a direct benefit to musical coding, that it is mutually beneficial for musicians and programmers alike. Participant 3 reported in their interview that code is important for their future career, that coding is useful, and that coding is a skill everyone should know. They stated, "you can combine your love from that few, and use coding to aid it, and make it even better right, which I think is like something you should consider." Participant 3 believed that coding can enhance anyone's passions.

Sonic Pi is Helpful for Learning Programming

Participants expressed that Sonic Pi is beneficial for learning coding. Participant 1 stated, "I believe that this is kind of a merger between two things that the interest for has grown, and I feel like the concept could be like very appealing to many people." Sonic Pi merges music and coding by using a "musical narrative" to attract people to coding.[397] Participant 2 reported in their interview

397. Aaron, "Live Coding Education," 45.

that Sonic Pi helped them improve their programming skills and because of this opportunity, they better understand Ruby, the coding language used in Sonic Pi. They stated, "I've gotten better at using [Sonic Pi], because I wasn't familiar with it before and now I am because I looked more into [Sonic Pi and Ruby] documentation." Participant 2 also stated, "if you recognize the overall concepts of programming well, at least in high level programming, ... you're able to pretty much program anything in any language." Participant 2 stated that programming concepts (e.g., functions, variables, iteration, and others) work in similar ways across different programming languages. Whether someone codes music, an app, a game, etc., the concepts used in one programming language are transferable to other programming languages. Participant 3 stated, "I think this will really make people want to do coding more because coding is fun. But, I just don't think that many people realize it, especially like you know, people who are like minorities, because there's not a lot of representation [in] the computer science field you know, for women, people of color and stuff like that. People shy away from the lack of representation, as well as just people thinking it's hard and boring. But, something like Sonic Pi is like really fun." To participant 3, Sonic Pi is both fun and useful for engaging populations that are underrepresented in computer science.

Research Question 5: Summary

For all the participants, coding is a valuable skill that will benefit each of them in their future endeavors. Additionally, coding is a valuable skill for any individual to learn. To participants, Sonic Pi and musical coding is beneficial for growing coding skills. Sonic Pi is an engaging and fun platform for learning coding and is recommended by participants to learn coding.

Summary

Participants' code and multimedia artifacts revealed a close alignment to the "four properties of sound:" pitch, duration, intensity/amplitude, and timbre.[398] Although the four properties of sound were useful for organizing some of the findings, participants' artifacts revealed findings and demonstrated ideas extending beyond the four properties and standard conventions of Western staff notation, including: form, non-traditional music notation, and randomization. Participants' perceptions of their music making, their ability to learn music making, interests in music courses, and interests in computer science courses revealed several themes. All three of the participants agreed their artifacts are music. Additionally, participants' varied responses about musicianship and composers suggests that making music is something anyone can engage in,

398. Benward and Saker, *Music in Theory and Practice,* xiv-xvi.

regardless of how one identifies themself. All participants agree that Sonic Pi is a tool for learning and understanding musical concepts and that Western staff notation is not required knowledge for making music. Finally, participants' interests in music or computer science courses were impacted by their prior experiences in music and/or coding, and plans for the future.

CHAPTER 6: DISCUSSION AND IMPLICATIONS

Robert Moog and Herbert Deutsch were among the first to teach young learners to make music using Moog synthesizers. Their attempts reached only a few students who were fortunate enough to study in "electronic music labs" for an elective course.[399] They were optimistic that computers and MIDI technology would facilitate wider access to music making in the future.[400] Papert explored music making with code in 1971 with Logo, before personal computers were commercially available.[401] Puckette affirmed this idea, stating that computers are affordable and computer-based compositions are "easy to document and re-create."[402] Computers have expanded our palette of compositional and performance tools, pushing the boundaries of the four properties of sound and the elements of music to include new techniques like non-traditional notation and randomization. However, this vision of making music with computers in classrooms is still a relatively small niche in music classes across the United

399. Deutsch, "Where was Technology and Music Education Twenty Years Ago?" 94.

400. Deutsch, "Where was Technology and Music Education Twenty Years Ago?" 94.

401. Papert and Solomon, "Twenty Things to Do with a Computer."

402. Puckette, *The Theory and Technique of Electronic Music*, xi.

States, and is positioned to compete with ensemble-based music programs.[403] Within this niche, learners are typically limited to making music with Digital Audio Workstations (DAWs) in what has become known as the "GarageBand phenomenon."[404] The findings from this study raise questions around learner engagement using music technology and code for the fields of music education and computer science education.

Musicianship, Identity, and Current U. S. Music Standards

Musicians demonstrate a variety of skills in their making of music. Using a computer as a computational tool to concretize music through Sonic Pi expanded the participants' ideas of musicianship to include coding as a musical skill. Despite their varied understandings of Western staff notation and their varied perceptions of musical identity, participants demonstrated and reported their understanding of musical ideas with code.

Sonic Pi Improved Participants' Understanding of Music

Participants' growth in understanding might call into question how music educators understand music; both culturally and educationally. Is there an

---

403. Williams, "The Elephant in the Room," 52.

404. Williams, "Reaching the 'Other 80%:' Using Technology to Engage 'Non-Traditional Music Students' in Creative Activities."

inclusive definition of what music is (or how it's made)? Music is defined in a number of ways in a variety of contexts; however, I agree with other scholars such as Kratus and Williams who posit that music education in the United States is anachronistic with the ways people make music outside of school contexts and definitions of music making.[405]

The four properties of sound (pitch, duration, amplitude, and timbre) is one framework for interpreting how one hears things. However, participants' understanding of music extended beyond these properties to concretely demonstrate understandings of non-traditional notation and randomization using a computer and Sonic Pi.[406] Participants' music can also be discussed within the context of the "elements of music." However, many scholars disagree on what the elements of music include with the exception that non-traditional notation and randomization are usually not included.

Although there are some commonalities, the four properties of sound and the elements of music are not interchangeable. For our discussion, I will reference

---

405. Kratus, "A Return to Amateurism in Music Education," 32; Williams, "Reaching the 'Other 80%:' Using Technology to Engage 'Non-Traditional Music Students' in Creative Activities"; Williams, "The Elephant in the Room," 53; John Kratus, "Music Education at the Tipping Point," *Music Educators Journal* 94, no. 2 (November 1, 2007): 44–45, https://doi.org/10.1177/002743210709400209.

406. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 461.

the Elements of Music from the book, *Listening to the World*, by Antoni Pizà.

Pizà identifies the following elements of music:

- Pitch: the highness or lowness of a sound.
    - Melody: the sequence of pitches that make up a musical line.
    - Harmony: the combination of different pitches played or sung at the same time to create a chord.
- Beat: a unit of time, the underlying pulse.
    - Rhythm: the combination of long and short beats.
    - Tempo: the speed of the music, measured in beats per minute (BPM).
    - Meter: the way in which beats are organized and grouped in a measure.
- Timbre: the unique quality or tone color of a sound; the instruments.
- Texture: the overall density and complexity of the music, whether it is thick or thin.
- Form: the structure of the music, such as the arrangement of sections and the way they relate to each other.
- Dynamics: the loudness or softness of a sound.
- Genre: a general category that refers to the purpose of music.
- Context, Society, Politics: The milieu in which the music develops, including teaching, notation, transmission, scholarship, censorship, freedom of speech, gender and sexuality, colonialism, diaspora, etc.[407]

Pizà's elements expand upon the National Core Arts Standards (NCCAS) elements, which include: pitch, rhythm, harmony, dynamics, timbre, texture, form, and style/articulation.[408] Neither Pizà's or the NCCAS elements of music

---

407. Antoni Pizà, *Listening to The World: A Brief Survey of World Music* (New York: The City University of New York), xiv–xv, accessed March 10, 2023, https://pressbooks.cuny.edu/apiza/.

408. "GLOSSARY for National Core Arts: Music STANDARDS" (National Coalition for Core Arts Standards, n.d.),

address notation or randomization. This makes sense, given that in some contexts, music does not require notation or randomization. Randomization, for example, is specific to aleatoric music and computational systems, including electronic music. It is appropriate to think of the elements of music as an extension of the properties of sound because the elements of music suggest systems to organize sound. Findings from this study suggest a possible extension of the elements of music to include preservation and systems of musical transfer (including multiple systems of notation), and electronic systems (including computational tools and emerging technologies like artificial intelligence).

Musical Identity

Although participants engaged with and created music, their views on being a musician and composer varied. By virtue of making music, participants 1 and 2 somewhat identified with being a musician and only participant 2 somewhat identified with being a composer. Isbell stated that strong musician identities "reflect the influence of significant people and events from [one's] youth."[409] This may suggest that the participants in this study had limited

---

https://www.nationalartsstandards.org/sites/default/files/NCCAS%20GLOSSARY%20for%20Music%20Standards%201%20column.pdf.

409. Daniel S. Isbell, "Musicians and Teachers: The Socialization and Occupational Identity of Preservice Music Teachers," *Journal of Research in*

influences from musicians, composers, or limited events contributing to their musical identity.

School-based music programs may be an ideal place to facilitate the development of musical identities due to influential people, like music educators, and their ability to organize events for young musicians. In order to accomplish this, schools may need to address access, participation, and equity in their programs.[410] Improved course offerings, specifically those with a low floor, may offer opportunities to students who do not have knowledge of Western staff notation.[411] Additionally, offering courses with a low floor may improve opportunities for learners of lower socioeconomic status.[412]

Music educators play an important role in fostering identity development in youth, to support both an understanding on how to make music and building

---

Music Education 56, no. 2 (2008): 162, https://doi.org/10.1177/0022429408322853.

410. Kenneth Elpus and Carlos R. Abril, "Who Enrolls in High School Music? A National Profile of U.S. Students, 2009–2013," *Journal of Research in Music Education* 67, no. 3 (2019): 324, https://doi.org/10.1177/0022429419862837.

411. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 404.

412. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 404; Elpus and Abril, "Who Enrolls in High School Music? A National Profile of U.S. Students, 2009–2013," 324.

an identity as musicians.[413] Music educators' perceptions of musicianship can broadly influence the development of youth musicians' identities. Music educators might ask themselves and their students what is a musician? What are the roles of musicians? When is someone a musician? What spaces (physical and virtual) do musicians occupy? Music educators might also consider whether being a musician is occupational or does it include leisure activities? Music educators could also consider the pathways one may pursue in their journey as a musician. When asked about continuing studies in music, none of the participants stated they would continue formal studies in music. However, all of the participants stated that they would continue making music, including as a "hobby." Music educators who emphasize amateur and hobbyist endeavors in music as valid pathways may impact learners' identities as musicians.

Historic marginalization has been an issue in both music and computer science education. Learners of low socioeconomic status and English language learners are underrepresented in music classrooms.[414] Computer science as a

413. Giulia Ripani, "Children's Representations of Music, Musical Identities, and Musical Engagement: Content and Socio-Demographic Influences," *Journal of Research in Music Education* 70, no. 3 (2021): 289.

414. Elpus and Abril, "Who Enrolls in High School Music? A National Profile of U.S. Students, 2009–2013," 325.

discipline has "historically ... marginalized women and students of color."[415] Shaw and Kafai state, "As learners come to identify with activities, they begin to see themselves as participants, developing a sense of belonging to a larger community."[416] Music education and computer science education "[possess] historically and culturally legitimized practices, norms, and recognized types of participants that have marginalized many groups."[417] This may have contributed to the participants' perceptions in this study of not identifying as musicians. Western practices, norms, and participating persons in music traditionally do not include computer musicians.

The modules within this study were self-guided and although it was facilitated by myself, prioritizing identity development as a musician during the study was not a focus. Participants' answers to the exit survey regarding musician and composer identities were mostly unchanged from their answers to

---

415. Mia Shaw and Kafai, Yasmin, "Charting the Identity Turn in K-12 Computer Science Education: Developing More Inclusive Learning Pathways for Identities" (ICLS 2020, International Society of the Learning Sciences (ISLS), 2020), 114, https://doi.dx.org/10.22318/icls2020.114.

416. Shaw and Kafai, Yasmin, "Charting the Identity Turn in K-12 Computer Science Education: Developing More Inclusive Learning Pathways for Identities," 114.

417. Shaw and Kafai, Yasmin, "Charting the Identity Turn in K-12 Computer Science Education: Developing More Inclusive Learning Pathways for Identities," 114.

the entrance questionnaire, suggesting that the study was not a significant enough event to influence personal perceptions of their identities. This may further support the idea that an educator's role is significant in identity development. If an educator does not engage in discourse around identity development, opportunities for identity development and pathways in music may be missed. Many of the participants related musical identity to level of commitment and that a "serious" commitment to music was necessary to be a musician. The music educator is key in conveying that one can be a musician, even if their commitment to music is for leisure.

Musical identity may also be fostered by the physical or virtual spaces in which one makes music. Sonic Pi may be utilized in many spaces. Sonic Pi was developed with robust performance features in mind, specifically the `live_loop` function, which is a continuous loop of the user's music. However, Sonic Pi's `live_loop` integration makes it equally capable as a learning and compositional tool. With the integration of `live_loop`, Sonic Pi "provides many new learning opportunities in the classroom due to its immersive nature and incredibly fast feedback cycle."[418] Much of this is influenced by the space a Sonic Pi user intends

418. Aaron, "Live Coding Education," 46.

to create.[419] The space, whether it is a venue, classroom, or a comfortable environment for creativity, is important for fostering identity. However, participants did not explore this capability with Sonic Pi, including the spaces in which Sonic Pi is performed. Future studies could explore identity development as a performer through live coding of music in platforms such as Sonic Pi.

Western Staff Notation was a Perceived Barrier, But Not with Sonic Pi

Sonic Pi makes use of high level, readable language to communicate musical ideas. Users communicate musical ideas to Sonic Pi by using descriptive language.[420] Papert relates this idea to teaching a physical activity (e.g., riding a bike). By using descriptive language, and structuring the language into manageable subroutines or “chunks,” participants were able to take their theoretical ideas about music and concretize them with Sonic Pi in concrete ways.[421] This approach to making music aligns with constructionist practices described by Papert and Resnick when learners constructed knowledge using

---

419. Robert A Stebbins, “Leisure Music Production Its Spaces and Places,” in *The Oxford Handbook of Music Making and Leisure*, ed. Roger Mantie and Gareth Dylan Smith (Oxford University Press, Incorporated, 2017), 349, http://ebookcentral.proquest.com/lib/bu/detail.action?docID=4745366.

420. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 1430.

421. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 1508.

descriptive language for music.[422]

To notate pitch and duration, Sonic Pi uses MIDI note names, MIDI numbers, and numerical durations. MIDI note names in Sonic Pi are objects that are part of the Sonic Pi language. MIDI numbers, on the other hand, are standardized as part of the Standard MIDI Files (SMF) Specification.[423] The incorporation of both MIDI note names and numbers makes pitch more accessible to users with varied musical backgrounds. Users with an understanding of note names, perhaps from experience with a musical instrument or Western staff notation, can relate their understanding of MIDI note names in Sonic Pi. Similarly, users with an understanding of MIDI numbers, maybe from experience with a DAW, can relate their understanding of MIDI numbers in Sonic Pi. Numerical durations are what makes Sonic Pi strongly-timed, meaning that they are precise and may be synchronized for time-based concurrent programming, an essential piece of music and audio programming that affords the ability to create harmonies, complex orchestrations, and more through the ability to

422. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 1454; Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 1163.

423. "Standard MIDI Files (SMF) Specification," accessed March 16, 2023, https://www.midi.org/specifications-old/item/standard-midi-files-smf.

simultaneously layer multiple voices.[424]

Making music with descriptive language in code may be a novel type of music notation. By learning to structure programs into smaller, “natural parts,” users can identify problems quickly in their music and fix them with descriptive language with which they are already familiar.[425] This, in turn, may make the music making process a more enjoyable experience for users. Users may also share the same sentiment as participant 3, who drew the conclusion that “it completely busts open that whole stereotype that you have to know an instrument or you have to read music in order to do musical things.”

Western staff notation may be a barrier for some learners to create music, however learners may create music with a variety of other non-computer-based types of notation. Songwriting, for example, sometimes focuses on lyrics or chord progressions, or a combination of the two. The songwriter may write lyrics and indicate chord changes above the lyrics using jazz chord symbols. Guitarists and other fretted instrument players may write down their ideas using tablature to

---

424. “Sam Aaron’s Sonic Pi Can Help You Make Music Through Code”; Wang, “The ChucK Audio Programming Language. ‘A Strongly-Timed and On-the-Fly Environ/Mentality,’” 5; Wang, Cook, and Salazar, “ChucK: A Strongly Timed Computer Music Language,” 10.

425. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 1454, 1611.

indicate the fret each note is played on. Frequently, creating music in this way happens in informal spaces outside of schools. The songwriter, in collaboration with musicians, may use descriptive language to communicate and clarify ideas in their songwriting. This example, along with music programming languages and other means of creating music are opportunities for multiple pathways for making music in classrooms.[426]

Several ethnographers have noted the use of alternative methods of music notation in their field work. Several contexts exist, including cultural contexts, that utilize oral (descriptive language usually incorporating onomatopoetic syllables) or tactile methods of preserving and transferring knowledge of music. Examples include: First Nations' music, New Guinean music, Ewe music of West Africa, Balinese music, Venda music of North Transvaal, Chopi music of East Africa, and more.[427] Ethnographers often use iconic notation, tablature, or a variation of Western staff notation enhanced with icons (which are frequently accompanied by a notation key), or a combination of the above to preserve their observations in writing. Friedson, in his observations of the Tumbuka people

---

426. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 914.

427. Alan P. Merriam, *The Anthropology of Music* (Evanston, Illinois: Northwestern University Press, 1964), 145–50; J. H. Kwabena Nketia, *The Music of Africa* (New York, NY: W. W. Norton & Company, Inc., 1974), 61–63.

utilized a rhythmic tablature in which letters represented right or left hands and the instrument tones they produced.[428] Kisliuk, in her observations of the BaAka people combined icons and tablature, in addition to modified Western staff notation as it met her needs for transcription.[429] Seeger combined icons and Western staff notation with a detailed key to transcribe the singing of the Suyá.[430] Seeger's icons were an effort to convey the tonality of the singing since the Suyá do not use Western scales. Western staff notation is limited, as are other systems of notation. However, granting flexibility in how we create music opens multiple pathways for students to engage in music making.[431]

Randomization and granular control of sound synthesis are beyond the standard conventions of Western staff notation and are usually omitted from educational materials focused on the elements of music. This is likely because

---

428. Steven M. Friedson, *Dancing Prophets: Musical Experience in Tumbuka Healing* (Chicago, Illinois: The University of Chicago Press, 1996), 137–54.

429. Michelle Kisliuk, *Seize the Dance: BaAka Musical Life and the Ethnography of Performance* (New York, NY: Oxford University Press, 1998), 102–3, 113, 117, 136–38.

430. Anthony Seeger, *Why Suyá Sing: A Musical Anthropology of an Amazonian People* (Chicago, Illinois: University of Illinois Press, 2004), 88–96.

431. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 914.

these concepts are related to electronic music and the elements of music, including those outlined by Pizà, are deeply rooted in acoustic music traditions. However, the findings from this study raise questions about when and how music educators might notate these concepts.

The best example of randomization written in Western staff notation is aleatoric music. In the 18th century aleatoric music often consisted of a collection of melodies or progressions that a performer assembled by physically rolling dice to determine the unique sequence of noted segments[432] A similar act can be replicated in Sonic Pi using the `dice` function.

One might argue that randomization is notated in Western staff notation, specifically in jazz music to indicate when it is appropriate to improvise. However, improvisation is not entirely random. Jazz musicians deduce the appropriate scale(s)/mode(s) to improvise from based on chord symbols in the notation, like in figure 26 below.

432. Philip Ball, "Can We Use Quantum Computers to Make Music?," Physics World, February 28, 2023, https://physicsworld.com/can-we-use-quantum-computers-to-make-music/.

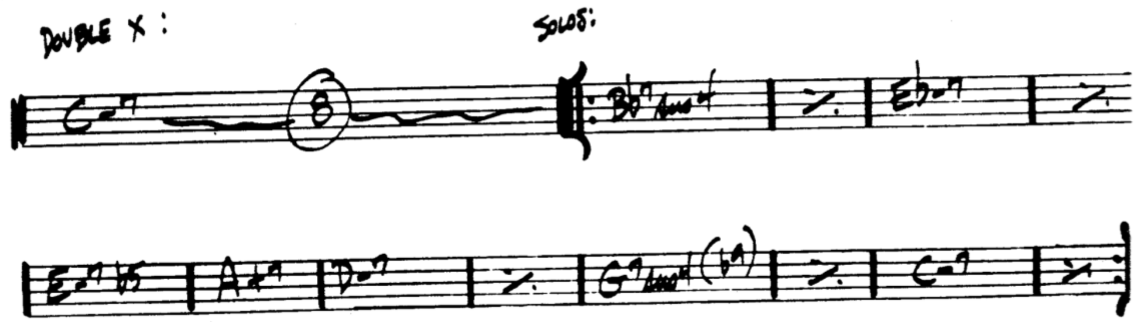


Figure 26. Solo Section, “Crescent” by John Coltrane, *The Real Book Volume I* (5th ed.), 99

The scale(s)/mode(s) then gives the performer a selection of notes to choose from so the improvisation is within the correct key. This can be replicated in Sonic Pi using the `choose()` function. As performers develop their improvisation skills, they also develop a repertoire of “licks” or short melodies or rhythms that they can recall when the lick is appropriate for the chord changes. This can also be replicated in Sonic Pi, by storing melodies or rhythms into variables and arrays.

PreK-12 National Music Standards

Code-based music and alternative forms of music notation provide opportunities for music educators to facilitate music creation with learners. However, current music standards in the United States may present hurdles to code-based music making. The National Core Arts Standards (NCCAS) for music and the National Association for Music Education (NAfME) music standards, may unintentionally limit the learning of musical ideas to those that fit within the framework of the four properties of sound, or elements of music. These two sets of standards are nearly the same, however there are some subtle differences. Within the sets of both standards are strands, or domain areas that are meant to align with the core musical activities (composing, performing, and audience-listening), including: creating, performing, responding, and connecting.[433] I will address the "Creating" sections within three areas of national music standards below based on relevance, including: PK-8 General Music, Composition/Theory, and Music Technology.

PK-8 general music. Notation in the PK-8 General Music standards address notation as either "iconic notation" or "standard notation."[434] Sequentially, in the NCCAS for Music, PreK-8 in figure 27, iconic notation precedes standard notation, suggesting that standard notation is the objective goal. This framing asserts the importance of standard notation in standards and curricula. This prioritization continues into formal studies of music at the post-secondary level. In fact, standard notation is the first topic of chapter one in

433. Tim Cain, "Theory, Technology and the Music Curriculum," 217.

434. National Core Arts Standards: PreK-8 General Music (2014), https://www.nationalartsstandards.org/sites/default/files/2021-11/Music%20at%20a%20Glance%20rev%2012-1-16.pdf; National Association for Music Education: PK-8 General Music Standards (2014), https://nafme.org/wp-content/uploads/2014/11/2014-Music-Standards-PK-8-Strand.pdf.

Music in Theory and Practice, a common textbook for music theory, following the introduction which includes the four properties of sound.[435]

Anchor Standard 2: Organize and develop artistic ideas and work.
Enduring Understanding: Musicians' creative choices are influenced by their expertise, context, and expressive intent.
Essential Question(s): How do musicians make creative decisions?

CREATING	Pre K (MU:Cr2.1.PK)	Kindergarten (MU:Cr2.1.K)	1st (MU:Cr2.1.1)	2nd (MU:Cr2.1.2)	3rd (MU:Cr2.1.3)	4th (MU:Cr2.1.4)	5th (MU:Cr2.1.5)	6th (MU:Cr2.1.6)	7th (MU:Cr2.1.7)	8th (MU:Cr2.1.8)
Plan and Make	a With substantial guidance, explore favorite musical ideas (such as movements, vocalizations, or instrumental accompaniments).	a With guidance, demonstrate and choose favorite musical ideas.	a With limited guidance, demonstrate and discuss personal reasons for selecting musical ideas that represent expressive intent.	a Demonstrate and explain personal reasons for selecting patterns and ideas for music that represent expressive intent.	a Demonstrate selected musical ideas for a simple improvisation or composition to express intent, and describe connection to a specific purpose and context.	a Demonstrate selected and organized musical ideas for an improvisation, arrangement, or composition to express intent, and explain connection to purpose and context.	a Demonstrate selected and develop ed musical ideas for improvisations, arrangements, or compositions to express intent, and explain connection to purpose and context.	a Select, organize, construct, and document personal musical ideas for arrangements and compositions within AB or ABA form that demonstrate an effective beginning, middle, and ending, and convey expressive intent.	a Select, organize, develop and document personal musical ideas for arrangements, songs, and compositions within AB, ABA, or theme and variation forms that demonstrate unity and variety and convey expressive intent.	a Select, organize, and document personal musical ideas for arrangements, songs, and compositions within expanded forms that demonstrate tension and release, unity and variety, balance, and convey expressive intent.
	b – With substantial guidance, select and keep track of the order for performing original musical ideas, using iconic notation and/or recording technology.	a With guidance, organize personal musical ideas using iconic notation and/or recording technology.	b With limited guidance, use iconic or standard notation and/or recording technology to document and organize personal musical ideas.	b Use iconic or standard notation and/or recording technology to combine, sequence, and document personal musical ideas.	b Use standard and/or iconic notation and/or recording technology to document personal rhythmic and melodic musical ideas.	b Use standard and/or iconic notation and/or recording technology to document personal rhythmic, melodic, and simple harmonic musical ideas.	b Use standard and/or iconic notation and/or recording technology to document personal rhythmic, melodic, and two-chord harmonic musical ideas.	b Use standard and/or iconic notation and/or audio/ video recording to document personal simple rhythmic phrases, melodic phrases, and two-chord harmonic musical ideas.	b Use standard and/or iconic notation and/or audio/ video recording to document personal simple rhythmic phrases, melodic phrases, and harmonic sequences.	b Use standard and/or iconic notation and/or audio/ video recording to document personal rhythmic phrases, melodic phrases, and harmonic sequences.

Figure 27. Standards for Creating - "Plan and Make", Music - PreK-8 Strand, National Core Arts Standards (https://www.nationalartsstandards.org/)

The NCCAS for Music, PreK-8 and the NAfME PK-8 General Music standards are the same in both wording and emphasis. Key terms are in red and bold in both documents. There is a definite emphasis on traditional Western music theory concepts. By grade 8, learners should have a grasp on several Western music theory concepts, including: melody, rhythm, harmony, binary and ternary forms (including introductions, transitions, and codas), and Western staff notation.[436] Some of these concepts exist outside of the Western tradition like

435. Benward and Saker, *Music in Theory and Practice*, 3.

436. "National Core Arts Standards: PreK-8 General Music," 1–3; "National Association for Music Education: PK-8 General Music Standards," 1–4.

melody and rhythm. To frame music education in only a classical Western context excludes non-Western traditions and novel techniques including music making with code.

Composition/theory. Subtle differences appear between the NCCAS for music and the NAfME music standards, most notably the omission of standards below “Proficient.” In figure 28, the standards for Creating are leveled by “HS” (high school) “Proficient,” “Accomplished,” and “Advanced.” The omission of levels below proficient suggests an assumption that students pursuing music in high school have studied music in lower grade levels, which ideally would have prepared them to create at a proficient level in high school. However, students with gaps in their musical studies may have difficulties meeting this expectation. Additionally, students who have little to no prior experience in a music classroom in the United States may have difficulties meeting these expectations. These standards may not be appropriate for those individuals.

Music - Composition and Theory Strand

	HS Proficient	HS Accomplished	HS Advanced	
CREATING	Anchor Standard 1: Generate and conceptualize artistic ideas and work. Enduring Understanding: The creative ideas, concepts, and feelings that influence musicians' work emerge from a variety of sources. Essential Question(s): How do musicians generate creative ideas?			
Imagine	MU:Cr1.1.C.Ia Describe how sounds and short musical ideas can be used to represent personal experiences, moods, visual images, and/or storylines.	MU:Cr1.1.C.IIa Describe *and demonstrate* how sounds and musical ideas can be used to represent *sonic events*, *memories*, visual images, *concepts, texts*, or storylines.	MU:Cr1.1.C.IIIa Describe and demonstrate *multiple ways in which* sounds and musical ideas can be used to represent *extended sonic experiences or abstract ideas*.	Imagine
CREATING	Anchor Standard 2: Organize and develop artistic ideas and work. Enduring Understanding: Musicians' creative choices are influenced by their expertise, context, and expressive intent. Essential Question(s): How do musicians make creative decisions?			
Plan and Make	MU:Cr2.1.C.Ia Assemble and organize sounds or short musical ideas to create initial expressions of selected experiences, moods, images, or storylines.	MU:Cr2.1.C.IIa Assemble and organize multiple sounds or musical ideas to create initial expressive statements of selected *sonic events*, *memories,* images, *concepts, texts,* or storylines.	MU:Cr2.1.C.IIIa Assemble and organize multiple sounds or extended musical ideas to create initial expressive statements of selected *extended sonic experiences or abstract ideas*.	Plan and Make
Plan and Make	MU:Cr2.1.C.Ib Identify and describe the development of sounds or short musical ideas in drafts of music within simple forms (such as one-part, cyclical, or binary).	MU:Cr2.1.C.IIb Describe and explain the development of sounds and musical ideas in drafts of music within a variety of simple or *moderately complex forms (such as* binary, *rondo*, *or ternary)*.	MU:Cr2.1.C.IIIb Analyze and demonstrate the development of sounds and extended musical ideas in drafts of music within a *variety of* moderately complex *or complex forms*.	Plan and Make
CREATING	Anchor Standard 3: Refine and complete artistic work. Enduring Understanding: Musicians evaluate, and refine their work through openness to new ideas, persistence, and the application of appropriate criteria. Essential Question(s): How do musicians improve the quality of their creative work?			
Evaluate and Refine	MU:Cr3.1.C.Ia Identify, describe, and apply teacher-provided criteria to assess and refine the technical and expressive aspects of evolving drafts leading to final versions.	MU:Cr3.1.C.IIa Identify, describe, and apply *selected* teacher-provided *or personally-developed* criteria to assess and refine the technical and expressive aspects of evolving drafts leading to final versions.	MU:Cr3.1.C.IIIa *Research,* identify, *explain*, and apply personally-developed criteria to assess and refine the technical and expressive aspects of evolving drafts leading to final versions.	Evaluate and Refine
	Enduring Understanding: Musicians' presentation of creative work is the culmination of a process of creation and communication. Essential Question(s): When is creative work ready to share?			
Present	MU:Cr3.2.C.Ia Share music through the use of notation, performance, or technology, and demonstrate how the elements of music have been employed to realize expressive intent.	MU:Cr3.2.C.IIa Share music through the use of notation, *solo or group* performance, or technology, and demonstrate and *describe* how the elements of music and *compositional techniques* have been employed to realize expressive intent.	MU:Cr3.2.C.IIIa Share music through the use of notation, solo or group performance, or technology, and demonstrate *and explain* how the elements of music, compositional techniques *and processes* have been employed to realize expressive intent.	Present
Present	MU:Cr3.2.C.Ib Describe the given context and performance medium for presenting personal works, and how they impact the final composition and presentation.	MU:Cr3.2.C.IIb Describe the *selected contexts* and performance mediums for presenting personal works, and *explain why* they *successfully* impact the final composition and presentation.	MU:Cr3.2.C.IIIb Describe a *variety of possible contexts and mediums* for presenting personal works, and explain *and compare how each could impact the success of* the final composition and presentation.	Present

Figure 28. Standards for Creating, Music - Composition and Theory Strand, National Core Arts Standards (https://www.nationalartsstandards.org/)

The NAfME standards in figure 29, which are based on the NCCAS, include columns for “Novice” and “Intermediate;” however, they are intentionally left blank, assuming the same prior knowledge as the NCCAS. With digital technologies in music, musical creativity may not be limited to proficiency in Western music theory only, which demands training in an instrument or voice prior to musical creativity.[437]

2014 Music Standards (Composition/Theory)

CREATING

Imagine

Generate musical ideas for various purposes and contexts.

Enduring Understanding: The creative ideas, concepts, and feelings that influence musicians' work emerge from a variety of sources. **Essential Question:** How do musicians generate creative ideas?

	Novice	Intermediate	Proficient	Accomplished	Advanced
Common Anchor #1		.	**MU:Cr1.1.C.Ia** Describe how sounds and short musical ideas can be used to represent personal experiences, moods, visual images, and/or storylines.	**MU:Cr1.1.C.IIa** Describe *and demonstrate* how sounds and musical ideas can be used to represent *sonic events*, *memories*, visual images, *concepts*, *texts*, or storylines.	**MU:Cr1.1.C.IIIa** Describe and demonstrate *multiple ways in which* sounds and musical ideas can be used to represent *extended sonic experiences or abstract ideas.*

Plan and Make

Select and develop musical ideas for defined purposes and contexts.

Enduring Understanding: Musicians' creative choices are influenced by their expertise, context, and expressive intent. **Essential Question:** How do musicians make creative decisions?

	Novice	Intermediate	Proficient	Accomplished	Advanced
Common Anchor #2			**MU:Cr2.1.C.Ia** Assemble and organize sounds or short musical ideas to create initial expressions of selected experiences, moods, images, or storylines.	**MU:Cr2.1.C.IIa** Assemble and organize multiple sounds or musical ideas to create initial expressive statements of selected *sonic events*, *memories*, images, *concepts*, *texts*, or storylines.	**MU:Cr2.1.C.IIIa** Assemble and organize multiple sounds or extended musical ideas to create initial expressive statements of selected *extended sonic experiences or abstract ideas.*
			MU:Cr2.1.C.Ib Identify and describe the development of sounds or short musical ideas in drafts of music within simple forms (such as one-part, cyclical, or binary).	**MU:Cr2.1.C.IIb** Describe and explain the development of sounds and musical ideas in drafts of music within a variety of simple or *moderately complex forms (such as binary, rondo, or ternary).*	**MU:Cr2.1.C.IIIb** Analyze and demonstrate the development of sounds and extended musical ideas in drafts of music within a *variety of* moderately complex *or complex forms*.

Evaluate and Refine

Evaluate and refine selected musical ideas to create musical work that meets appropriate criteria.

Enduring Understanding: Musicians evaluate and refine their work through openness to new ideas, persistence, and the application of appropriate criteria. **Essential Question:** How do musicians improve the quality of their creative work?

	Novice	Intermediate	Proficient	Accomplished	Advanced
Common Anchor #3			**MU:Cr3.1.C.Ia** Identify, describe, and apply teacher-provided criteria to assess and refine the technical and expressive aspects of evolving drafts leading to final versions.	**MU:Cr3.1.C.IIa -** Identify, describe, and apply *selected* teacher-provided *or personally-developed* criteria to assess and refine the technical and expressive aspects of evolving drafts leading to final versions.	**MU:Cr3.1.C.IIIa -** *Research*, identify, *explain*, and apply personally-developed criteria to assess and refine the technical and expressive aspects of evolving drafts leading to final versions.

Page 1 of 9

2014 Music Standards (Composition/Theory)

Present

Share creative musical work that conveys intent, demonstrates craftsmanship, and exhibits originality.

Enduring Understanding: Musicians' presentation of creative work is the culmination of a process of creation and communication. **Essential Question:** When is creative work ready to share?

	Novice	Intermediate	Proficient	Accomplished	Advanced
			MU:Cr3.2.C.Ia Share music through the use of notation, performance, or technology, and demonstrate how the elements of music have been employed to realize expressive intent.	**MU:Cr3.2.C.IIa** Share music through the use of notation, *solo or group* performance, or technology, and demonstrate and *describe* how the elements of music and *compositional techniques* have been employed to realize expressive intent.	**MU:Cr3.2.C.IIIa** Share music through the use of notation, solo or group performance, or technology, and demonstrate *and explain* how the elements of music, compositional techniques *and processes* have been employed to realize expressive intent.
			MU:Cr3.2.C.Ib Describe the given context and performance medium for presenting personal works, and how they impact the final composition and presentation.	**MU:Cr3.2.C.IIb** Describe the *selected contexts* and performance mediums for presenting personal works, and *explain why* they *successfully* impact the final composition and presentation.	**MU:Cr3.2.C.IIIb** Describe a *variety of possible contexts and mediums* for presenting personal works, and explain *and compare how each could impact the success of* the final composition and presentation.

Figure 29. Standards for Creating, Composition/Theory, 2014 Music Standards, National Association for Music Education (https://nafme.org/wp-content/uploads/2014/11/2014-Music-Standards-Composition-Theory-Strand.pdf)

437. Stephen Travis Pope et al., "Touched by Machine?: Composition and Performance in the Digital Age," *Computer Music Journal* 19, no. 3 (1995): 14, https://doi.org/10.2307/3680650; Beckstead, "Will Technology Transform Music Education?," 45.

Making music with Sonic Pi or other music coding environments provide an entry point for novice or beginner composers, or for individuals without training in an instrument or voice, like participant 3 who exclaimed they "didn't need to know music" in order to create music with Sonic Pi.[438] However, students with gaps in their musical studies, or with little or no prior knowledge, may face a barrier in how they present their musical creations when abiding by the NCCAS standards. For example, in standard MU:Cr3.2.C.la from figures 28 and 29, students will "Share through the use of notation, performance, or technology and demonstrate how the elements of music have been employed to realize expressive intent." The national standards for music may not provide an entry point for novice or beginner composers. To achieve the prior-mentioned standard, the novice composer would be expected to construct knowledge reflecting 10-years of PreK-8 music standards that precede it. What is a reasonable amount of time for this expectation? Is this expectation equitable?

Music technology. In addition to the omissions of "Novice" and "Intermediate" standards, in figure 30, as with the Composition/Theory standards, the Music Technology standards are exceptionally vague.

---

438. Pope et al., "Touched by Machine?: Composition and Performance in the Digital Age," 14; "Sam Aaron's Sonic Pi Can Help You Make Music Through Code."

2014 Music Standards (Music Technology)

CREATING					
Imagine *Generate musical ideas for various purposes and contexts.*					
CA #1	**Enduring Understanding:** The creative ideas, concepts, and feelings that influence musicians' work emerge from a variety of sources. **Essential Question:** How do musicians generate creative ideas?				
	Novice	Intermediate	Proficient	Accomplished	Advanced
			MU:Cr1.1.T.Ia Generate melodic, rhythmic, and harmonic ideas for compositions or improvisations using digital tools.	**MU:Cr1.1.T.IIa** Generate melodic, rhythmic, and harmonic ideas for compositions *and* improvisations using digital tools and *resources*.	**MU:Cr1.1.T.IIIa** Generate melodic, rhythmic, and harmonic ideas for compositions and improvisations that incorporate digital tools, resources, *and systems*.
Plan and Make *Select and develop musical ideas for defined purposes and contexts.*					
CA #2	**Enduring Understanding:** Musicians' creative choices are influenced by their expertise, context, and expressive intent. **Essential Question:** How do musicians make creative decisions?				
	Novice	Intermediate	Proficient	Accomplished	Advanced
			MU:Cr2.1.T.Ia Select melodic, rhythmic, and harmonic ideas to develop into a larger work using digital tools and resources.	**MU:Cr2.1.T.IIa** Select melodic, rhythmic, and harmonic ideas to develop into a larger work *that exhibits unity and variety* using digital and *analog* tools.	**MU:Cr2.1.T.IIIa** Select, develop, *and organize* multiple melodic, rhythmic and harmonic ideas to develop into a larger work that exhibits unity, variety, *complexity, and coherence* using digital and analog tools, *resources, and systems*.
Evaluate and Refine *Evaluate and refine selected musical ideas to create musical work that meets appropriate criteria.*					
Common Anchor #3	**Enduring Understanding:** Musicians evaluate and refine their work through openness to new ideas, persistence, and the application of appropriate criteria. **Essential Question:** How do musicians improve the quality of their creative work?				
	Novice	Intermediate	Proficient	Accomplished	Advanced
			MU:Cr3.1.T.Ia Drawing on feedback from teachers and peers, develop and implement strategies to improve and refine the technical and expressive aspects of draft compositions and improvisations.	**MU:Cr3.1.T.IIa** *Develop and implement varied strategies to improve* and refine the technical and expressive aspects of draft compositions and improvisations.	**MU:Cr3.1.T.IIIa** Develop and implement varied strategies *and apply appropriate criteria* to improve and refine the technical and expressive aspects of draft compositions and improvisations.
	Present *Share creative musical work that conveys intent, demonstrates craftsmanship, and exhibits originality.*				
	Enduring Understanding: Musicians' presentation of creative work is the culmination of a process of creation and communication. **Essential Question:** When is creative work ready to share?				
	Novice	Intermediate	Proficient	Accomplished	Advanced
			MU:Cr3.2.T.Ia Share compositions or improvisations that demonstrate a proficient level of musical and technological craftsmanship as well as the use of digital tools and resources in developing and organizing musical ideas.	**MU:Cr3.2.T.IIa** Share compositions and improvisations that demonstrate an *accomplished* level of musical and technological craftsmanship *as well as the use of digital* and analog tools and resources in developing and organizing musical ideas.	**MU:Cr3.2.T.IIIa** *Share a portfolio of musical creations representing varied styles and genres that demonstrates an advanced* level of musical and technological craftsmanship as well as the use of digital and analog tools, resources *and systems* in developing and organizing musical ideas.

Page 1 of 6

Figure 30. Standards for Creating, Music Technology, 2014 Music Standards, National Association for Music Education (https://nafme.org/wp-content/uploads/2014/11/2014-Music-Standards-Technology-Strand.pdf)

One might explain this as a means to encompass varied music technologies. However, frequent reference to terms associated with the elements of music negate compositional tools beyond the elements of music (e.g., randomization and sampling). In figures 30 and 31, randomization and sampling, as well as other common elements of electronic music like control and automation, are not mentioned.[439] Participants in this study demonstrated and reported on techniques that exhibit knowledge of the elements of music, but also techniques unique to

439. Puckette, *The Theory and Technique of Electronic Music,* 61, 89.

electronic music, especially randomization and sampling, and granular control of parameters such as amplitude and effects.

Music - Music Technology Strand				
CREATING	Anchor Standard 1: Generate and conceptualize artistic ideas and work. Enduring Understanding: The creative ideas, concepts, and feelings that influence musicians' work emerge from a variety of sources. Essential Question(s): How do musicians generate creative ideas?			
	HS Proficient	HS Accomplished	HS Advanced	
Imagine	MU:Cr1.1.T.Ia Generate melodic, rhythmic, and harmonic ideas for compositions or improvisations using digital tools.	MU:Cr1.1.T.IIa Generate melodic, rhythmic, and harmonic ideas for compositions and improvisations using digital tools and resources.	MU:Cr1.1.T.IIIa Generate melodic, rhythmic, and harmonic ideas for compositions and improvisations that incorporate digital tools, resources, and systems.	Imagine
CREATING	Anchor Standard 2: Organize and develop artistic ideas and work. Enduring Understanding: Musicians' creative choices are influenced by their expertise, context, and expressive intent. Essential Question(s): How do musicians make creative decisions?			
	HS Proficient	HS Accomplished	HS Advanced	
Plan and Make	MU:Cr2.1.T.Ia Select melodic, rhythmic, and harmonic ideas to develop into a larger work using digital tools and resources.	MU:Cr2.1.T.IIa Select melodic, rhythmic, and harmonic ideas to develop into a larger work that exhibits unity and variety using digital and analog tools.	MU:Cr2.1.T.IIIa Select, develop, and organize multiple melodic, rhythmic and harmonic ideas to develop into a larger work that exhibits unity, variety, complexity, and coherence using digital and analog tools, resources, and systems.	Plan and Make
CREATING	Anchor Standard 3: Refine and complete artistic work. Enduring Understanding: Musicians evaluate, and refine their work through openness to new ideas, persistence, and the application of appropriate criteria. Essential Question(s): How do musicians improve the quality of their creative work?			
	HS Proficient	HS Accomplished	HS Advanced	
Evaluate and Refine	MU:Cr3.1.T.Ia Drawing on feedback from teachers and peers, develop and implement strategies to improve and refine the technical and expressive aspects of draft compositions and improvisations.	MU:Cr3.1.T.IIa Develop and implement varied strategies to improve and refine the technical and expressive aspects of draft compositions and improvisations.	MU:Cr3.1.T.IIIa Develop and implement varied strategies and apply appropriate criteria to improve and refine the technical and expressive aspects of draft compositions and improvisations.	Evaluate and Refine
	Enduring Understanding: Musicians' presentation of creative work is the culmination of a process of creation and communication Essential Question(s): When is creative work ready to share?			
	HS Proficient	HS Accomplished	HS Advanced	
Present	MU:Cr3.2.T.Ia Share compositions or improvisations that demonstrate a proficient level of musical and technological craftsmanship as well as the use of digital tools and resources in developing and organizing musical ideas.	MU:Cr3.2.T.IIa Share compositions and improvisations that demonstrate an accomplished level of musical and technological craftsmanship as well as the use of digital and analog tools and resources in developing and organizing musical ideas.	MU:Cr3.2.T.IIIa Share a portfolio of musical creations representing varied styles and genres that demonstrates an advanced level of musical and technological craftsmanship as well as the use of digital and analog tools, resources and systems in developing and organizing musical ideas.	Present

Figure 31. Standards for Creating, Music - Music Technology Strand, National Core Arts Standards (https://www.nationalartsstandards.org/)

Each set of standards above has four subsections within the "Creating" section, including: Imagine, Plan and Make, Evaluate and Refine, and Present. These subsections are strikingly similar to the design thinking process; however, the standards are unaligned with the subsections, or are unclear. For example, in both music technology figures 30 and 31, each level of proficiency for the subsection "Imagine" begins with "Generate melodic, rhythmic, or harmonic ideas for compositions or improvisations..." with differences in proficiency tied to the "digital tools," "resources," and "systems."[440] In my opinion, the differentiation of proficiency is unclear. One might assume "digital tools" to refer to the software platform being utilized (e.g., DAW, notation software, electronic instrument(s), programming environment, etc.). The problem here is that each "digital tool" requires different skills and procedures to generate a musical idea. In alignment with the current standards, one would remain "proficient" even if using multiple tools. "Resources" is another broad term, but may apply to libraries of files, perhaps samples. For each "digital tool" the use of samples requires different skills and procedures. "Systems" may intersect with what I have mentioned as digital tools. Rather than tie proficiency level of "Imagine" to unclear terms, music educators could tie proficiency levels to the skills and understandings of procedures utilized for generating musical ideas.

Since learning music with code pushes past the boundaries of the National Core Arts Standards for music and National Association for Music Education music standards, the Computer Science Teachers Association standards for

---

440. "National Association for Music Education: Music Standards (Music Technology)" (National Association for Music Education, 2014); "National Core Arts Standards: Music - Music Technology Strand" (National Core Arts Standards, 2014), www.nationalcoreartsstandards.org.

computer science may be used to address the deficits in music standards; such as algorithms, variables, lists/collections, iteration, and objects, among many others. If educators were to actively explore standards from two domains, such engagement aligns with O'Leary's discussion on music-centered making or create an entirely new interdiscipline of music and computer science.[441] Such an intersection may align with constructionist forms of music making that organically occurs in leisure outside of the classroom.

Summary

Participants reported that through Sonic Pi, they learned or improved musical skills and increased their understanding of music. However, participants' growth in understanding raises the question of how we understand music culturally and educationally. The elements of music extend the four properties of sound; however the elements of music represent traditional Western music theory in context and may unintentionally exclude creative techniques found in electronic music.

Music educators are important in fostering musical identities in youth. However, their perceptions of musicianship can positively or negatively influence

441. Jared O'Leary, "Intersections of Popular Musicianship and Computer Science Practices," 164.

the development of youth musicians' identities. It is important for music educators to relate musical identity with leisure activities in music as well as professional activities. Music educators can accomplish this by building positive relationships with their students and by providing various opportunities and spaces for learners to be musically creative.

Participants used descriptive language and subroutines to take their theoretical ideas about music and construct them with Sonic Pi. By using descriptive language in Sonic Pi and other music programming languages, making music is possible without knowledge of Western staff notation that may unintentionally limit pathways. By granting flexibility to learners in how they create music, music educators can open multiple pathways for students to engage in music making. This process may also include reevaluating and reimagining current music standards to include descriptive language and electronic music techniques.

Interest in Continuing to Make Music, But Not Necessarily with Code

Participants enjoyed making music with code but struggled to make connections to the value of making music with code. This may be attributed to the historical and cultural practices, norms, and recognized participating persons in Western music, for which making music with code is outside the tradition. Computers, however, are beginning to occupy space in Western music tradition, especially in music production with DAWs, effects plugins, and virtual instruments. Making music with code is becoming a popular composition and performance medium, with applications in music education, music production, and software development.[442]

Music for Leisure

Making music can be a valuable activity for leisure. Music education in the United States often prioritizes virtuosity in performance, especially at the high school level. This is problematic for a couple of reasons. First, music making goals of young learners are diverse and do not always align with performing. Second, the prioritization of performance virtuosity makes entry into music programs less accessible, especially at the high school level. Lastly, performance virtuosity implies that the purpose of music education in the United States is to produce virtuoso performers, a characteristic usually associated with professional musicians. Musicians frequently engage in other musical activities, sometimes to supplement their income, though these activities may also be for leisure, including: teaching, composing/arranging, booking, producing, or repairs.[443]

---

442. Freeman and Magerko, "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch," 64; Aaron, "Sonic Pi - Reliable Randomisation for Performances," 242; Rambarran, "'DJ Hit That Button' Amateur Laptop Musicians in Contemporary Music and Society," 586.

443. Stebbins, "Leisure Music Production Its Spaces and Places," 347.

Music activities in elementary and middle schools emphasize performance of a variety of musics. For this reason, a study investigating impacts of making music with code on younger participants may reveal opportunities to foster musical creativity with digital technologies.

Rather than engaging in the narrow focus of pursuing music for a career as a performer, music educators might emphasize music making as an activity that can be pursued outside of school and throughout one's life, in a variety of spaces.[444] This includes several forms of musical amateurism, which was the focus of early music education throughout the United States.[445] After the mid-twentieth century, music education in the United States "shifted away from amateurism toward promoting a type of semiprofessional musicianship that is more appropriate for the audition room and concert stage than for venues of informal music-making beyond high school and college."[446] First and foremost, to move forward with making music in novel ways may require a shift from thinking about music education which emphasizes performance virtuosity and toward a

444. Kratus, "A Return to Amateurism in Music Education," 35; Roger Mantie, "Music Education as Leisure Education," in *Music, Leisure, Education: Historical and Philosophical Perspectives*, ed. Roger Mantie (Oxford University Press, 2022), 209, https://doi.org/10.1093/oso/9780199381388.003.0009.

445. Kratus, "A Return to Amateurism in Music Education," 32.

446. Kratus, "A Return to Amateurism in Music Education," 32.

way that embraces new ideas and technologies in which learners construct their knowledge.[447] Participants 1 and 2 in this study characterized music as a hobby, while participant 3 identified contexts and spaces for which they would make music. Stebbin's Serious Leisure Perspective may identify participants 1 and 2 as hobbyists, particularly makers or tinkerers, and participant 3's engagement as project-based leisure.[448] While this study may have been a leisurely endeavor for the participants, their demonstration of learning or improving musical skills and increasing their understanding of music through Sonic Pi is testimony that making music with code is an activity anyone can pursue.

Time for, and space in leisure activities may be problematic for historically marginalized people(s). This did not seem to be an issue in this study; participants represented diverse races and genders, stated on their entrance questionnaires, and viewed the study as a leisure activity as indicated in their interviews. However, when considering leisure in music, White privilege may be a

---

447. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, 529–530.

448. Concepts - The Serious Leisure Perspective. (n.d.). The Serious Leisure Perspective (SLP). Retrieved March 9, 2023, from https://www.seriousleisure.net/concepts.html

relevant issue.[449] Mowatt recommends that we analyze our leisure and recreational programs for a perpetuation of non-White absence.[450]

Sonic Pi Compared to Other Music Making Tools

Code allows novel ways of making music that extend beyond standard features of most digital audio workstations (DAWs) and other music notation tools. Participant 1 indicated that making music with Sonic Pi was not as simple as other digital tools, for example, the DAW GarageBand. Participant 2 also indicated that they previously experimented with music making using the DAWs GarageBand and Reaper. Their familiarity with DAW workflow may have prompted them to draw comparisons with Sonic Pi. However, their comparisons of DAW workflow and Sonic Pi are limited by their experiences with each tool.

Low Floors, High Ceilings, Wide Walls

In Sonic Pi, the user can create a sound with as little as one line of code. Granted, this may not be as interesting as sequencing a melody; however, the user can construct knowledge about creating music in Sonic Pi *while* they create.

---

449. Rasul A Mowatt, "Notes From A Leisure Son: Expanding An Understanding of Whiteness in Leisure," *Journal of Leisure Research* 41, no. 4 (Fourth Quarter 2009): 515, https://doi.org/10.1080/00222216.2009.11950188.

450. Mowatt, "Notes From A Leisure Son: Expanding An Understanding of Whiteness in Leisure," 522.

This is consistent with Papert's idea of "low floors" and "high ceilings;" the user can be creative with little knowledge, and as they construct more knowledge, their creations are almost limitless.[451] This is also consistent with the artifacts of the participants in this study; as they constructed more knowledge, their compositions grew in complexity. Papert developed the Logo programming language at MIT as a means to facilitate this idea.[452] With Logo, "everyone of whatever age and whatever level of academic performance, to programming, to more general knowledge of computation and indeed ... to mathematics, to physics and to all the formal subjects including linguistics and music" constructs new knowledge using graphics, animations, music, and robotics.[453] In figure 32, Frere Jacques is realized in Logo.

---

451. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 908-909.

452. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play,* 569.

453. Papert and Solomon, "Twenty Things to Do with a Computer," 2.

```
TO FRERE1
1  SING MUSIC OF "1! 3! 5! 1!"  "2 2 2 2"
END

TO FRERE2
1  SING MUSIC "5! 6! 8!"  "2 2 4"
END

TO FRERE3
1  SING MUSIC '8! 10! 8! 6! 5! 1!"  "1 1 1 1 2 2"
END

TO FRERE4
1  SING MUSIC "1! -8! 1!"  AND  "2 2 4"
END

TO FREREJACQUES
1  FRERE1
2  FRERE1
3  FRERE2
4  FRERE2
5  FRERE3
6  FRERE3
7  FRERE4
8  FRERE4
9  FREREJACQUES
END
```

Figure 32. Frere Jacques, Seymour Papert & Cynthia Solomon, "Twenty Things to Do with a Computer," 24

Mitchel Resnick extended Seymour Papert's notion of low floors, high ceilings, with the idea of wide walls, because "It's not enough to provide a single path from a low floor to a high ceiling; it's important to provide multiple pathways."[454] While many pathways for coding support this notion, music is often limited. For example, Logo is limited to 5 octaves of pitch, a polyphony of 4 voices, and random choice of pitch.[455] Scratch, the successor to Logo, is problematic in

454. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play,* 909.

455. Papert and Solomon, "Twenty Things to Do with a Computer," 23-25.

different ways; "for users wishing to create music through sequencing musical notes or with engaging sounds and instruments, they often can face high floors, low ceilings and narrow walls due to the complex numeric music mappings and mathematical representations, as well as data structures that limit musical expression."[456] Sonic Pi, however, offers easy entry to music creation (low floors), robust functionality (high ceilings), and multiple ways for users to musically express themselves through code (wide walls).

456. Payne and Ruthmann, "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?"

```
1  define :frere1 do
2    play_pattern_timed [:C4, :D4, :E4, :C4], [1, 1, 1, 1]
3  end
4
5  define :frere2 do
6    play_pattern_timed [:E4, :F4, :G4], [1, 1, 2]
7  end
8
9  define :frere3 do
10   play_pattern_timed [:G4, :A4, :G4, :F4, :E4, :C4], [0.5, 0.5, 0.5,
   0.5, 1, 1]
11 end
12
13 define :frere4 do
14   play_pattern_timed [:C4, :G3, :C4], [1, 1, 2]
15 end
16
17 define :frerejacques do
18   frere1
19   frere1
20   frere2
21   frere2
22   frere3
23   frere3
24   frere4
25   frere4
26 end
27
28 frerejacques
```

Figure 33. Frere Jacques in Sonic Pi

In figure 33, I have written Frere Jacques in Sonic Pi using a similar procedure to the Logo example in figure 29. Aside from some differences in syntax, the procedure is the same with the exception of the recursive call of `FREREJACQUES` in figure 32, line 9 of `TO FREREJACQUES`. In figure 33, `frerejacques` on line 28 must be called outside the function definition to be heard. Both Logo and Sonic Pi support basic control structures. Where Sonic Pi pulls ahead is with its ability to utilize envelopes, samples, synthesizers, and of course, the `live_loop`. The code from figure 33 can be remixed into virtually any style of music, played in a variety of spaces, by any kind of musician, professional or amateur.[457] This makes Sonic Pi an ideal platform with a low floor, high ceiling, and wide walls that schools might consider for introducing learners to making music.

Access to Sonic Pi in schools is possible by simply installing Sonic Pi on existing computers running Windows, macOS, and most distributions of Linux. However, Sonic Pi does not have installers for ChromeOS (Chromebooks), iOS (iPad and iPhone), or Android (tablet and smartphone), which may be a barrier to schools or individual learners using these devices. The internet is only needed to access the installer file. Once installed, Sonic Pi will run without the internet.

DAWs, Effects, & Virtual Instruments

The concept of low floors, high ceilings, and wide walls can serve as an assessment tool for evaluating music creation tools. DAWs and notation software have a higher floor; the user will need to learn some fundamentals about the software user interface before they are able to create anything. The height of the ceiling and width of the walls are dependent on the features of the software.

---

457. Kratus, "A Return to Amateurism in Music Education," 32; Stebbins, "Leisure Music Production Its Spaces and Places," 350.

Some free DAWs and notation software offer several features; however, most DAWs and notation softwares bind features to the cost of the software. The installation of plugins like virtual instruments and effects further raise the floor by demanding the user learn additional workflows. On the other hand, the installation of plugins both raises the ceiling and widens the walls. Music educators considering using these technologies may also need to consider potential barriers for learners.

Software Development

Sonic Pi is a valuable resource to leverage in today's music classroom. Sonic Pi is available to users for free with no paid tiers and contains an extensive application programming interface (API) that removes the need for plugins. It is one of many music and sound programming languages, including Csound, Max, Pure Data, SuperCollider, and ChucK mentioned in Chapter 1.[458] What sets Sonic Pi apart from these programming languages? All of these programming languages are free with the exception of Max. Like Sonic Pi, Csound, SuperCollider, and ChucK are all text-based languages. Sonic Pi is criticized because it is software that "10-year-old kids [use] that's also going to be used by

458. Wang, Cook, and Salazar, "ChucK: A Strongly Timed Computer Music Language," 13.

professional musicians or professional programmers."[459] Sonic Pi provides an accessible, low floor with its high level, readable syntax. Although each of these languages are capable of high ceilings, the syntax of these languages may narrow the width of their walls. However, learning procedures and control structures in one language transfers to other languages. Benefiting from Sonic Pi's low floor, a new coder of music will more than likely gain skills that are transferable to other music and sound programming languages, opening opportunities for music and sound software development.

Music and sound software development presents numerous opportunities. For example, the reason why your washing machine or dryer might play a happy melody for you when it starts or finishes is because it was programmed by someone who understands both music and coding. This example is one of many electronic devices and technologies that we interact with that require knowledge of music and coding to manufacture. This also includes a number of small physical devices based on microcontroller technologies like doorbells or alarms; programming for these devices is often intended to provide aural cues to users on status updates. Musical elements are sometimes embedded using bespoke software; however, there are many applications, particularly in web development,

459. "Sam Aaron's Sonic Pi Can Help You Make Music Through Code."

in which the musical elements are coded.

There may exist some cases in software development in which musical elements are programmed with music or sound programming languages. For example, ChucK is used for the development of many Smule applications, has integrations for Unity Game Engine, and can run natively in web browsers.[460] In most software development, however, musical elements are programmed in languages with broad applications, for example: C, C++, Python, and JavaScript. Learning to code with music may provide a foundation for developing music and sound in software applications.

Musical Transactions Promote Computer Science

Numerous studies have demonstrated that music is beneficial for engaging learners in computer science education.[461] Sonic Pi's "simple, engaging musical narrative" is used to teach basic computing concepts.[462] Additionally, "Students'

---

460. Ge Wang, "Synthesis Chapter Four: Digital Synthesis Language Sampler 15," accessed March 13, 2023, https://cmtext.indiana.edu/synthesis/chapter4_synth_languages15.php.

461. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 663; Magerko et al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 16; Krug et al., "Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop."

462. Aaron, "Live Coding Education," 45.

natural interest in sound and music [is] an entrée through which to introduce STEM concepts."[463] During a 2013 pilot study, EarSketch was used successfully to improve student engagement and content knowledge in computing.[464] However, it is important for music educators to consider whether these integrations of music in computer science education have benefitted music education.

When the intent of an educational program is to further knowledge of a discipline (e.g., computer science) by integrating another discipline in a subservient role (e.g., music), learners may gain new understandings of both disciplines; however, a transaction is made because they are encouraged to pursue the primary discipline at the cost of the subservient discipline.[465] This problem is both curricular and pedagogical. The curriculum/educational program is intentionally designed with educational outcomes that support the primary discipline by utilizing engaging tools from the subservient discipline. The

---

463. Burg, Romney, and Schwartz, "Computer Science 'Big Ideas' Play Well in Digital Sound and Music," 663.

464. Magerko et. al., "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education," 5.

465. Liora Bresler, "The Subservient, Co-Equal, Affective, and Social Integration Styles and Their Implications for the Arts," *Arts Education Policy Review* 96, no. 5 (June 1, 1995): 33, https://doi.org/10.1080/10632913.1995.9934564.

teacher's biases and pedagogies, including the ways in which they present the educational content, may influence students' perceptions of the integrated disciplines. In either instance, when music is used to lure students to computer science education and computer science objectives, a music for computer science transaction has been made.

Many curricula that integrate music and computer science are intended to benefit computer science education. An exemplar of this approach, EarSketch, "teaches computing principles through digital music composition and remixing."[466] EarSketch, developed at Georgia Institute of Technology, embeds a JavaScript and Python editor in a DAW environment. EarSketch's curricular offerings have expanded to include several courses for secondary schools, most of which prioritize computer science education, including: Computer Science Principles (Regular or AP), Your Voice is Power, Project STEM: Introduction to Computer Science, CAPACITY: Culturally Authentic Practice to Advance Computational Thinking in Youth, in addition to Hour of Code tutorials.[467] Your Voice is Power

466. Jason Freeman, Brian Magerko, and Regis Verdin, "EarSketch: A Web-Based Environment for Teaching Introductory Computer Science Through Music Remixing," *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015, 5.

467. "Curriculum Catalogue," EarSketch Teachers, accessed March 14, 2023, https://www.teachers.earsketch.org/curriculum-catalogue.

is unique amongst the EarSketch curriculum offerings in that learners investigate how musicians impact changes in social justice and use EarSketch to facilitate music creation inspired by societal needs.[468] While the content is framed in historical music contexts and music creation, the curriculum is only aligned with CSTA computer science standards and not music standards despite specific foci on music history and elements of music, including song structure (form) and rhythm.[469]

A similar music and coding platform, TunePad, was developed by the Tangible Interaction Design and Learning (TIDAL) Lab at Northwestern University in collaboration with the EarSketch team at Georgia Institute of Technology. TunePad offers two pathways to learners: “Tunetorials,” a curriculum framed by music topics, and “Guidebook,” a curriculum framed by computer science topics. In both pathways, learners experience music and computer science; the pathway serves as a framework to relate learning materials to prior knowledge, experiences, or interests. TunePad’s curricula are not standards-aligned. TunePad was initially intended to teach coding with music

---

468. “YVIP Curriculum 2022: Tell Your Story - Google Drive,” accessed March 14, 2023, https://drive.google.com/drive/folders/1FL5KyJxTEmCmrVRHD4EdT04h_7-7pfJw.

469. “YVIP Curriculum 2022: Tell Your Story - Google Drive.”

because "music provides a particularly rich context through which to explore concepts of computer programming."[470] However, the most recent version of TunePad seems to better integrate both disciplines without an intentional transaction.

Sonic Pi differs from EarSketch and TunePad in that its development was motivated by varied individual and institutional objectives. Before Sonic Pi, Sam Aaron developed a code-based music platform called Overtone.[471] Aaron performed live-coded music on the platform prior to creating Sonic Pi. Aaron developed Sonic Pi for the Raspberry Pi Foundation to "demonstrate a new style of classroom teaching that was appropriate to the strategic ambitions of the Raspberry Pi Foundation."[472] Aaron was selected for development for his expertise, which at the time was risky because "creative experiences for students resulted in a degree of tension with the schools [in the United Kingdom] computing curriculum, which was clearly situated within the Science, Technology, Engineering and Mathematics (STEM) priorities of UK educational

---

470. Horn, Banerjee, and West, "Music and Coding as an Approach to a Broad-Based Computational Literacy," 84.

471. "Sam Aaron's Sonic Pi Can Help You Make Music Through Code."

472. Aaron, Blackwell, and Burnard, "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 79.

policy, rather than Arts and Music."[473] Aaron found that the "method of delivery and the means with which a teacher can engage the students with the content" was most problematic.[474] Aaron's intent with Sonic Pi is to continue to develop a platform consistent with low floors, high ceilings, and wide walls because he states that it is "a really cool thing for human expression that professional artists would like to use and how do I package that in a way that children can work with effectively."[475] The ways in which a learner interacts with Sonic Pi and chooses how to continue with it, whether it is informal or facilitated by a teacher, depends entirely on the goals of the learner (and the teacher).

Music education in schools may benefit by emphasizing making music as an activity that can be pursued outside of school and throughout one's life, perhaps as a professional, but just as importantly, for leisure.[476] Music educators are best positioned to facilitate this shift in thinking. Sonic Pi is an ideal platform

---

473. Aaron, Blackwell, and Burnard, "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming," 79.

474. "Sam Aaron's Sonic Pi Can Help You Make Music Through Code."

475. Resnick, *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*, 909; "Sam Aaron's Sonic Pi Can Help You Make Music Through Code."

476. Kratus, "A Return to Amateurism in Music Education," 34–35; Roger Mantie, "Music Education as Leisure Education," 209–10.

for engaging learners to make music with code. Sonic Pi is a medium that offers a low floor for entry, a high ceiling for exceptional functionality, and wide walls to accommodate diverse pathways from diverse users. Sonic Pi is also widely accessible; it is free of cost, available on most platforms, and only requires the internet to install the application. While digital audio workstations (DAWs) are creative tools for music production, they often present a high floor for entry and may require the addition of effects plugins and virtual instruments to raise its ceiling, which may impede the creative processes of their users.

The *music for computer science transaction* is both a curricular and pedagogical problem. Curriculum developers may need to consider how music and computer science are integrated and whether the integration is mutually beneficial. Educators may need to consider the learning outcomes they intend to teach and determine if the integration of music is transactional. If so, the educator might ask how learning in their classrooms can be modified so learners construct truly interdisciplinary knowledge. Educators may be best positioned to facilitate the greatest impact making music with code in constructionist ways because of their direct interaction with learners. Pre-written curriculums lack this direct connection with the learner. Music educators may benefit in the development of truly interdisciplinary music/computer science lessons by collaborating with computer science educators. Such interdisciplinary collaborations and experiences may enable new music coders to gain skills that are transferable to other music and sound programming languages that can open opportunities for music and sound software development.

APPENDICES

Appendix 1: IRB Expedited Approval

Charles River Campus Institutional Review Board
25 Buick Street, Suite 158
Boston, Massachusetts 02215
T 617-358-6115 / www.bu.edu/irb

BOSTON UNIVERSITY

Notification of IRB Approval: Expedited Review

December 14, 2021

Nathaniel Stottlemyer, MA
College of Fine Arts
855 Commonwealth Avenue
Boston, MA 02215

Protocol Title:	Musicking with Code: Composing & Performing using Sonic Pi with High-School Students
Protocol #:	6347E
Funding Agency:	Not funded
IRB Review Type:	Expedited, Category 7
IRB Approval Date:	December 14, 2021
IRB Expiration Date:	December 13, 2022

Dear Nathaniel Stottlemyer:

On December 14, 2021, the IRB reviewed the above-referenced protocol via expedited procedures in accordance with 45 CFR 46.110. The Reviewer has confirmed that the research meets criteria for IRB approval at 45 CFR 46.111.

In addition, the protocol approval includes:

- In accordance with 45 CFR 46.404 and 46 CFR 46.408, the IRB determined that the research did not involve greater than minimal risk, that assent would be obtained from the child, and that the permission of one parent is sufficient. Signed assent will be obtained.

This approval includes the following:

1.) 50 participants
2.) Approved application (initial submission)
3.) Forum and social medial recruitment instruments
4.) Entrance questionnaire
5.) Exit survey
6.) Interview questions

As the Principal Investigator, you are responsible for ensuring that this study is conducted in accordance with federal regulations, state laws, and institutional policies. As such please note:

- No participants may be involved in study procedures prior to the IRB approval date or after the expiration date.
- All unanticipated problems or serious adverse events must be reported to the IRB immediately.
- Any modifications made to this study must be reviewed and approved by the IRB in advance of implementation.
- Protocol events (including deviations, reportable adverse events and/or unanticipated problems etc.) must be reported to the IRB. Unexpected adverse events or unanticipated problems of any kind must be reported in writing to the IRB as they occur.
- For new studies using drugs and devices and new Investigators, a QI visit may be scheduled.
- Please submit an Annual Update Application six weeks prior to the expiration of your study.
- Failure to adhere to the IRB Policies and Procedures listed above and available on the BU CRC IRB website is considered non-compliance, which could result in termination of the study approval with notification to the study sponsor.

Any correspondence with the IRB office regarding this action should note the Protocol Number indicated at the top of this letter.

If you have any questions, please contact me at hartpj@bu.edu.

Sincerely,

Paul G. Hart, CIP, MA, JD
Senior IRB Analyst
Charles River Campus IRB

cc: Jared O'Leary, PhD

Appendix 2: Site Approval

CCS Collegium CHARTER SCHOOL

435 Creamery Way, Suite #300
Exton, PA 19341
610-903-1300 (phone)
610-903-1317 (fax)
www.collegiumcharter.com

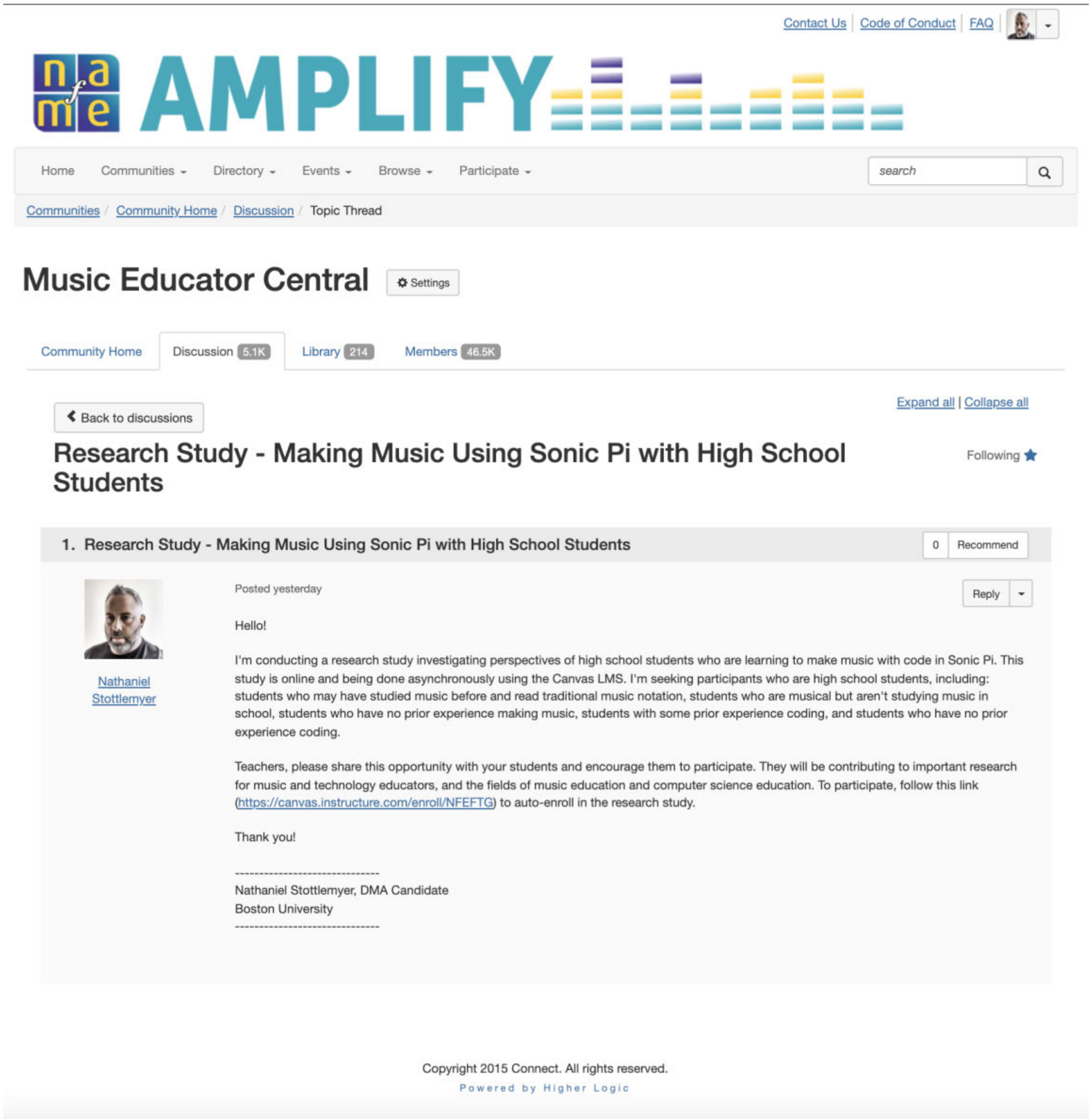
Dear IRB Reviewer:

If the proposal submitted by Nate Stottlemyer and team entitled Musicking with Code: Composing & Performing using Sonic Pi with High-School Students is approved by BU's IRB, it is my intent to collaborate as detailed in the submitted IRB proposal, which includes providing a meeting place for participants after school for extracurricular activities.

Sincerely,

Kristin Gallahan
kgallahan@ccs.us
484.401.2902

Appendix 3: Sample Forum Recruitment Post



Contact Us | Code of Conduct | FAQ

AMPLIFY

Home Communities Directory Events Browse Participate search

Communities / Community Home / Discussion / Topic Thread

Music Educator Central

Settings

Community Home | Discussion 5.1K | Library 214 | Members 46.5K

Back to discussions

Expand all | Collapse all

Research Study - Making Music Using Sonic Pi with High School Students

Following

1. Research Study - Making Music Using Sonic Pi with High School Students

0 Recommend

Nathaniel Stottlemyer

Posted yesterday

Reply

Hello!

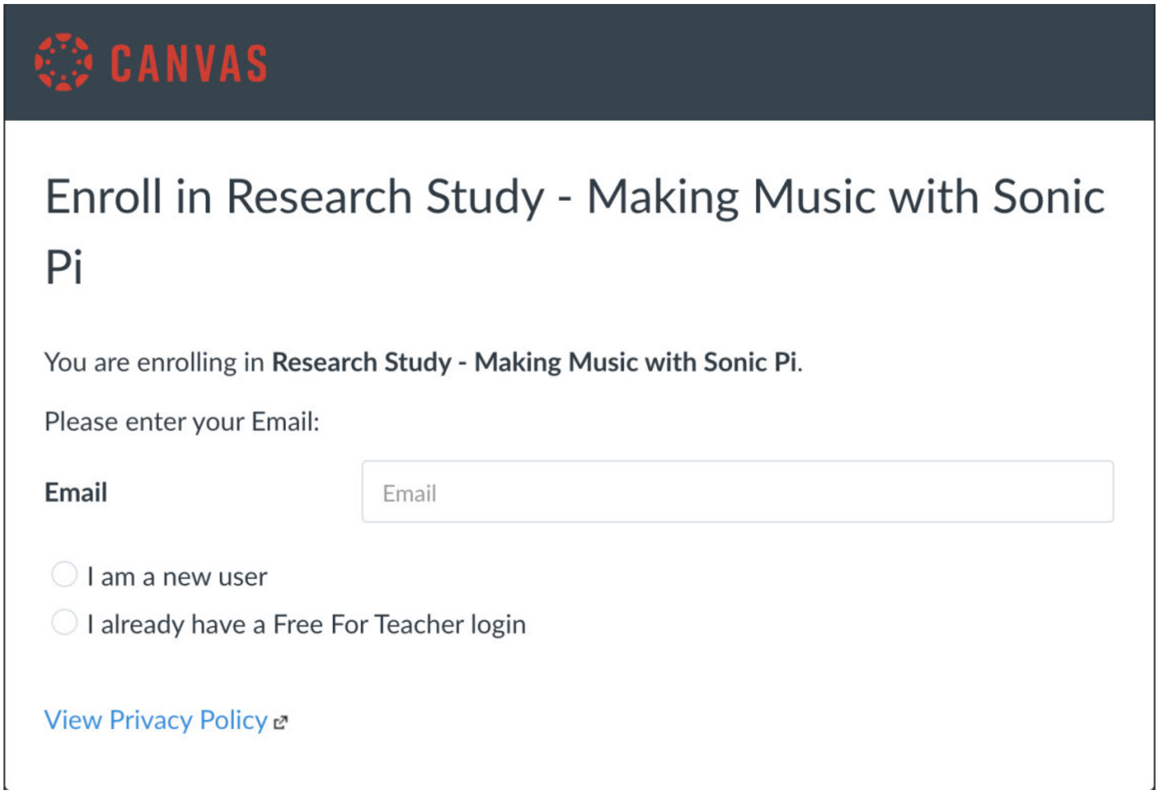
I'm conducting a research study investigating perspectives of high school students who are learning to make music with code in Sonic Pi. This study is online and being done asynchronously using the Canvas LMS. I'm seeking participants who are high school students, including: students who may have studied music before and read traditional music notation, students who are musical but aren't studying music in school, students who have no prior experience making music, students with some prior experience coding, and students who have no prior experience coding.

Teachers, please share this opportunity with your students and encourage them to participate. They will be contributing to important research for music and technology educators, and the fields of music education and computer science education. To participate, follow this link (https://canvas.instructure.com/enroll/NFEFTG) to auto-enroll in the research study.

Thank you!

Nathaniel Stottlemyer, DMA Candidate
Boston University

Appendix 4: Research Study Canvas Self-Enroll



Appendix 5: Letter of Consent/Assent

BOSTON UNIVERSITY

CONSENT/ASSENT FORM

Title of Project: **Musicking with Code: Composing & Performing using Sonic Pi with High School Students**

Principal Investigators: **Nathaniel Stottlemyer, M.A. & Jared O'Leary, Ph.D. (Advisor)**

Study Summary
The purpose of this research study is to investigate perceptions of high school students who study music-making with code.

Participants who take part in this research study will complete approximately 5 hours of study materials, self paced, within a 2-month period. Participants will access research study materials online using Canvas LMS at (link here).

Participants will engage in learning Sonic Pi, a virtual instrument programmed with the Ruby programming language, to create their own original music.

There are no major risks to the participants taking part in this research study. Anticipated risks include potential fatigue with completing tasks and loss of confidentiality in event of a breach.

Introduction
Please read this form carefully. The purpose of this form is to provide you with important information about taking part in a research study. If you have any questions about the research or any portion of this form, please ask us. Taking part in this research study is up to you. If you decide to take part in this research study we will ask you to sign this form. We will give you a copy of the signed form.

The person in charge of this study is Nathaniel Stottlemyer, a doctoral student at Boston University. Dr. Jared O'Leary, Mr. Stottlemyer's faculty advisor is supervising the study. Nathaniel Stottlemyer can be reached at nsto@bu.edu. Dr. O'Leary can be reached at jared.oleary@asu.edu. We will refer to these persons as the "researchers" throughout this form.

What should I know about a research study?
Participation in research is voluntary, which means that it is something for which you volunteer. It is your choice to participate in the study, or not to participate. If you choose to participate now, you may change your mind and stop participating later. If you decide not to participate, that decision will not result in any penalty or loss of benefits to which you are otherwise entitled.

Why is this study being done?
The purpose of this research study is to investigate perceptions of high school students who study music-making with code.

We are asking you to take part in this study because you are a high school student who has expressed interest in music and coding.

About 50 participants will take part in this research study nationally, across the United States.

Who is Funding the Study?
The study is unfunded.

How long will I take part in this research study?
We expect that you will be in this research study for 5 hours. During this time, we will ask you to access online materials using Canvas LMS at (link here).

BOSTON UNIVERSITY

What will happen if I take part in this research study?
If you agree to take part in this study, we will ask you to sign the consent form before we conduct any study procedures.

Study Activities
We will ask you to complete 11 study components in the Canvas LMS virtual space housing the research study. You may take breaks during the study. Below is a summary of what you may expect within each component.

Component	Expected Duration	You will be asked to do the following:
1	10 minutes	Complete the Statement of Consent/Assent.
2	20 minutes	Complete an entrance questionnaire regarding demographic information, and your experiences with music, code, and electronic devices.
3	15 minutes	Complete installation of Sonic Pi on a computer.
4	30 minutes	Complete and submit a "Live-Coding" music project with a computer. Live-coding is a unique aspect of Sonic Pi that enables you to write and modify code live to make music. This permits the music programmer to perform live for audiences.
5	30 minutes	Complete and submit a "Synths" music coding project with a computer. Synth is an abbreviation for synthesiser, which is a device used to create sounds with electricity. Sonic Pi makes using synths, which can be quite complicated, a simple and fun way to make music.
6	30 minutes	Complete and submit a "Samples" music coding project with a computer. Samples are pre-recorded sounds. When you record a sound, you've just created a sample. Sonic Pi has several built-in public domain samples ready for you to use. In this component, you will play and manipulate them.
7	30 minutes	Complete and submit a "Randomization" music coding project with a computer. Using random numbers allows you to simulate chance in your music. This means you can compose chance in your music.
8	30 minutes	Complete and submit a "Effects" music coding project with a computer. Sonic Pi allows you to easily add studio effects to your music. For example, you can simulate different environments with reverb or echo, or use distortion to creatively mangle your sounds.
9	60 minutes	Complete and submit "Your Musical Creation" music coding project with a computer. Your musical creation will be a culmination of the skills you learned throughout the research study. You will choose and apply skills that highlight your musical interests.
10	10 minutes	Complete an exit survey rating your perceptions about the study.
11	30 minutes	Complete an exit interview regarding your perceptions about the study. We will be using Boston University's secure Zoom account to conduct and record these interviews. Zoom requires the use of a web browser but does not require any software download. For more information about Zoom security and privacy, please see the Boston University webpage on Zoom Meetings, or ask the research team. Although all internet communication we have with you via Zoom remote study visits are kept confidential, and the communication platforms that we use are password-protected, Boston University cannot fully guarantee the privacy or security of any content sent electronically including Zoom recordings.

What are the risks of taking part in this research study?

Risks of Completing Tasks
You may get tired during the tasks. You can rest at any time.

Loss of Confidentiality
The main risk of allowing us to use and store your information for research is a potential loss of privacy. We will protect your privacy by labeling your information with a code and keeping the key to the code in a password-protected computer with encryption.

BOSTON UNIVERSITY

You will be informed of any significant new findings developed during the course of this research which may affect your willingness to continue participation.

The following people or groups may review your study records for purposes such as quality control or safety:

- The Researcher and any member of their research team
- The Institutional Review Board at Boston University. The Institutional Review Board is a group of people who review human research studies for safety and protection of people who take part in the studies.
- Central University Offices

Are there any benefits from being in this research study?
You may or may not benefit from taking part in this study. Possible benefits may include the acquiring of new musical and/or programming knowledge.

Others may benefit in the future from the information that is learned in this study.

Study Participation and Early Withdrawal
Taking part in this study is your choice. You are free not to take part or to withdraw at any time for any reason. No matter what you decide, there will be no penalty or loss of benefit to which you are entitled. If you decide to withdraw from this study, the information that you have already provided will be kept confidential.

Also, the researcher may take you out of this study without your permission. This may happen because:

- The researcher thinks it is in your best interest
- You can't make the required study visits
- Other administrative reasons

Use of Your Study Information
Personally identifiable information collected from you during this study will NOT be shared publicly, used for future research studies, or shared with other researchers for future research.

Will I get paid for taking part in this research study?
We will not pay you for taking part in this study.

What will it cost me to take part in this research study?
There are no costs to you for taking part in this research study.

Who do I ask if I have questions or concerns about this research study?
Please contact us with any concerns or questions about the research, or any research-related problems:

- Nathaniel Stottlemyer can be reached at nsto@bu.edu.
- Dr. Jared O'Leary can be reached at jared.oleary@asu.edu.

If you have questions about your rights as a research participant, or if you have any complaints or concerns and want to speak with someone independent of the research team, you may contact the Boston University IRB at 617-358-6115. The IRB Office webpage has information where you can learn more about being a participant in research, and you can also complete a Participant Feedback Survey.

Statement of Consent/Assent

I have read the information in this consent/assent form including risks and possible benefits. I have been given the chance to ask questions. My questions have been answered to my satisfaction, and I agree to participate in the study.

SIGNATURE

________________________________ __________________

Name of Study Participant Age of Study Participant

________________________________ __________________

Signature of Study Participant Date

Parental Permission (If participant is under 18 years of age):

I give permission for my child to participate in this research study.

____

Name of Parent

________________________________ __________________

Signature of Parent Date

To be completed by the researcher:

I have explained the research to the research participant and answered all their questions. I will give a copy of the signed consent form to the participant.

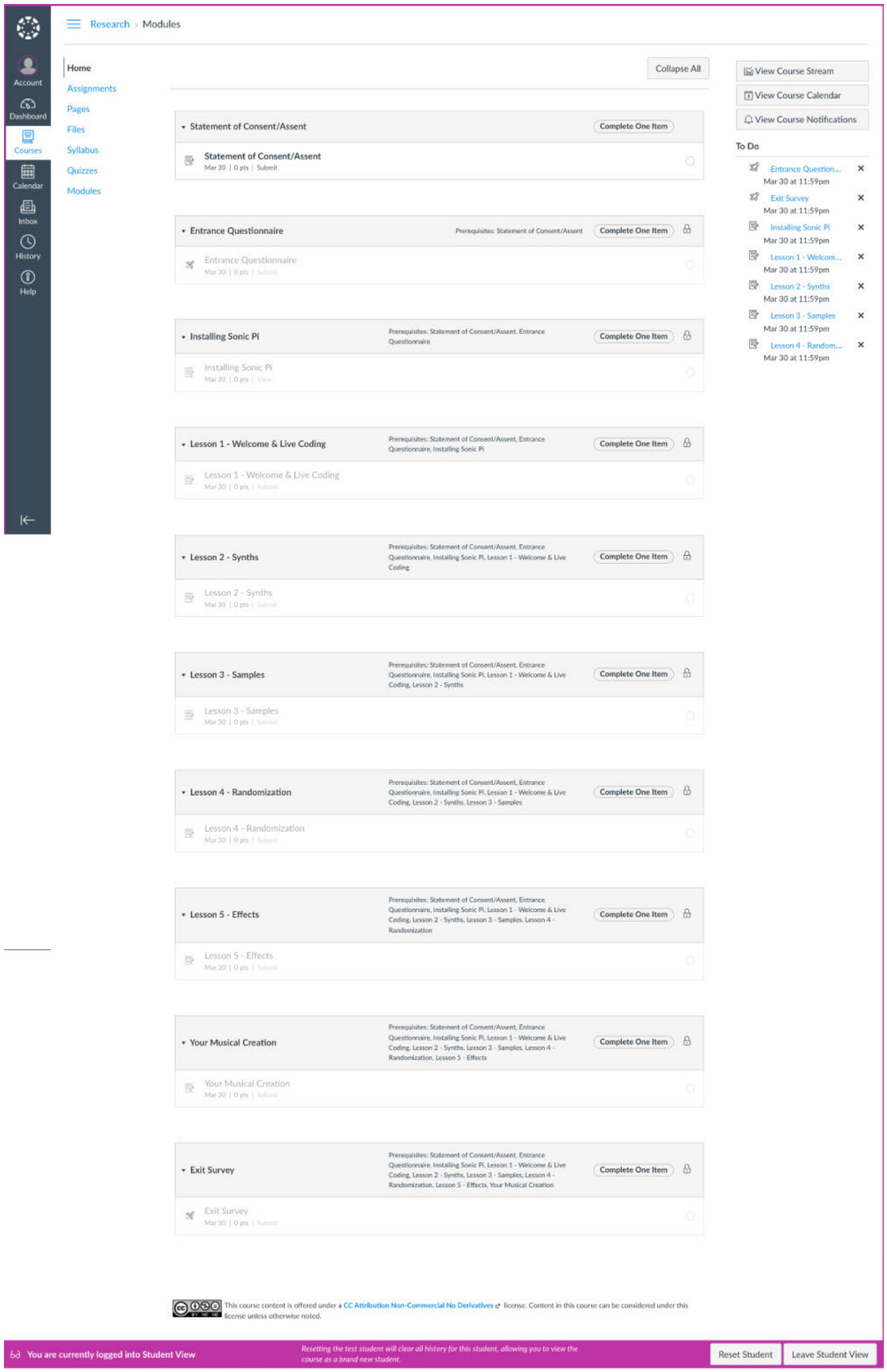
NATHANIEL STOTTLEMYER______________

Name of Researcher Obtaining Consent

________________________________ __________________

Signature of Researcher Obtaining Consent Date

Appendix 6: Research Study - Participant View



Appendix 7: Entrance Questionnaire

BOSTON UNIVERSITY

Protocol Title: MUSICKING WITH CODE: COMPOSING & PERFORMING USING SONIC PI WITH HIGH-SCHOOL STUDENTS	
Principal Investigator: Nathaniel Stottlemyer, M.A. & Jared O'Leary, Ph.D. (Advisor)	
Description of Study Population: Participants are high school students from various remote locations throughout the United States.	
Version Date: 11-21-2021	

Participant Entrance Questionnaire

1. Have you ever created music of your own? If so, how?
2. Do you use any digital electronic devices? How are digital electronic devices useful to you?
3. What does the term "music technology" mean to you?
4. Do you read music notation? If so, what kind of notation(s) do you read and how did you learn to read music notation?
5. Are you in any music classes or lessons currently? If not, have you taken any music classes or lessons in the past?
6. Do you currently make music? If so, how (e.g. in a band, a community organization, drum circle, with a DAW, etc.)?
7. What are your opinions about the following questions?
   a. Do you have to read traditional music notation to create music?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

   b. Do you consider yourself a musician?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

   c. Do you consider yourself a composer?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely
8. Do you have experience programming? If so, what languages are you experienced with?
9. Are you in any computer science classes currently?

BOSTON UNIVERSITY

Demographic Survey

Participants are requested to complete this demographic survey. This information is voluntary.

1. What gender do you identify as?
    a. Male
    b. Female
    c. Nonbinary
    d. Prefer not to answer
2. Please specify your ethnicity. (Select all that apply)
    a. White/Caucasian
    b. Black/African-American
    c. Hispanic/Latino/Latina/Latinx
    d. Asian
    e. Native American/Alaskan
    f. Native Hawaiian/Pacific Islander
    g. Other/Unknown ____________
    h. Prefer not to answer
3. Which languages are you capable of speaking fluently? (Select all that apply)
    a. English
    b. Spanish
    c. French
    d. Mandarin
    e. Cantonese
    f. Korean
    g. Japanese
    h. Arabic
    i. Other ___________
    j. Prefer not to answer

Appendix 8: Exit Survey

BOSTON UNIVERSITY

Protocol Title: MUSICKING WITH CODE: COMPOSING & PERFORMING USING SONIC PI WITH HIGH-SCHOOL STUDENTS	
Principal Investigator: Nathaniel Stottlemyer, M.A. & Jared O'Leary, Ph.D. (Advisor)	
Description of Study Population: Participants are high school students from various remote locations throughout the United States.	
Version Date: 11-21-2021	

Participant Exit Survey

1. Do you have to read traditional music notation to create music?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

2. Do you consider what you created during this study to be music?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

3. Do you feel you have learned something new as a result of this study?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

4. Do you consider yourself a musician?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

5. Do you consider yourself a composer?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

6. Now that the study is over, are you interested in continuing to make music with code?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

7. Are you less interested or more interested in making music than you were at the beginning of this study?

   Less Interested ☐ ☐ ☐ ☐ ☐ ☐ More Interested

8. Are you interested in learning more coding, but not necessarily related to coding music?

   Not at all ☐ ☐ ☐ ☐ ☐ ☐ Absolutely

Appendix 9: Interview Questions

BOSTON UNIVERSITY

Protocol Title: MUSICKING WITH CODE: COMPOSING & PERFORMING USING SONIC PI WITH HIGH-SCHOOL STUDENTS	
Principal Investigator: Nathaniel Stottlemyer, M.A. & Jared O'Leary, Ph.D. (Advisor)	
Description of Study Population: Participants are high school students from various remote locations throughout the United States.	
Version Date: 11-21-2021	

Participant Interview Questions

1. Was this a valuable experience? Why or why not?
2. Would you recommend this experience to others? What about this experience makes it something you would recommend?/What about this experience prevents you from recommending it?
3. Did you experience anything unexpected during this experience? What was unexpected?/How was it predictable?
4. What part of this experience influenced your thoughts to continue/discontinue making music?

IF participant already read any variety of music notation:
1. How did you learn to read music notation?
2. How did this experience enrich your understanding of music notation?

IF participant was experienced writing code:
1. What programming languages are you familiar with?
2. How did this experience enrich your understanding of coding?

Appendix 10: Using Sonic Pi to Listen to Examples

1. Download and install Sonic Pi from https://sonic-pi.net/ for your operating system.
2. Open Sonic Pi.
3. Copy and paste code examples to the Sonic Pi editor, excluding numbers from the left column.
4. Click "run" and listen to your chosen example.

Appendix 11: Case 1, Lesson 1 Artifact

```
live_loop :foo do
  sample :ambi_choir, rate: 0.5
  sample :perc_impact1, rate: 2
  sleep 1
end

live_loop :lol do
  sample :bd_haus, rate: 1
  sleep 2
end
```

Appendix 12: Case 1, Lesson 2 Artifact

```
live_loop :foo do
  play 60, amp: 2
  play 70, amp: 0.5
  sleep 0.5
  play 50, amp: 3
  sleep 0.25
  play 60, attack: 0.5, attack_level: 1, decay: 1, sustain_level: 0.4,
sustain: 2, release: 0.5
  sleep 1
end
```

Appendix 13: Case 1, Lesson 3 Artifact

```
live_loop :foo do
  sample :loop_amen
  sample :ambi_drone
  play 60
  sample :loop_electric
  sleep 1.7
end
```

Appendix 14: Case 1, Lesson 4 Artifact

```
loop do
  play rrand(30, 90)
  sleep 0.5
  play choose([20, 50, 80])
  sleep 0.5
  play rand
  sleep 1
end
```

Appendix 15: Case 1, Lesson 5 Artifact

```
in_thread do
  loop do
    sample :drum_heavy_kick
    sleep 1
  end
end

in_thread do
  loop do
    use_synth :fm
    play 40, release: 0.2
    sleep 0.5
  end
end

in_thread do
  loop do
    use_synth :dpulse
    play 52, sustain: 2, amp: 0.5
    sleep 2
  end
end

loop do
  use_synth :subpulse
  play 60
  sleep 1
end
```

Appendix 16: Case 1, Capstone Artifact

```
in_thread do
  loop do
    sample :drum_heavy_kick, amp: 2
    sleep 1
  end
end

in_thread do
  loop do
    use_synth :fm
    play 40, release: 0.2
    sleep 0.5
  end
end

in_thread do
  loop do
    use_synth :dpulse
    play 52, attack: 0.5, sustain: 2, release: 0.5, amp: 0.25
    sleep 2
  end
end

loop do
  use_synth :subpulse
  play 60
  sleep 0.5
  play 70
  sleep 0.5
  play 80
  sleep 0.5
  play 70
  sleep 0.5
end
```

Appendix 17: Case 2, Lesson 1 Artifact

```
# Welcome to Sonic Pi
use_synth :chiplead

play :C3
sleep 0.5
play :D3
sleep 0.5
play :E3
sleep 0.5
play :F3
sleep 0.5
play :G3
sleep 0.5
play :A3
sleep 0.5
play :B3
sleep 0.5
play :C4
```

Appendix 18: Case 2, Lesson 2 Artifact

```
use_synth :saw
play 72, attack: 2, release: 0.2
sleep 1
play 75
sleep 1
play 79
sleep 1
play 82
```

Appendix 19: Case 2, Lesson 3 Artifact

```
# Welcome to Sonic Pi

use_synth :saw
play 72, attack: 2, release: 0.2
sleep 1
play 75
sleep 1
play 79
sleep 1
play 82
sample :loop_amen, rate: 0.6
```

Appendix 20: Case 2, Lesson 4 Artifact

```
use_synth :fm
loop do
  play rrand(50, 95)
  sleep rrand(0.1, 0.5)
end
```

Appendix 21: Case 2, Lesson 5 Artifact

```
define :fxTest do

  with_fx :reverb do
    play 50
    sleep 0.5
    play 60
  end
  sleep 0.5
  play 50
end

fxTest
```

Appendix 22: Case 2, Capstone Artifact

```
define :song do
  2.times do
    background
  end

  in_thread do
    4.times do
      melody
    end
  end

  2.times do
    background
  end
end

define :background do
  play :C5
  play :G4
  sleep 0.5
  play :D4
  sleep 0.5
  play :C4
  sleep 0.5
  play :C5
  play :G4
  sleep 0.5
  play :D4
  sleep 0.5
  play :C4
  sleep 0.5
  play :A4
  sleep 0.5
  play :C5
  play :G4
  sleep 0.5
  play :D4
  sleep 0.5
```

```
  play :C4
  sleep 0.5
  play :C5
  play :G4
  sleep 0.5
  play :D4
  sleep 0.5
  play :C4
  sleep 1
end

define :melody do
  play :C5
  play :C4
  sleep 0.5
  play :D5
  sleep 0.5
  play :C5
  sleep 0.5
  play :C5
  play :C4
  sleep 0.5
  play :D5
  sleep 0.5
  play :C5
end

use_synth :kalimba
song
```

Appendix 23: Case 2, Capstone Artifact Second Submission

```
intro_music = [[:G4, :C4], :D4, :C4]

a_music = [[:C6, :G5], :B5, :G5, [:G5, :C6], [:G5, :B5], :E5, :A5,
          [:G5, :C6], :B5, :G5, [:G5, :C6], :B5, :C5,
          [:G5, :C6, :E5], [:B5, :G5], :G5, [:G5, :C6, :E5], :E5, [:D6, :A5],
          [:E6, :G5], :G6, :C6, [:E6, :G5], :G6, :C6]
a_timing = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 1]

a_background_music = [[:C4, :C3], :D4, :C4, :C4, :D4, :C4]
a_background_timing = [0.5, 0.5, 0.5, 0.5, 0.5, 1.0]

b_music = [[:B5, :G5], :A5, :E5, [:B5, :E5], :A5, :Gs5]
b_timing = [0.5, 0.5, 0.5, 0.5, 0.5, 1.0]

b_background_music = [:A4, :D5, :A5, :A4, :D5, :A5, :Fs5]
b_background_timing = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]


define :intro do
  2.times do
    play_pattern_timed intro_music, [0.5]
  end
  play :A4
  sleep 0.5
  play_pattern_timed intro_music, [0.5]
  play_pattern_timed intro_music, [0.5, 0.5, 1]
end

define :play_section do |music, timing, background, background_timing, repeat|
  in_thread do
    play_pattern_timed music, timing
  end
  repeat.times do
    play_pattern_timed background, background_timing
  end
end

define :haggstrom do
  2.times do
    intro
  end
  play_section a_music, a_timing, a_background_music, a_background_timing, 4
```

```
  /play_section b_music, b_timing, b_background_music, b_background_timing, 1/
end

use_synth :kalimba
haggstrom
```

Appendix 24: Case 3, Lesson 1 Artifact

```
live_loop :multi_beat do
  use_random_seed 2000
  8.times do
    c = rrand(70, 130)
    n = (scale :e1, :minor_pentatonic).take(3).choose
    synth :tb303, note: n, release: 0.1, cutoff: c if rand < 0.9
    sample :elec_hi_snare if one_in(6)
    sample :drum_cymbal_closed if one_in(2)
    sample :drum_cymbal_pedal if one_in(3)
    sample :bd_haus, amp: 1.5 if one_in(4)
    sleep 0.125
  end
end
```

Appendix 25: Case 3, Lesson 2 Artifact

```
# Welcome to Sonic Pi

play 72
sleep 0.5
play 75
sleep 0.5
play 79
sleep 0.5
play 80
sleep 0.5
play 72
sleep 0.5
play 79
sleep 0.5
play 80
sleep 0.5
play 70
```

Appendix 26: Case 3, Lesson 3 Artifact

```
sample :ambi_choir, rate: 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sample :elec_bong, attack: -0.5
sample :bass_hard_c, attack: -0.5
sleep 0.5
sample :bass_hard_c, attack: -0.5
sleep 0.5
sample :bass_hard_c, attack: -0.5
sleep 0.5
sample :elec_filt_snare
sample :bass_hard_c, attack: -0.5
sleep 0.5
sample :ambi_choir, rate: 0.5
```

Appendix 27: Case 3, Lesson 4 Artifact

```
loop do
  play 60, amp: rand
  sleep 0.2
  play 70, amp: rand
  sample :ambi_choir
end
```

Appendix 28: Case 3, Lesson 5 Artifact

```
with_fx :reverb do
  with_fx :wobble, phase: 0.5, decay: 8 do
    play 60
    sleep 0.5
    sample :ambi_drone
    sleep 0.5
    play 62
    sleep 0.5
    sample :ambi_piano
    sleep 0.5
    play 60, rate: -0.5
  end
end
```

Appendix 29: Case 3, Capstone Artifact

```
with_fx :reverb do
  with_fx :wobble, phase: 0.5, decay: 8 do
    play 60
    sleep 0.5
    sample :ambi_drone
    sleep 0.5
    play 62
    sleep 0.5
    sample :ambi_piano
    sleep 0.5
    play 60, rate: -0.5
  end
end

sample :ambi_choir, rate: 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sleep 0.5
sample :drum_bass_hard, attack: -0.5
sample :elec_bong, attack: -0.5
sample :bass_hard_c, attack: -0.5
sleep 0.5
sample :bass_hard_c, attack: -0.5
sleep 0.5
sample :bass_hard_c, attack: -0.5
sleep 0.5
sample :elec_filt_snare
sample :bass_hard_c, attack: -0.5
sleep 0.5
sample :ambi_choir, rate: 0.5
```

```
loop do
  play 60, amp: rand
  sleep 0.2
  play 70, amp: rand
  sample :ambi_choir
end
```

BIBLIOGRAPHY

Aaron, Sam. "Live Coding Education," *The MagPi Educator's Edition*, Education Special Issue 2, 44-47, https://www.raspberrypi.org/magpi-issues/MagPi-EduEdition02.pdf.

Aaron, Samuel. "Sonic Pi - Reliable Randomisation for Performances." *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC* 2016-Novem (2016): 242–243. https://doi.org/10.1109/VLHCC.2016.7739697.

Aaron, Samuel, Alan F Blackwell, and Pamela Burnard. "The Development of Sonic Pi and Its Use in Educational Partnerships: Co-Creating Pedagogies for Learning Computer Programming." *Journal of Music, Technology, and Education* 9, no. 1 (2016): 75–94. https://doi.org/10.1386/jmte.9.1.75_1.

Ariza, Christopher. "An Open Design for Computer-Aided Algorithmic Music Composition: AthenaCL." PhD diss. – New York University. ProQuest Dissertations Publishing, 2005. (Publication No. 3195500).

Armitage, Joanne, and Helen Thornham. "Don't Touch My MIDI Cables: Gender, Technology and Sound in Live Coding." *Feminist Review* 127, no. 1 (2021): 90–106. https://doi.org/10.1177/0141778920973221.

Ball, Philip. "Can We Use Quantum Computers to Make Music?" Physics World, February 28, 2023. https://physicsworld.com/can-we-use-quantum-computers-to-make-music/.

Barton, Todd. "6 — Gestures: Sculpting Energy." Buchla Blog, n.d. https://toddbarton.com/2014/03/6-gestures-sculpting-energy/.

Beckstead, David. "Will Technology Transform Music Education?" *Music Educators Journal* 87, no. 6 (2001): 44–49.

Benedict, Cathy, and Jared O'Leary. "Reconceptualizing 'Music Making:' Music Technology and Freedom in the Age of Neoliberalism." *Action, Criticism, and Theory for Music Education* 18, no. 1 (2019): 26–43. https://doi.org/10.22176/act18.1.26.

Benward, Bruce and Marilyn Saker. *Music in Theory and Practice* (Vol. 1). New York, NY: McGraw-Hill Higher Education, 2009.

Beynon, Meurig. “Mindstorms Revisited: Making New Construals of Seymour Papert’s Legacy.” In *Educational Robotics in the Makers Era*, 560:3–19, 2017. https://doi.org/10.1007/978-3-319-55553-9.

Bovermann, Till, and Dave Griffiths. “Computation as Material in Live Coding.” *Computer Music Journal* 38, no. 1 (2014): 40–53. https://doi.org/10.1162/COMJ.

Bresler, Liora. “The Subservient, Co-Equal, Affective, and Social Integration Styles and Their Implications for the Arts.” *Arts Education Policy Review* 96, no. 5 (June 1, 1995): 31–37. https://doi.org/10.1080/10632913.1995.9934564.

Brooks, Kevin, and Chris Lindgren. “Responding to the Coding Crisis: From Code Year to Computational Literacy.” In *Strategic Discourse: The Politics of (New) Literacy Crises*, edited by Lynn C. Lewis. Computers and Composition Digital Press, 2015. http://hdl.handle.net/10919/96777.

Brown, Andrew R., and Andrew Sorensen. “Interacting with Generative Music through Live Coding.” *Contemporary Music Review* 28, no. 1 (2009): 17–29. https://doi.org/10.1080/07494460802663991.

Bula, Josh A. “Technology-Based Music Courses and Non-Traditional Music Students in Secondary Schools.” Ph.D. Diss. – Florida State University. ProQuest Dissertations Publishing, 2011. (Publication No. 3502826)

Burg, Jennifer, Jason Romney, and Eric Schwartz. “Computer Science ‘Big Ideas’ Play Well in Digital Sound and Music.” *SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 2013, 663–68. https://doi.org/10.1145/2445196.2445391.

Cain, Tim. “Theory, Technology and the Music Curriculum.” *British Journal of Music Education* 21, no. 2 (2004): 215–21. https://doi.org/10.1017/S0265051704005650.

Campbell, Patricia Shehan, Claire Connell, and Amy Beegle. “Adolescents' Expressed Meanings of Music in and out of School.” *Journal of Research in Music Education* 55, no. 3 (2007): 220-236. http://www.jstor.org/stable/4543122.

Cardenas, Alexandra. "Live Coding: A New Approach to Musical Composition." *Dancecult* 10, no. 1 (2018). https://doi.org/10.12801/1947-5403.2018.10.01.14.

Cayari, Christopher. "Virtual Vocal Ensembles and the Mediation of Performance on YouTube." PhD diss. University of Illinois at Urbana-Champaign. ProQuest Dissertations Publishing, 2016. (Publication No. 10301985)

Cheng, Lee. "Teaching Live Coding of Electronic Dance Music: A Case Study." *Dancecult* 10, no. 1 (2018). https://doi.org/10.12801/1947-5403.2018.10.01.10.

Collins, Karen. "In the Loop: Creativity and Constraint in 8-Bit Video Game Audio." *Twentieth-Century Music* 4, no. 2 (2007): 209–27. https://doi.org/10.1017/S1478572208000510.

Collins, Nick. "Live Coding and Teaching SuperCollider." *Journal of Music, Technology, and Education* 9, no. 1 (2016): 5–16. https://doi.org/10.1386/jmte.9.1.5_1.

Collins, Nick, Alex McLean, Julian Rohrhuber, and Adrian Ward. "Live Coding in Laptop Performance." *Organised Sound: An International Journal of Music Technology* 8, no. 3 (2003): 321–330.

Concepts - The Serious Leisure Perspective. (n.d.). The Serious Leisure Perspective (SLP). Retrieved March 9, 2023, from https://www.seriousleisure.net/concepts.html.

Cremata, Radio. "The Use of Music Technology Across the Curriculum in Music Education Settings: Case Studies of Two Universities". D.M.A. Diss. – Boston University. ProQuest Dissertations Publishing, 2010. (Publication No. 3430388)

Creswell, John W. *Research Design: Qualitative, Quantitative, and Mixed-Methods Approaches.* Thousand Oaks: SAGE Publications Inc., 2009. Kindle.

Deutsch, Herbert A. "Where Was Technology and Music Education Twenty Years Ago?" *Journal of Popular Music Studies* 21, no. 1 (2009): 90–96. https://doi.org/10.1111/j.1533-1598.2009.01171.x.

Dillon, Steven. *Music, Meaning and Transformation: Meaningful Music Making for Life.* United Kingdom: Cambridge Scholars Publishing, 2007. https://eprints.qut.edu.au/6703/.

Dillon, Steven C. "The Student as Maker: An Examination of the Meaning of Music to Students in a School and the Ways in Which We Give Access to Meaningful Music Education." PhD diss., Institute for Education LaTrobe University, 2001. https://eprints.qut.edu.au/17077/.

Dockstader, Tod. "Morton Subotnick: The Wild Bull a Composition for Electronic-Music Synthesizer." *The Musical Quarterly* 55, no. 1 (1969): 136–139.

EarSketch Teachers. "Curriculum Catalogue." Accessed March 14, 2023. https://www.teachers.earsketch.org/curriculum-catalogue.

Elpus, Kenneth, and Carlos R. Abril. "Who Enrolls in High School Music? A National Profile of U.S. Students, 2009–2013." *Journal of Research in Music Education* 67, no. 3 (2019): 323–338. https://doi.org/10.1177/0022429419862837.

Falbel, Aaron. "Constructionism: Tools to Build (and Think) With." *LEGO Dacta*, n.d.

Freeman, Jason, and Brian Magerko. "Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch." *Journal of Music, Technology, and Education* 9, no. 1 (2016): 57–74.

Freeman, Jason, Brian Magerko, and Regis Verdin. "EarSketch: A Web-Based Environment for Teaching Introductory Computer Science Through Music Remixing." *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015, 5.

Freire, Paulo. *Pedagogy of the Oppressed.* New York, NY: Bloomsbury, 2000. Kindle.

Friedson, Steven M. *Dancing Prophets: Musical Experience in Tumbuka Healing.* Chicago, Illinois: The University of Chicago Press, 1996.

Gibbs, Mark. "Sonic Pi: Realtime Music Creation for the Raspberry Pi (and More). *Network World Online* (2016): 2.

https://www.networkworld.com/article/3053644/sonic-pi-realtime-music-creation-for-the-raspberry-pi-and-more.html

"GLOSSARY for National Core Arts: Music STANDARDS." National Coalition for Core Arts Standards, n.d. https://www.nationalartsstandards.org/sites/default/files/NCCAS%20GLOSSARY%20for%20Music%20Standards%201%20column.pdf.

Goldman, Andrew. "Live Coding Helps to Distinguish between Embodied and Propositional Improvisation." *Journal of New Music Research* 48, no. 3 (2019): 281–93. https://doi.org/10.1080/09298215.2019.1604762.

GOTOpia. "Sam Aaron's Sonic Pi Can Help You Make Music Through Code," March 6, 2023. https://gotopia.tech/articles/220/Sam-Aaron-SonicPi-can-help-you-make-music-through-code.

Greenfield, Michael D., Henkjan Honing, Sonia A. Kotz, and Andrea Ravignani. "Synchrony and Rhythm Interaction: From the Brain to Behavioural Ecology." *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 376, no. 1835 (2021): 20200234. https://doi.org/10.1098/rstb.2020.0324.

Han, Jinseung. "Digitally Processed Music Creation (DPMC): Music Composition Approach Utilizing Music Technology." Ed.D. Diss. – Teachers College, Columbia University. ProQuest Dissertations Publishing, 2011. (Publication No. 3484387).

Hancock, Oliver. "Play-Based, Constructionist Learning of Pure Data: A Case Study." *Journal of Music, Technology, and Education* 7, no. 1 (2014): 93–112.

He, Zhengting. "How to Create Delay-Based Audio Effects on the TMS320C672x DSP." Dallas, 2005. https://www.ti.com/lit/an/spraaa5/spraaa5.pdf.

Hiller, L. A., and L. M. Isaacson. "Musical Composition with a High-Speed Digital Computer." *Journal of the Audio Engineering Society* 6, no. 3 (1958): 154–60.

Horn, Michael S, Amartya Banerjee, and Melanie West. "Music and Coding as an Approach to a Broad-Based Computational Literacy." In *Non-Formal and Informal Science Learning in the ICT Era*, edited by Michail Giannakos, 83–97. Singapore: Springer Singapore, 2020. https://doi.org/10.1007/978-981-15-6747-6_5.

Hughes, Alayna. “Maker Music: Incorporating the Maker and Hacker Community into Music Technology Education.” *Journal of Music, Technology, and Education* 11, no. 3 (2018): 288. http://dx.doi.org/10.1386/jmte.11.3.287_1

Isbell, Daniel S. “Musicians and Teachers: The Socialization and Occupational Identity of Preservice Music Teachers.” *Journal of Research in Music Education* 56, no. 2 (2008): 162–78. https://doi.org/10.1177/0022429408322853.

Kisliuk, Michelle. *Seize the Dance: BaAka Musical Life and the Ethnography of Performance*. New York, NY: Oxford University Press, 1998.

Kratus, John. “A Return to Amateurism in Music Education.” *Music Educators Journal* 106, no. 1 (2019): 31–37. https://doi.org/10.1177/0027432119856870.

———. “Music Education at the Tipping Point.” *Music Educators Journal* 94, no. 2 (November 1, 2007): 42–48. https://doi.org/10.1177/002743210709400209.

Krug, Lusa Douglas, Edtwuan Bowman, Taylor Barnett, Lori Pollock, and David Shepherd. “Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop.” In *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 397–403. Association for Computing Machinery, Inc, 2021. https://doi.org/10.1145/3408877.3432424.

Kruse, Adam J. “‘They Wasn’t Makin’ My Kinda Music’: A Hip-Hop Musician’s Perspective on School, Schooling, and School Music.” *Music Education Research* 18, no. 3 (2016): 240–253. https://doi.org/10.1080/14613808.2015.1060954.

Lyon, Eric. “Dartmouth Symposium on the Future of Computer Music Software: A Panel Discussion.” *Computer Music Journal* 26, no. 4 (2002): 13–30. https://doi.org/10.1162/014892602320991347.

Magerko, Brian, Jason Freeman, Tom Mcklin, Mike Reilly, Elise Livingston, Scott Mccoid, and Andrea Crews-Brown. "EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education." *ACM Transactions on Computing Education (TOCE)* 16, no. 4 (2016): 1–25. https://doi.org/10.1145/2886418.

Magnusson, Thor. "Algorithms as Scores: Coding Live Music." *Leonardo Music Journal* 21 (2011): 19–23. https://doi.org/10.1162/LMJ_a_00056.

———. "Scoring with Code: Composing with Algorithmic Notation." *Organised Sound: An International Journal of Music Technology* 19, no. 3 (2014): 268–75. https://doi.org/10.1017/S1355771814000259.

Manaris, Bill, Blake Stevens, and Andrew R Brown. "JythonMusic: An Environment for Teaching Algorithmic Music Composition, Dynamic Coding and Musical Performativity." *Journal of Music, Technology, and Education* 9, no. 1 (2016): 35. http://dx.doi.org/10.1386/jmte.9.1.33_1

Mantie, Roger. "Music Education as Leisure Education." In *Music, Leisure, Education: Historical and Philosophical Perspectives*, edited by Roger Mantie. Oxford University Press, 2022. https://doi.org/10.1093/oso/9780199381388.003.0009.

Margolis, Jerome N. "A School Synthesizer Program Comes of Age." *Music Educators Journal* 74, no. 4 (1987): 32–36. https://doi.org/10.2307/3397960.

Martinez, Anthony, and Cheridan Christnact. "Women Are Nearly Half of U.S. Workforce but Only 27% of STEM Workers." Census.gov. Accessed April 2, 2023. https://www.census.gov/library/stories/2021/01/women-making-gains-in-stem-occupations-but-still-underrepresented.html.

McCoid, Scott, Jason Freeman, Brian Magerko, Christopher Michaud, Tom Jenkins, Tom Mcklin, and Hera Kan. "EarSketch: An Integrated Approach to Teaching Introductory Computer Music." *Organised Sound* 18, no. 2 (2013): 146–160. https://doi.org/10.1017/S135577181300006X.

Merriam, Alan P. *The Anthropology of Music*. Evanston, Illinois: Northwestern University Press, 1964.

Mowatt, Rasul A. “Notes From A Leisure Son: Expanding An Understanding of Whiteness in Leisure.” *Journal of Leisure Research* 41, no. 4 (Fourth Quarter 2009): 511-528, https://doi.org/10.1080/00222216.2009.11950188.

“National Association for Music Education: Music Standards (Music Technology).” National Association for Music Education, 2014.

“National Association for Music Education: PK–8 General Music Standards.” National Association for Music Education, 2014.

“National Core Arts Standards: Music - Music Technology Strand.” National Core Arts Standards, 2014. www.nationalcoreartsstandards.org.

“National Core Arts Standards: PreK-8 General Music.” State Education Agency Directors of Arts Education, 2014.

Ness, Catherine Van, and Jake Varn. “Governors Leading On K-12 Digital Access State Successes In Bridging The Digital Divide For K-12 Students.” National Governors Association, 2021. https://www.nga.org/news/commentary/governors-prioritize-expanding-internet-access-for-k-12-students/.

Nielsen, Teresa R. “Teen Playlist: Music Discovery, Production, and Sharing Among a Group of High School Students.” D.M.A. Diss. – Boston University, 2016. http://hdl.handle.net/2144/19561.

O’Leary, Jared. “A Corpus-Assisted Discourse Analysis of Chiptune-Related Practices Discussed within Chipmusic.Org.” PhD diss., (Arizona State University, 2019) 205.

———. *#CSK8 Podcast*, 106. “Lifelong Kindergarten with Mitch Resnick.” Interview by Jared O’Leary, aired October 25, 2021, on Jared O’Leary //Multiplicity. https://jaredoleary.com/csk8feed/106.

———. “Intersections of Popular Musicianship and Computer Science Practices.” *Journal of Popular Music Education* 4, no. 2 (2020): 153–174. https://doi.org/10.1386/jpme_00023_1.

Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 2020. Kindle.

Papert, Seymour, and Cynthia Solomon. "Twenty Things to Do with a Computer." A.I. Laboratory, Massachusetts Institute of Technology, June 1971. https://dspace.mit.edu/handle/1721.1/5836

Payne, William, and S. Alex Ruthmann. "Music Making in Scratch: High Floors, Low Ceilings, and Narrow Walls?" *The Journal of Interactive Technology & Pedagogy*, no. 15 (May 16, 2019). https://jitp.commons.gc.cuny.edu/music-making-in-scratch-high-floors-low-ceilings-and-narrow-walls/.

Perecman, Ellen and Sara R. Curran, *A Handbook for Social Science Field Research Essays & Bibliographic Sources on Research Design and Methods.* Thousand Oaks: Sage Publications, 2006. Kindle.

Petrie, Christopher. "Programming Music with Sonic Pi Promotes Positive Attitudes for Beginners." *Computers & Education* 179, no. March 2021 (2021): 104409. https://doi.org/10.1016/j.compedu.2021.104409.

Picard, Rosalind W., Seymour Papert, Walter Bender, Bruce Blumberg, Cynthia Breazeal, David Cavallo, Tod Machover, Mitchel Resnick, Deb Roy, and Carol Strohecker. "Affective Learning - A Manifesto." *BT Technology Journal* 22, no. 4 (2004): 253–269. https://doi.org/10.1023/B:BTTJ.0000047603.37042.33.

Pizà, Antoni. *Listening to The World: A Brief Survey of World Music*. New York: The City University of New York. Accessed March 10, 2023. https://pressbooks.cuny.edu/apiza/.

Pope, Stephen Travis, John Rahn, Carlos Cerana, Haruhiro Katayose, Frank Pecquet, and Richard Karpen. "Touched by Machine? Composition and Performance in the Digital Age." *Computer Music Journal* 19, no. 3 (1995): 13–17. https://doi.org/10.2307/3680650.

Puckette, Miller. *The Theory and Technique of Electronic Music.* Hackensack: World Scientific Publishing Co. Pte. Ltd., 2007. http://msp.ucsd.edu/techniques/v0.11/book.pdf.

Rambarran, Shara. "'DJ Hit That Button' Amateur Laptop Musicians in Contemporary Music and Society." In *The Oxford Handbook of Music Making and Leisure*, edited by Roger Mantie and Gareth Dylan Smith, 585–600. Oxford University Press, Incorporated, 2017. http://ebookcentral.proquest.com/lib/bu/detail.action?docID=4745366.

Resnick, Mitchel. "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," In *C&C '07: Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, 1–6, 2007. https://doi.org/10.1145/1254960.1254961.

———. *Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play*. Cambridge, MA: The MIT Press, 2017. Kindle.

———. "Technologies for Lifelong Kindergarten." *Educational Technology Research and Development* 46, no. 4 (1998): 43–55. https://doi.org/10.1007/BF02299672.

Ripani, Giulia. "Children's Representations of Music, Musical Identities, and Musical Engagement: Content and Socio-Demographic Influences." *Journal of Research in Music Education* 70, no. 3 (2021): 271–296.

Saldaña, Johnny. *The Coding Manual for Qualitative Researchers*. 4th ed. London, England: Sage Publications Ltd., 2021.

Schembri, Frankie. "Ones and Zeroes, Notes and Tunes." *MIT Technology Review*, February 21, 2018. https://www.technologyreview.com/2018/02/21/145234/ones-and-zeroes-notes-and-tunes/.

Seeger, Anthony. *Why Suyá Sing: A Musical Anthropology of an Amazonian People*. Chicago, Illinois: University of Illinois Press, 2004.

Shaw, Mia, and Kafai, Yasmin. "Charting the Identity Turn in K-12 Computer Science Education: Developing More Inclusive Learning Pathways for Identities." International Society of the Learning Sciences (ISLS), 2020. https://doi.dx.org/10.22318/icls2020.114.

Sonic Pi, "Programming Structures," Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial#section-5.

Sonic Pi, "Randomisation," Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial#section-4.

Sonic Pi, “Samples,” Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial#section-3.

Sonic Pi, “Synths,” Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial#section-2.

Sonic Pi, “Welcome to Sonic Pi,” Sonic Pi - Tutorial, n.d., https://sonic-pi.net/tutorial#section-1.

“Standard MIDI Files (SMF) Specification.” Accessed March 16, 2023. https://www.midi.org/specifications-old/item/standard-midi-files-smf.

Stake, Robert E. “Qualitative Case Studies.” In *The Sage Handbook of Qualitative Research*. Edited by N. K. Denzin and Y. S. Lincoln. Sage Publications Ltd., 2007.

Stebbins, Robert A. “Leisure Music Production Its Spaces and Places.” In *The Oxford Handbook of Music Making and Leisure*, edited by Roger Mantie and Gareth Dylan Smith, 347–361. Oxford University Press, Incorporated, 2017. http://ebookcentral.proquest.com/lib/bu/detail.action?docID=4745366.

———. “The Serious Leisure Perspective.” In *The Idea of Leisure: First Principles*. Taylor & Francis Group, 2012.

Thompson, Penny. “Cognitive Development: The Theory of Jean Piaget,” in *Foundations of Educational Technology* (Oklahoma State University Libraries, 2019) np. https://doi.org/10.22488/okstate.19.000002.

Vee, Annette. *Coding Literacy: How Computer Programming Is Changing Writing*. Cambridge: Cambridge: MIT Press, 2017.

Wang, Ge. “Synthesis Chapter Four: Digital Synthesis Language Sampler 15.” Accessed March 13, 2023. https://cmtext.indiana.edu/synthesis/chapter4_synth_languages15.php.

———. “The ChucK Audio Programming Language. ‘A Strongly-Timed and On-the-Fly Environ/Mentality.’” Ph.D. diss. – Princeton University ProQuest Dissertations Publishing, 2008. (Publication No. 3323202)

Wang, Ge, and Perry R Cook. "2004: On-the-Fly Programming: Using Code as an Expressive Musical Instrument." In *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*, edited by Alexander Refsum Jensenius and Michael J Lyons, 193–210. Cham: Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-47214-0_13.

Wang, Ge, Perry R Cook, and Spencer Salazar. "ChucK: A Strongly Timed Computer Music Language." *Computer Music Journal* 39, no. 4 (2015): 10–29. https://doi.org/10.1162/COMJ a 00324.

Williams, David A. "The Elephant in the Room." *Music Educators Journal* 98, no. 1 (2011): 51–57. https://doi.org/10.1177/0027432111415538.

Williams, David Brian. "Reaching the 'Other 80%:' Using Technology to Engage 'Non-Traditional Music Students' in Creative Activities." In *Tanglewood II "Technology and Music Education" Symposium*. University of Minnesota, 2007.

Williams, Roger Neil. "Investigating Culturally Responsive Teaching in the Jamaican Secondary Music Classroom: A Multiple Case Study." D.M.A. diss. – Boston University, 2022. https://open.bu.edu/handle/2144/43567

"YVIP Curriculum 2022: Tell Your Story - Google Drive." Accessed March 14, 2023. https://drive.google.com/drive/folders/1FL5KyJxTEmCmrVRHD4EdT04h_7-7pfJw

CURRICULUM VITAE

[redacted]

[redacted]

[redacted]