A Case Study of Middle Schoolers' Use of Computational Thinking Concepts and Practices during Coded Music Composition



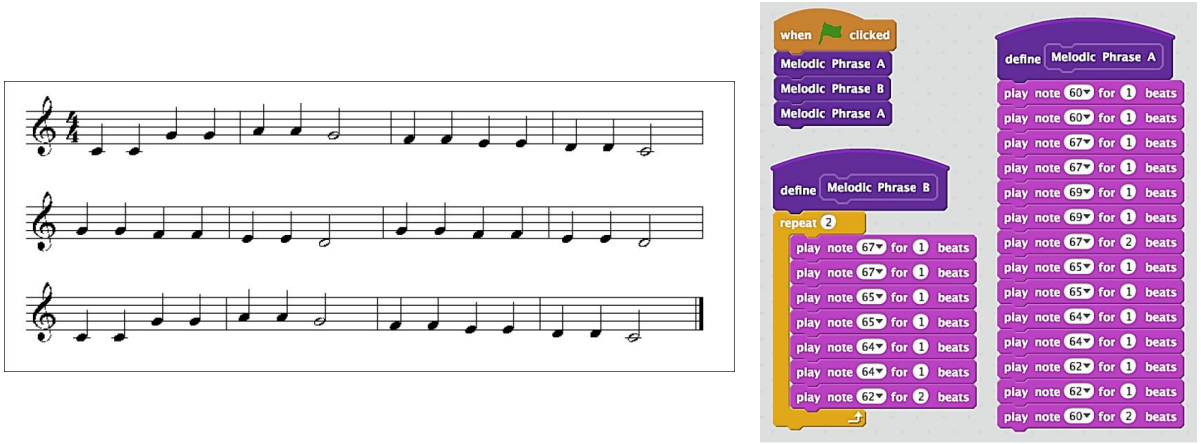

Yifan Zhang, Douglas Lusa Krug, Chrystalla Mouza, David C. Shepherd, and Lori Pollock

ericzh@udel.edu

This paper is based upon work supported by the National Science Foundation under grants 2048793 and 2048792.

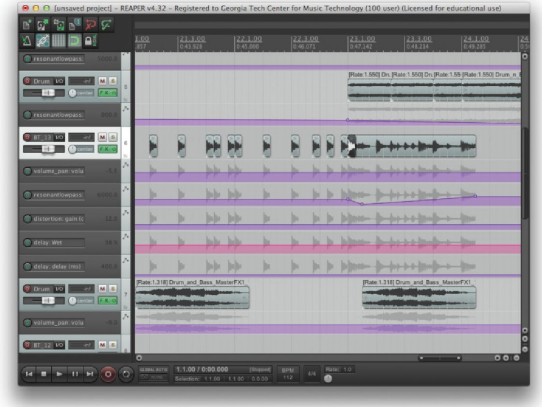
Teaching Computational Thinking (CT) with Music

- The integration of music and CT is particularly promising [Baratè et al., 2017; Bell and Bell, 2018; Petrie, 2021]

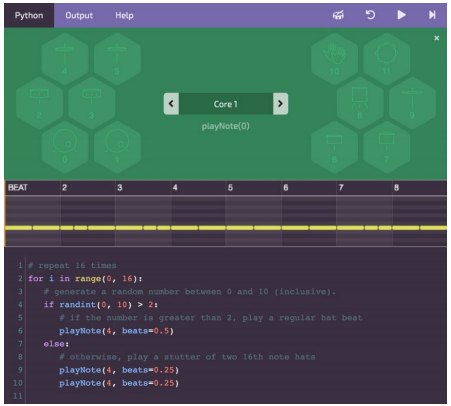


Twinkle Twinkle Little Star in Common Music Notation (left) and in Scratch (right) [Bell and Bell, 2018]

Supported by Music Coding Environments

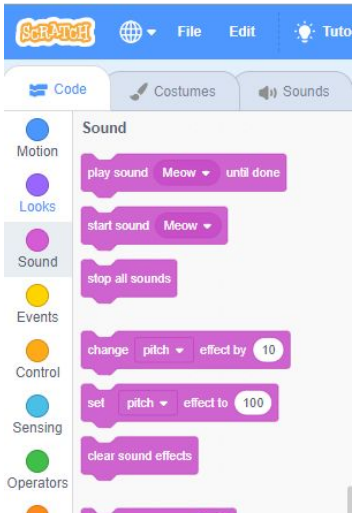


EarSketch, Python, remixing music
[Magerko et al., 2019]



TunePad, Python, interactive web-based
[Horn et al., 2020]

Sonic Pi, Ruby, live coding
[Aaron, 2016]



Scratch, Block-based
[Resnick et al., 2009;
Greher and Heines, 2014]

And, Curricula for Integrating Music & CT

- Media computation course for university non-majors [Guzdial, 2003]
- Coding exercises for middle school students in Blockly [Baratè et al., 2017]
- Connect CT and music with K-12 students in Scratch [Bell and Bell, 2018]
- Code Beats, teaches middle school students to program using Sonic Pi and more recently, TunePad [Krug et al., 2021; this paper]

Research Studies on Music & CT Integration

- Demonstration of CT [Petrie, 2021]
- Engagement [Engelman et al., 2017; Freeman et al., 2019; Jamshidi and Marghitu, 2019]
- Creativity [Engelman et al., 2017; Freeman et al., 2019]
- Affective outcomes [McKlin et al., 2019; Köppe 2020; Burnard et al., 2016]
- Better student performance and reduced course dropout rate [Tarversaro et al., 2020]

CT = Concepts + Practices

- Researchers highly recommend examining CT concepts and practices simultaneously
  [Brennan and Resnick, 2012; Zhang and Nouri, 2019; Allsop, 2019; Horst et al., 2020]
- However, research focusing on CT concepts and practices during integration of CT and music is lacking

Concepts	Practices
● **Sequences** ● **Loops** ● **Parallelism** ● **Data** ● Events ● Conditionals ● Operators	● **Being incremental and iterative** ● **Testing and debugging** ● Reusing and remixing ● Abstracting and modularizing

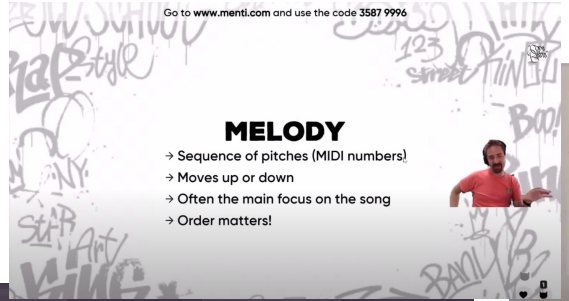
[Brennan and Resnick, 2012]

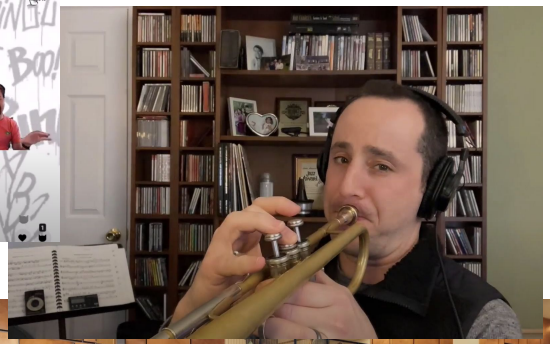
Our Contribution

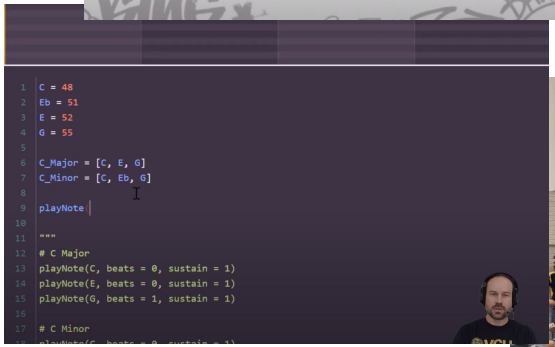
- First to analyze process logs to explore CT practices during integrated music & CT learning in K-12
  - Focus on tinkering behaviors
- Also analyzed coding products to identify CT concepts during music coding

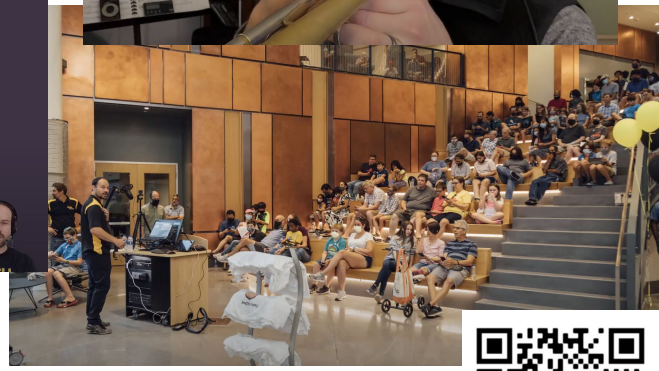
Our Study Context: Code Beats Camp

- 2-week (10 days) summer camp
- 1-hour online streamed learning session
- 1-hour office hour
- Open-ended after-class programming assignments
- Capstone project competition



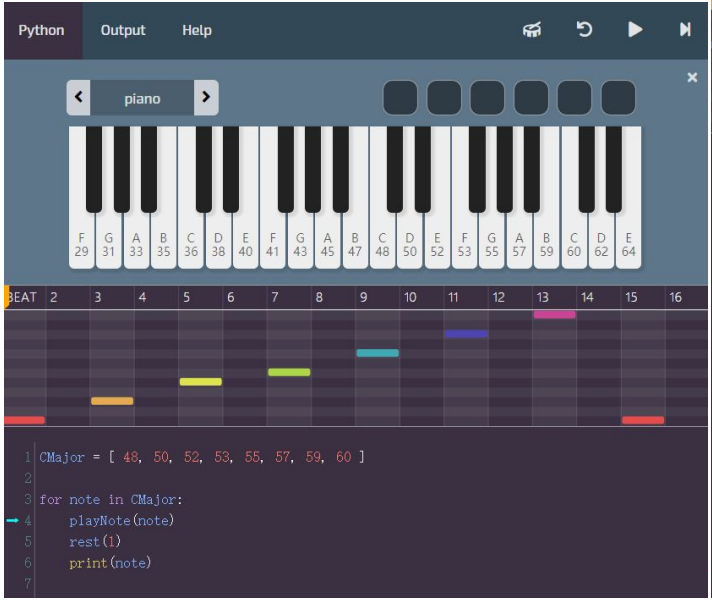


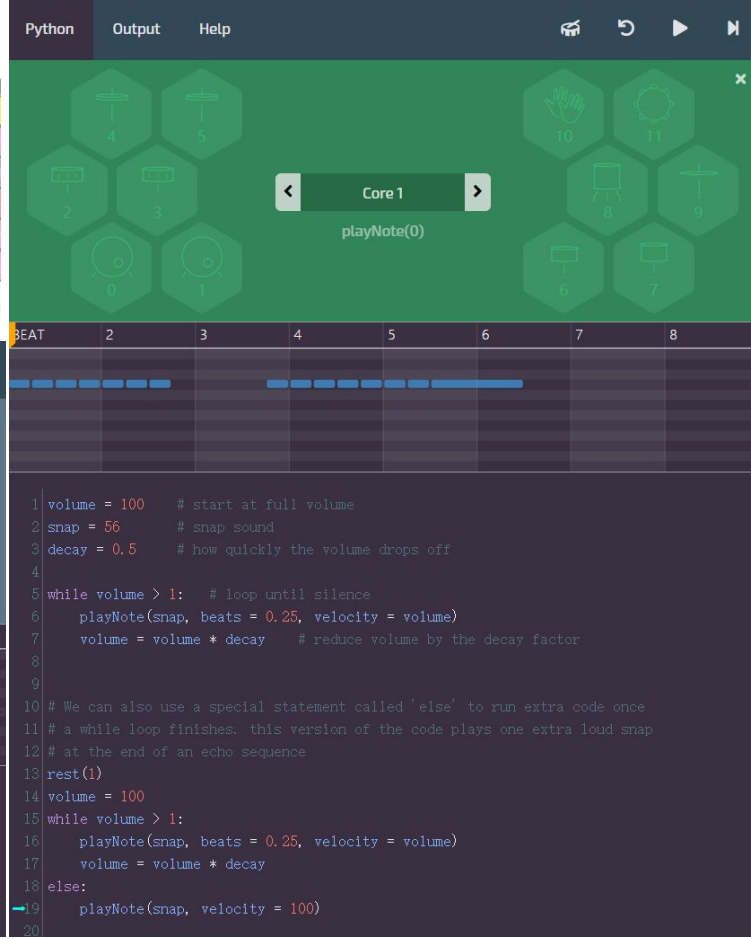




TunePad

www.tunepad.live





Case Study Activity Focus

- Day 5, day 8, and capstone projects
- Day 5
  - Given a song with 8 background instruments
  - Create 2 more instruments
- Day 8
  - Given a song with 6 background instruments
  - Create 2 more instruments

Category	Items	Day 5	Day 8
Coding Concepts	Lists	√	
	Repetition with nested lists		√
Music Concepts	Chords	√	
	Chord progression		√
CT Concepts	Sequence	√	√
	Parallelism	√	√
	Data	√	√
	Loop		√
Task Requirements	Define 4 new chords	√	
	Use given chords		√
	Each measure has 4 beats	√	√
Task Recommendations	Play recommended chords for each measure	√	
	Use *for* with *lists* to create a melody		√

Research Questions

- RQ1: What CT concepts are evident in participants' coding products? What percentage of participants met the requirements and recommendations of daily activities?
- RQ2: What CT practices, in the form of tinkering behaviors, were exhibited in participants' process logs during music coding in daily activities?
- RQ3: How did participants' tinkering behaviors during daily tasks compare with their tinkering behaviors during the final competition capstone task?

Data Collection

- 195 enrolled, **132** consented as participants
- Collaborated with TunePad developer to log coding process data
- Logged **138,735 coding events** and associated code snapshots over consented participants
  - *edit-instrument*
  - *error-instrument*
  - *play-instrument*
  - *play-project*
- Line-based granularity

	# of participants	# of events logged
Overall	132	138,735
Day 5	22	2,260
Day 8	16	1,459
Capstone project	14	21,110

Data Analysis

- *RQ1: What CT concepts are evident in participants' coding products? What percentage of participants met the requirements and recommendations of daily activities?*
- Developed metrics based on requirements and recommendations
- Examine code products

RQ1 Findings

- Majority showed success in defining (even nested) chords/lists and using given chords/lists
- Only 10-50% ensured the required 4 beats in each measure
  - We observed evidence of exploration in RQ2
  - Maybe caused by creativity of music
- Over half used loops when recommended

Concept	Require/Recommend	Num of Participants			
		Day 5 (n = 22)		Day 8 (n = 16)	
Note (Variable)	Define new notes		0		1 (6.3%)
	Change notes		1 (4.5%)		0
Chord (List)	Define required chords	R	15 (68.2%)		
	Define nested chords				10 (62.5%)
	Use given chords			R	15 (93.8%)
	Play recommend chords	C	7 (31.8%)		
Function	4 beats in each measure	R	2 (9.1%)	R	8 (50%)
	1-5 functions called		7 (31.8%)		
	6-10 functions called		4 (18.2%)		
	11-15 functions called		4 (18.2%)		
	15+ functions called		7 (31.8%)		
	0 function changed		14 (63.6%)		
	1-2 functions changed		7 (31.8%)		
	3+ functions changed		1 (4.5%)		
Loop	Use loop			C	9 (56.3%)

Data Analysis

- *RQ2: What CT practices, in the form of tinkering behaviors, were exhibited in participants' process logs during music coding in daily activities?*

- *RQ3: How did participants' tinkering behaviors during daily tasks compare with their tinkering behaviors during the final competition capstone task?*

Tinkering behaviors [Dong et al., 2019]

- **Construction-based tinkering**
- **Test-based tinkering**
- Prototype-based tinkering

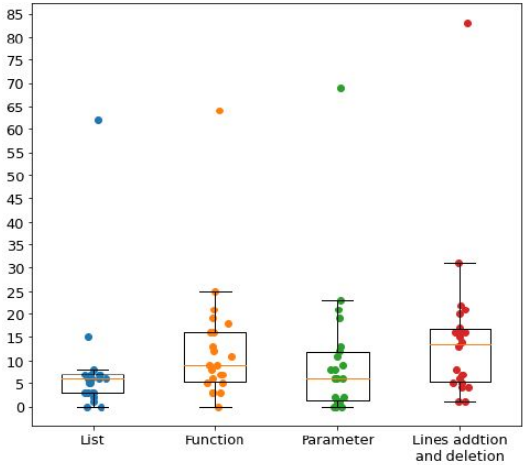
Data Analysis

- **Construction-based tinkering**
  - Token changes, including *list, function, parameter,* and *loop*
  - Adding or deleting lines of code
  - More changes, more construction-based tinkering
- **Test-based tinkering**
  - Number of edits between plays (i.e., code executions)
  - Lower number of edits between plays, more test-based tinkering

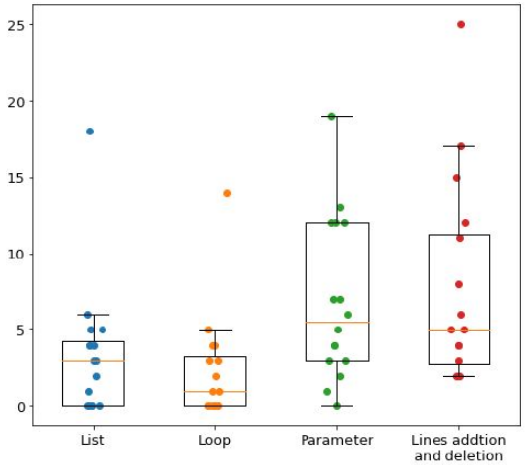
Construction-based Tinkering Findings

Number of changes per participant

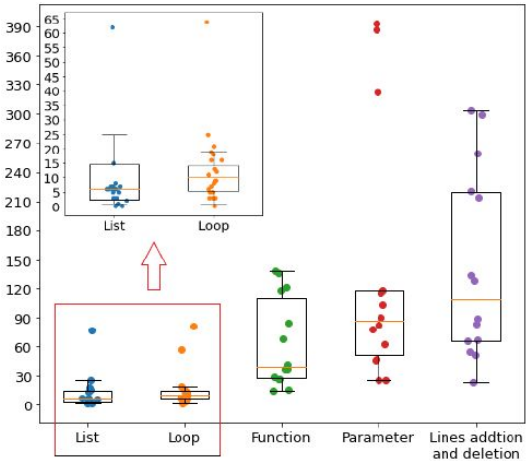
Token and number of line changes for day 5 (n=22)



Token and number of line changes for day 8 (n=16)



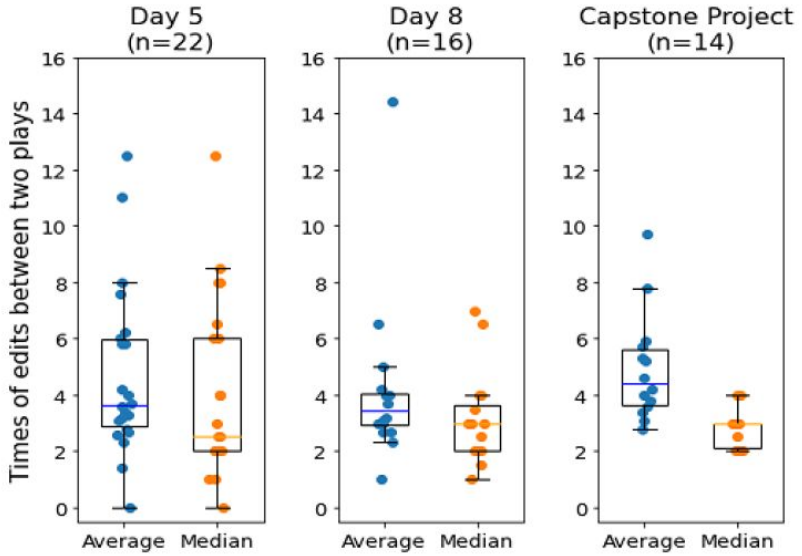
Token and number of line changes for capstone project (n=14)



- Fairly broad range of construction-based tinkering among participants
- Considerably different scale for capstone project while similar patterns among camp days and capstone project

Test-based Tinkering Findings

- The majority of participants made 3 to 5 edit events between two plays across days
- The low number of edits between plays suggests that participants were following test-based tinkering behavior



Summary of Key Findings

- CT concepts:
  - Participants showed success in defining chords/lists, using loops
  - We expected more participants to meet the requirements and recommendations during the after-class activities
  - Maybe due to the online nature of camp or creative nature of music
- CT practices:
  - Broad range of construction-based tinkering, which did not relate to gender, prior music experience or prior interests in computing
  - A major goal of our work is to broaden participation in computing, this finding is encouraging
  - All participants demonstrated some test-based tinkering with few edits between plays

Threats to Validity

- Small number of participants who actually edited code during the after-class activities
  - Mainly due to the nature of the camp: online, 1-hour, no supervision
  - Will increase that number in future in-person formats to gather more data

- Did not collect qualitative data focusing on students' thinking processes while coding
  - Coding logs may not reflect the real thinking
  - Could be mitigated using think-aloud methods

Conclusions and Future Work

- One of the first to use process logs in music coding to explore CT practices in an integrated approach to CT learning
  - Broad range of construction-based tinkering
  - Small difference in test-based tinkering with few edits between plays
- CT concepts demonstrated in student coding products
  - success in defining chords/lists, using loops
- Future work includes collecting a greater volume of process log data that may help uncover additional patterns; collecting qualitative data

- We appreciate TunePad develop team for data logging
- This paper is based upon work supported by the National Science Foundation under grants 2048793 and 2048792

Reference

- Sam Aaron. 2016. Sonic Pi–performance in education, technology and art. International Journal of Performance Arts and Digital Media 12, 2 (2016), 171–178.
- Yasemin Allsop. 2019. Assessing computational thinking process using a multiple evaluation approach. International journal of child-computer interaction 19 (2019), 30–55.
- Adriano Baratè, Andrea Formica, Luca A Ludovico, and Dario Malchiodi. 2017. Fostering computational thinking in secondary school through music-an educational experience based on google blockly. In Proceedings of the 9th International Conference on Computer Supported Education - Volume 2: CSEDU,, Vol. 2. INSTICC, SciTePress, 117–124. https://doi.org/10.5220/0006313001170124
- Judith Bell and Tim Bell. 2018. Integrating computational thinking with a music education context. Informatics in Education 17, 2 (2018), 151–166.
- Karen Brennan and Mitchel Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada, Vol. 1. 25.
- Pam Burnard, Zsolt Lavicza, Carrie Anne Philbin, et al. 2016. Strictly coding: Connecting mathematics and music through digital making. Proceedings of Bridges 2016: Mathematics, Music, Art, Architecture, Education, Culture (2016), 345–350.
- Yihuan Dong, Samiha Marwan, Veronica Catete, Thomas Price, and Tiffany Barnes. 2019. Defining tinkering behavior in open-ended block-based programming assignments. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education. 1204–1210.
- Shelly Engelman, Brian Magerko, Tom McKlin, Morgan Miller, Doug Edwards, and Jason Freeman. 2017. Creativity in authentic STEAM education with EarSketch. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. 183–188.
- Jason Freeman, Brian Magerko, Doug Edwards, Tom Mcklin, Taneisha Lee, and Roxanne Moore. 2019. EarSketch: engaging broad populations in computing through music. Commun. ACM 62, 9 (2019), 78–85.
- Gena R. Greher and Jesse M. Heines. 2014. Computational Thinking in Sound: Teaching the Art and Science of Music and Technology. Oxford University Press, Inc., USA.
- Mark Guzdial. 2003. A Media Computation Course for Non-Majors. In Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education (Thessaloniki, Greece) (ITiCSE '03). Association for Computing Machinery, New York, NY, USA, 104–108. https://doi.org/10.1145/961511.961542
- Michael Horn, Amartya Banerjee, Melanie West, Nichole Pinkard, Amy Pratt, Jason Freeman, Brian Magerko, and Tom McKlin. 2020. TunePad: Engaging learners at the intersection of music and code. (2020).
- Rachel Horst, Kedrick James, Yuya Takeda, and William Rowluck. 2020. From play to creative extrapolation: Fostering emergent computational thinking in the makerspace. Journal of Strategic Innovation and Sustainability 15, 5 (2020), 40–54.
- Fatemeh Jamshidi and Daniela Marghitu. 2019. Using music to foster engagement in introductory computing courses. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education. 1278–1278.
- Christian Köppe. 2020. Program a Hit -- Using Music as Motivator for Introducing Programming Concepts. In Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '20). Association for Computing Machinery, New York, NY, USA, 266–272. https://doi.org/10.1145/3341525.3387377
- Douglas Lusa Krug, Edtwuan Bowman, Taylor Barnett, Lori Pollock, and David Shepherd. 2021. Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (Virtual Event, USA) (SIGCSE '21). Association for Computing Machinery, New York, NY, USA, 397–403. https://doi.org/10.1145/3408877.3432424
- Brian Magerko, Jason Freeman, Tom Mcklin, Mike Reilly, Elise Livingston, Scott Mccoid, and Andrea Crews-Brown. 2016. Earsketch: A steam-based approach for underrepresented populations in high school computer science education. ACM Transactions on Computing Education (TOCE) 16, 4 (2016), 1–25.
- Tom McKlin, DanaWanzer, Taneisha Lee, Brian Magerko, Doug Edwards, Sabrina Grossman, and Jason Freeman. 2019. Implementing EarSketch: Connecting classroom implementation to student outcomes. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education. 634–640.
- Christopher Petrie. 2021. Interdisciplinary computational thinking with music and programming: a case study on algorithmic music composition with Sonic Pi. Computer Science Education (2021), 1–23.
- Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: Programming for All. Commun. ACM 52, 11 (nov 2009), 60–67. https://doi.org/10.1145/1592761.1592779
- Daniele Traversaro, Giovanna Guerrini, and Giorgio Delzanno. 2020. Sonic Pi for TBL Teaching Units in an Introductory Programming Course. In Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization. 143–150.
- LeChen Zhang and Jalal Nouri. 2019. A systematic review of learning computational thinking through Scratch in K-9. Computers & Education 141 (2019), 103607.

Q&A