

EarSketch: An Authentic, STEAM-Based Approach to Computing Education

Dr. Roxanne Moore, Georgia Institute of Technology

Roxanne Moore is currently a Research Engineer at Georgia Tech with appointments in the school of Mechanical Engineering and the Center for Education Integrating Mathematics, Science, and Computing (CEISMC). She is involved with engineering education innovations from K-12 up to the collegiate level. She received her Ph.D. in Mechanical Engineering from Georgia Tech in 2012.

Mr. Douglas Edwards, Georgia Institute of Technology

Douglas Edwards is a K-12 Science Technology Engineering Mathematics (STEM) educational researcher with the Georgia Institute of Technology. His educational experience in the Atlanta area for the past twenty years includes high school mathematics teaching, Math/Science Magnet Program Director, Title I educational data specialist, and Associate Professor of Information Technology. As a former US Air Force electronics engineer, Doug was also an engineering project manager.

Jason Freeman, Georgia Institute of Technology

Jason Freeman is an Associate Professor of Music at Georgia Tech. His artistic practice and scholarly research focus on using technology to engage diverse audiences in collaborative, experimental, and accessible musical experiences. He also develops educational interventions in K-12, university, and MOOC environments that broaden and increase engagement in STEM disciplines through authentic integrations of music and computing. His music has been performed at Carnegie Hall, exhibited at ACM SIGGRAPH, published by Universal Edition, broadcast on public radio's Performance Today, and commissioned through support from the National Endowment for the Arts. Freeman's wide-ranging work has attracted support from sources such as the National Science Foundation, Google, and the Aaron Copland Fund for Music. He has published his research in leading conferences and journals such as Computer Music Journal, Organised Sound, NIME, and ACM SIGCSE. Freeman received his B.A. in music from Yale University and his M.A. and D.M.A. in composition from Columbia University.

Dr. Brian Magerko, Georgia Institute of Technology

Dr. Magerko is an Associate Professor of Digital Media & head of the Adaptive Digital Media (ADAM) Lab at Georgia Tech. He received his B.S. in Cognitive Science from Carnegie Mellon (1999) and his MS and Ph.D. in Computer Science and Engineering from the University of Michigan (2001, 2006). His research explores the intersection of creativity, cognition, and computing. This interdisciplinary work leads to studying creativity and human cognition, building artificial intelligence systems that can creatively collaborate with human users, and exploring the use of human creativity as a gateway to better understanding how to effectively teach computing skills. Much of this work results in cutting edge digital media experiences in digital games, interactive narrative, and educational media.

Dr. Magerko has been research lead on over \$5 million of federally-funded research; has authored over 60 peer reviewed articles related to cognition, creativity, and computation; has had his work shown at galleries and museums internationally; and co-founded a learning environment for computer science - called EarSketch - that has been used by tens of thousands of learners worldwide.

Tom McKlin, SageFox Consulting Group

Dr. Anna Xambo, Georgia Institute of Technology

Anna Xambó is a postdoctoral fellow at Center for Music Technology and Digital Media Program (Georgia Tech) for the EarSketch project. She completed her PhD in computer-supported collaboration on interactive tabletops for music performance at The Open University (UK). Her background is in information and media technologies (MSc Universitat Pompeu Fabra), and anthropology (BA, MA Universitat de Barcelona), specializing in HCI and Music Technology. Her research interests include ways of exploiting music technology and computer music concepts using physical computing and computational tools



that can ultimately inform STEAM education, HCI research and Sound and Music Computing (SMC) research. As a composer, performer and producer of experimental electronic music, she performs under different aliases (peterMann, pulso). Her works are usually published through the label Carpal Tunnel.

Personal website: annaxambo.me

EarSketch: An Authentic, STEAM-based Approach to Computing Education

Abstract

Demand for computer scientists is robust, but the pipeline for producing them is not. US universities are only meeting about a third of demand for computer scientists, and recruiting a diverse student body is a struggle; the number of women in computer science has actually declined in the past decade. To help change the perception of the computing field, researchers at Georgia Institute of Technology developed EarSketch. EarSketch is an authentic STEAM (STEM + Arts) environment for teaching and learning programming (i.e. where learners are engaged in authentic practices both in computing and in the aesthetics of music remixing) aimed at increasing and broadening participation in computing. In the EarSketch environment, students write code to manipulate, or remix, musical samples. It is an integrated programming environment, digital audio workstation, curriculum, and audio loop library.

EarSketch has already been piloted in multiple classroom environments, including Computer Science Principles (CSP) classes in Atlanta-area high schools, in summer workshops, as part of a MOOC music technology course, in undergraduate computing courses for non-majors, and in a graduate digital media course at Georgia Tech. EarSketch is unique from other STEAM projects in computing education in that it is authentic both from an artistic perspective and from a computing perspective. That is, students create music in popular, personally relevant styles and genres, like dubstep and techno, and also learn to code in an industry-relevant language, like Python or JavaScript, in a free, browser-based environment. In addition, the barriers to entry are kept low; no previous knowledge of music performance or composition is required to engage successfully with EarSketch.

In this paper, we present a description of the EarSketch environment and curriculum. We also present an overview of the classroom environments in which EarSketch has been implemented to date, including professional development feedback, student artifacts, student engagement data, and student achievement. The authors believe that EarSketch has the potential to serve as an introductory programming unit for a variety of courses in both pre-college and college settings. Based on initial data, EarSketch is an effective method for teaching programming of musical content and is effective in improving motivation to succeed on computing problems.

Introduction

In disciplines ranging from science to the humanities to the arts, computational thinking has become central to how we create, communicate, experiment, evaluate, iterate, and innovate [1]. Computer science education research has generally focused on engagement [2], [3] compelling domains for learning computing [4]–[9], and specific pedagogical strategies [10], [11]. However, as a nation we have yet to make major strides in developing computing literacy as a core 21st century STEM skill for tackling multidisciplinary problems [12]. Computing is seen as uncool and approaches to teaching it may be uninspiring [13]. Yet demand for computer scientists in the job market remains robust: over 1.4 million computing-related jobs are estimated to be available nationally in 2018, with US universities only meeting a third of that demand in degree production [12].

Music, a ubiquitous part of human culture with directly observable neurological foundations in the human brain [14], [15], may serve as a more ubiquitous approach to engaging prospective students in computing than other domains. For example, recent research has suggested that for African-Americans, music is such an activity [16], [17]. A recent study of the occupational interests of 22,000 teenagers, for instance, found that African-American males were about half as likely to be interested in computer programming careers than white males but a third more likely to be interested in musical careers than white males. The same trend was found in other underrepresented groups as well; girls, for instance, were more than 40% more interested in musical careers than in computer programming [18].

To change the perception of computing, CS Principles (CSP) is emerging as a new standard for Advanced Placement (AP) and other high school CS courses. CSP takes a broader view of computing literacy, focusing not only on algorithms, data structures, and programming, but also on the social, cultural, and technological impacts of computing. The course, slated for official launch in 2016, has already had success at motivating and engaging students in pilot deployments nationwide [19], [20].

EarSketch will promote so-called 21st Century Skills such as creativity, collaboration, and communication in any computing course, but it is specifically tailored to support CS Principles (CSP). At the core of the curriculum are 35 learning objectives organized under 7 “big ideas” and 6 computational thinking practices [21]. The AP assessment includes both a machine graded exam and a portfolio of human-graded student “performance tasks” [22] completed throughout the course. For one of these tasks, students write programs of their choosing either individually or in pairs, and submit extensive descriptions of the programs, their intent, their design, and the process by which they were created. Students and teachers may choose the programming environments (such as EarSketch) to use in the course.

Background and Relevant Literature

STEAM aims to integrate design and the arts into STEM education and research in order to instill greater creativity, innovation, and cultural relevance into the global STEM workforce of the 21st century [23]–[25]. STEAM projects in computing education have included physical computing and digital fabrication [7], [26], music [27]–[29], games, storytelling, and animation [6], [30]. These projects generally aim to motivate students to study computing topics by contextualizing them in the arts, and their success depends on the personal motivation of students in that particular artistic domain. For example, the integration of an artistic domain of little interest to most students (e.g. square dancing) may have little positive effect or even negative repercussions, while a domain with near-universal appeal across ethnicities, socio-economic backgrounds, and gender — such as music— holds greater promise. To further support broad and effective adoption, we contend that the integrated artistic practices must have low barriers of entry, thereby enabling personal, creative expression by students with no prior domain-specific experience (i.e. a musical domain must be accessible to students who are not musicians).

We contend that a STEAM approach is most effective when both the technical and artistic elements of the learning experience are authentic. For example, in a computational music context, students create music in popular, personally relevant styles and genres (rather than the folk songs, classical music, and pure waveforms predominant in other approaches); learn to code in an industry-relevant language (like Python or JavaScript); use music production paradigms common in the recording industry (like digital audio workstation workflows); and are assessed in part through projects in which they write code to creatively remix music. EarSketch provides exactly such an authentic STEAM context for introductory computing education.

The coupling of music and computer science education is well established; there is indeed a history both of using computers as a means to teach music [31] and of using music as a means to teach programming, dating back to children writing music programs with the LOGO programming language [32] and *Tuneblocks* [27]. Recent projects have inserted digital music into introductory computer science curriculum by using music libraries in conjunction with a traditional programming language — e.g. learning about concepts such as array manipulation through producing sound rather than doing linear searches of text [2], coupling Java and a MIDI library so that students can code layered sounds to produce melodies [33], using the *myro.chuck* Python module to control music synthesis [34], and algorithmically generating MIDI messages through Python [35].

The above prior educational approaches tend to approach music from the level of either signal processing and synthesis (i.e. manipulating or generating buffers of amplitude samples to transform or create sound) or MIDI (i.e. manipulating or generating MIDI messages that represent individual musical notes in a song). With the former approach, beginning students are often limited to creating abstract sounds (i.e. sine waves) or simple transformations (i.e. volume

adjustments) that seem disconnected from real music and music technology applications. With the latter approach, students must have a thorough understanding of musical pitch, rhythm, and notation, and even with that background in hand, are often limited to creating simple nursery rhyme or folk song melodies rather than the multi-layered textures common to contemporary popular music. EarSketch approaches music from the level of the digital audio workstation and the audio clip, incorporating music production and remixing in popular styles such as hip hop, dubstep, pop, techno, and electronica to provide a powerful, culturally situated motivational context that requires no prior musical training.

EarSketch Platform

The EarSketch platform includes a programming environment, digital audio workstation, curriculum, audio loop library, and social sharing service that enables students to write Python or JavaScript code to create and share musical remixes. EarSketch students write code to creatively manipulate musical samples and effects while learning computing fundamentals such as loops, lists, and functions. EarSketch is free to use and can be accessed online at ears sketch.gatech.edu.

EarSketch Curriculum

The EarSketch curriculum for CSP high school students is an authentic, problem-based curriculum that consists of two units. Our study currently works with a set of three high schools that are diverse in ethnicity as well as socioeconomic status, as shown in Table 1 below. We chose a problem based learning framework, because it has been shown to provide relevance that is more engaging for students in general and for underrepresented youth in particular [36], [37]. The first problem, called the Client Creativity Challenge, requires the student to design a thirty to forty second musical introduction for their client. The client can be an organization of the student’s choice with instructor approval or IBNX radio. IBNX is an Atlanta based internet radio station that has agreed to play the students’ artifacts. The second project, called the Collaboration Challenge, requires a student pair to design a two to three minute song as a video background or for entry in to the Brave New Voices Mixtape competition.

Table 1: Demographic Information

| Study High School Demographics | | | | | | |
|--------------------------------|---------------------|-----------|-------|----------|-------|-------------|
| High School | %Free/Reduced Lunch | Caucasian | Black | Hispanic | Asian | Multiracial |
| School 1* | 78 | 10 | 41 | 42 | 4 | 3 |
| School 2 | 81 | 6 | 35 | 47 | 9 | 3 |
| School 3 | 41 | 38 | 25 | 21 | 12 | 4 |

*As reported by the school's assistant principal, because the school opened in fall 2015

The curriculum is aligned to the programming standards of the Georgia Computer Science Principles (CSP) course. This course’s standards are similar to the Learning Objectives of the Advanced Placement (AP) CSP course that will launch in the fall of 2016. However, the Georgia CSP does not have further specific requirements, as does AP CSP. As an example the Georgia

CSP standard on abstraction, “Use abstraction to manage complexity in programs”, is the same as the AP CSP Learning Objective. Yet, in addition, the AP CSP course has fifteen specific Essential Knowledge requirements, under the abstraction learning objective, while the specifics of the Georgia course standard are left to the discretion of the teacher. Table 2 below shows the alignment of the course to the Georgia CSP standards, AP CSP learning objectives, problem-based projects, and student understandings for each lesson. As shown in the table the student understandings scaffold based on project relevance, showing the learner the value of a coding topic in EarSketch through a musical purpose. As an example, string data types and looping in EarSketch using the makeBeat function is of value for musical creativity and musical repetition.

Table 2: Standards and Objectives Alignment for EarSketch to CSP

| Unit 1 | Introduction to EarSketch | | |
|--------|---|---|--|
| Lesson | Georgia Computer Science Principles (CSP) Standard | Related AP Computer Science Principles (CSP) Learning Objective | Student Understandings |
| 0 | 2.4 Develop a program for creative expression or to satisfy personal curiosity which may have visual, audible, or tactile results | 5.1.1 Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. | Understand Client Creativity Challenge, Charter, & Rubric and identify purpose and navigation of EarSketch Interface |
| 1 | 6.1 Explain how programs implement algorithms. | 5.2.1 Explain how programs implement algorithms. | Apply core EarSketch function fitMedia to code, run, and save an introductory script. Recognize data types, functions and parameters through using fitMedia. |
| 2 | 6.4 Locate and correct errors in a program. | 5.4.1 Evaluate the correctness of a program. | Understand need for documentation and apply debugging procedures. |
| 3 | 6.1 Explain how programs implement algorithms. | 5.2.1 Explain how programs implement algorithms. | Apply core EarSketch function setEffect |
| 4 | 6.1 Explain how programs implement algorithms. | 5.2.1 Explain how programs implement algorithms. | Apply core EarSketch function makeBeat |
| 5 | 8.5 Connect computing within economic, social, and cultural contexts. | 7.4.1 Explain the connections between computing and social, economic, and cultural contexts. | Understand how music tempo, pitch, repetition, and contrast affect mood. Apply understanding to design first draft of algorithm for the challenge. |
| 6 | 6.1 Explain how programs implement algorithms. | 5.2.1 Explain how programs implement algorithms. | Apply for loops to create musical repetition and update algorithm. |
| 7 | 6.1 Explain how programs implement algorithms. | 5.2.1 Explain how programs implement algorithms. | Apply conditionals to create musical contrast and update algorithm. |
| 7 | 8.5 Connect computing within economic, social, and cultural contexts. | 7.3.1 Analyze the beneficial and harmful effects of computing. | Justify whether to license challenge production based on the interaction of intellectual property law and economics. |
| 8 | 6.3 Evaluate a program for correctness i.e. program style affecting the determination of | 5.4.1 Evaluate the correctness of a program. | Self evaluate Challenge Production in respect to rubric . |
| Unit 2 | Advanced EarSketch | | |
| Lesson | Georgia Computer Science Principles (CSP) Learning Standard | Related AP Computer Science Principles (CSP) Learning Objective | Student Understandings |
| 0 | 6.7 Collaborate to solve a problem using programming. | 5.1.3 Collaborate to develop a program. | Understand Collaboration Challenge, Charter, and Rubric. Identify how to upload Freesound samples and record and upload user defined sounds. |
| 1 | 6.7 Collaborate to solve a problem using programming. | 5.1.3 Collaborate to develop a program. | Recognize musical form, Collaborative management principles, Music mixing and mastering roles, and develop a collaboration plan with clear roles, and internal progress reviews. |
| 2 | 6.2 Use abstraction to manage complexity within programs. | 5.3.1 Use abstraction to manage complexity within programs. | Apply user defined functions to musical form to manage program complexity. |
| 3 | 6.2 Use abstraction to manage complexity within programs. | 5.3.1 Use abstraction to manage complexity within programs. | Apply lists as abstract data type to contain favorite music samples. |
| 4 | 6.2 Use abstraction to manage complexity within programs. | 5.5.1 Employ appropriate mathematical and logical concepts in programming. | Apply random number generator to list to design musical transition. |
| 5 | 6.2 Use abstraction to manage complexity within programs. | 5.5.1 Employ appropriate mathematical and logical concepts in programming. | Apply random number generator to string operations to design randomized musical contrast. |
| 6 | 6.2 Use abstraction to manage complexity within programs. | 5.5.1 Employ appropriate mathematical and logical concepts in programming. | Apply envelopes in setEffects to add musical amplitude adjustments. |
| 7 | 8.4 Analyze the beneficial and harmful effects of computing. | 7.3.1 Analyze the beneficial and harmful effects of computing. | Justify a collaborative decision on how to protect the intellectual property of their challenge production. |
| 8 | 6.5 Justify and explain a program's correctness. | 5.4.1 Evaluate the correctness of a program. | Self evaluate challenge production in respect to rubric. |

Professional Learning:

Seven teachers received twenty hours of professional learning (PL) in June of 2015 to instruct the curriculum during the 2015-2016 school year. The professional learning was designed with two parts, content of the student curriculum and pedagogy [38], [39]. Part one was on how to use the EarSketch environment with Python, going over the student curriculum at a fast pace. Part two was on the pedagogy behind the curriculum, so that teachers would understand the rationale and sequence behind the problem-based curriculum. District personnel had informed us that all teachers would have a good computer science (CS) background. At the beginning of the PL,

based on self-reporting, we found that two of the teachers did have a strong background and the others did not. Two of the other teachers at the PL had taught web design and did not have a strong CS background, while the seventh teacher had a CS degree, but had never taught CS. Hence, four of the seven teachers did not have a CS background. This required us to spend most of our PL time on coding in EarSketch using Python, and very little time on pedagogy. The teachers rated the PL in content, context, and process on a 5 point Likert scale, where 1 signifies Strongly Disagree and 5 signifies Strongly Agree. All ratings were above 4 indicating that the teachers positively received the PL. The highest rating was a 5.0 for the statement “The facilitators helped me understand how to implement my learning”. The lowest ratings of 4.20 were about the facilities and the statement “Sufficient time was provided for guided practice and tasks”.

Implementation and Initial Results

EarSketch has been specifically targeted toward CSP and AP CSP, but is also applicable toward other K-20 curricula. Some descriptions of courses in which EarSketch was implemented and brief discussions of results, where available, are provided below.

CS Principles piloting: The EarSketch team collected data on the high school student CSP curriculum using EarSketch taught by three of the teachers that attended the PL. Two of the three study teachers were the teachers that self-reported that they did not have a good CS background. Based on bi-weekly observations of the teachers over a ten-week period, all three of the teachers implemented the problem-based contexts. However, only one of the teachers consistently used the lesson scaffolding designed by the EarSketch team.

Based on this 2015-2016 implementation in three high schools, results reveal that students’ attitudes positively and statistically significantly increased across all constructs in our Student Engagement survey, which included constructs such as computing confidence, computing enjoyment, computing perceived usefulness, motivation to succeed in computing, identity and belonging in computing, and intention to persist. In addition, students showed statistically significant gains from pre to post testing on a content knowledge assessment targeting computer programming (not specific to EarSketch). More results will be included in future publications. Claims about EarSketch and its effectiveness in various capacities will be strengthened as more high schools are added to the study over the next two years.

MOOC: EarSketch has been offered as part of a massive open online course (MOOC) on Coursera by one of the authors (Freeman) (<https://www.coursera.org/learn/music-technology/>). The course, based on a popular undergraduate survey course in music technology at Georgia Tech, teaches basic music technology theory and history along with traditional music production techniques in digital audio workstation software, then focuses on algorithmic composition and Python coding with EarSketch for the second half of the course. Since 2013, the course has been

offered twice in a cohort-based format and is currently available on-demand. Over 43,000 learners have enrolled.

Graduate Media Class: EarSketch was employed in fall 2014 and fall 2015 at Georgia Tech in a digital media graduate course called The Computer as an Expressive Medium. The class is used as an introduction to programming for digital media students with limited computing background, with a focus on using programming to express concepts, emotions, and critical thought. EarSketch was used as the initial foray into expressive programming, using music remixing as the main focus. The syllabus subsequently moves on to generative poetry and interactive installation.

Undergraduate curriculum for non-majors: We are developing a new EarSketch curriculum geared towards introductory computing courses for non-majors at the college level, and we want to see if we can replicate the results we have seen at the high school level in this different context, and see what different issues with deployment/curriculum/engagement come up in this context. Our pilot study at Georgia Gwinnett College began in January 2016.

Conclusions & Future Work

Among current computing offerings, EarSketch is unique in its authentic STEAM approach, using the near-universal appeal of music [15], and the extremely low barrier to music creation afforded by the digital audio workstation paradigm to create a computationally and musically authentic experience requiring no prior music or CS knowledge. The current EarSketch student curriculum, organized as a series of thematic modules, focuses on integrated learning of core CS concepts, Python or JavaScript programming constructs, EarSketch API functions, and music and music technology concepts. For example, students learn the benefits of abstraction and the syntax for defining and calling their own functions at the same time they learn about repetition and variation in popular musical forms; they then define musical sections as functions and use function parameters to introduce musical variations when sections repeat.

EarSketch's authenticity in the computing domain depends on students recognizing that the programming language they learn through it (Python or JavaScript) has broad industry relevance beyond creating music.

Our goal is to continue to integrate EarSketch with CSP and test it in a variety of school settings. Both EarSketch and CSP seek authentic high school student engagement, computing literacy, and broadened participation. EarSketch directly supports the learning objectives and computational thinking practices at the core of CSP, and the CSP framework enables students and teachers to use any programming environment, including EarSketch's.

Beyond CSP and AP CSP, EarSketch has shown promising results to MOOC audiences, graduate audiences, and undergraduate audiences. One of the next audiences to pursue is the middle school and late elementary school audience, a project which has recently been funded by the Arthur M. Blank Family Foundation. Through this project, a block programming version of EarSketch will be developed along with new lessons and materials in support of this new version of EarSketch for piloting in Atlanta- area elementary and middle schools during summer camps.

Acknowledgments

EarSketch receives funding from the National Science Foundation (CNS #1138469, DRL #1417835, and DUE #1504293), the Scott Hudgens Family Foundation, and the Arthur M. Blank Family Foundation.

References

- [1] J. M. Wing, “Computational thinking and thinking about computing,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1881, pp. 3717–3725, 2008.
- [2] M. Guzdial and E. Soloway, “Teaching the Nintendo generation to program,” *Commun. ACM*, vol. 45, no. 4, pp. 17–21, Apr. 2002.
- [3] A. Bruckman, M. Biggers, B. Ericson, T. McKlin, J. Dimond, B. DiSalvo, M. Hewner, L. Ni, and S. Yardi, ““Georgia computes!”: improving the computing education pipeline,” in *Proceedings of the 40th ACM technical symposium on Computer science education*, Chattanooga, TN, USA, 2009, pp. 86–90.
- [4] M. Conway, S. Audia, T. Burnette, D. Cosgrove, and K. Christiansen, “Alice: lessons learned from building a 3D system for novices,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, The Hague, The Netherlands, 2000, pp. 486–493.
- [5] D. J. Malan and H. H. Leitner, “Scratch for budding computer scientists,” in *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, Covington, Kentucky, USA, 2007, pp. 223–227.
- [6] C. Kelleher, R. Pausch, and S. Kiesler, “Storytelling alice motivates middle school girls to learn computer programming,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, San Jose, California, USA, 2007, pp. 1455–1464.
- [7] L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett, “The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2008, pp. 423–432.
- [8] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, and B. Silverman, “Scratch: programming for all,” *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009.
- [9] B. J. DiSalvo, K. Crowley, and R. Norwood, “Learning in Context: Digital Games and Young Black Men,” *Games and Culture*, vol. 3, no. 2, pp. 131–141, Apr. 2008.
- [10] L. Williams, E. Wiebe, K. Yang, M. Ferzli, and C. Miller, “In Support of Pair Programming in the Introductory Computer Science Course,” *Computer Science Education*, vol. 12, no. 3, pp. 197–212, 2002.
- [11] C. D. Hundhausen, N. H. Narayanan, and M. E. Crosby, “Exploring studio-based instructional models for computing education,” in *SIGCSE*, 2008, vol. 8, pp. 392–396.

- [12] J. Cuny, "Address to the Computer Science Community," 2012. [Online]. Available: http://www.ncwit.org/sites/default/files/legacy/pdf/CS10K_Cuny.pdf. [Accessed: 13-Apr-2015].
- [13] Q. H. Mahmoud, "Revitalizing Computing Science Education.," *IEEE Computer*, vol. 38, no. 5, pp. 98–100, 2005.
- [14] A. J. Blood and R. J. Zatorre, "Intensely pleasurable responses to music correlate with activity in brain regions implicated in reward and emotion," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 20, p. 11818, 2001.
- [15] I. Peretz and R. J. Zatorre, "Brain organization for music processing," *Annual Review of Psychology*, vol. 56, pp. 89–114, 2005.
- [16] P. A. Noguera, "The Trouble with Black Boys: The Role and Influence of Environmental and Cultural Factors on the Academic Performance of African American Males," *Urban Education*, vol. 38, no. 4, pp. 431–459, Jul. 2003.
- [17] F. A. Lee, R. K. Lewis, J. R. Sly, C. Carmack, S. R. Roberts, and P. Basore, "Promoting Positive Youth Development by Examining the Career and Educational Aspirations of African American Males: Implications for Designing Educational Programs," *Journal of Prevention & Intervention in the Community*, vol. 39, no. 4, pp. 299–309, 2011.
- [18] K. A. S. Howard, A. H. Carlstrom, A. D. Katz, A. Y. Chew, G. C. Ray, L. Laine, and D. Caulum, "Career aspirations of youth: Untangling race/ethnicity, SES, and gender," *Journal of Vocational Behavior*, vol. 79, no. 1, pp. 98–109, Aug. 2011.
- [19] O. Astrachan and A. Briggs, "The CS Principles Project," *ACM Inroads*, vol. 3, no. 2, pp. 38–42, Jun. 2012.
- [20] O. Astrachan, T. Barnes, D. D. Garcia, J. Paul, B. Simon, and L. Snyder, "CS principles: piloting a new course at national scale," in *Proceedings of the 42nd ACM technical symposium on Computer science education*, New York, NY, USA, 2011, pp. 397–398.
- [21] The College Board, "AP Computer Science Principles: Draft Curriculum Framework," Aug-2013. [Online]. Available: <http://media.collegeboard.com/digitalServices/pdf/ap/2013-0607-comp-sci-principles-cf-final.pdf>.
- [22] The College Board, "Computer Science Principles: Performance Assessment," 2013. [Online]. Available: <http://media.collegeboard.com/digitalServices/pdf/ap/CSP-PB-tasks.pdf>.
- [23] B. Magerko, J. Freeman, T. McKlin, S. McCoid, T. Jenkins, and E. Livingston, "Tackling Engagement in Computing with Computational Music Remixing," in *Proceedings of the Special Interest Group on Computer Science Education*, Denver, CO, 2013.
- [24] K. Peppler, R. Santo, M. Gresalfi, and K. Salen, *Script Changers: Digital Storytelling with Scratch*. Cambridge, MA: MIT Press, 2014.
- [25] K. Peppler, K. Salen, M. Gresalfi, and R. Santo, *Short Circuits: Crafting with DIY Electronics*. Cambridge, MA: MIT Press, 2014.
- [26] W. W. Y. Lau, G. Ngai, S. C. F. Chan, and J. C. Y. Cheung, "Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students," *SIGCSE Bull.*, vol. 41, no. 1, pp. 504–508, Mar. 2009.
- [27] J. Bamberger, *The Mind Behind the Musical Ear*. Harvard University Press, 1995.
- [28] G. Wang, D. Trueman, S. Smallwood, and P. R. Cook, "The Laptop Orchestra as Classroom," *Computer Music Journal*, vol. 32, no. 1, pp. 26–37, Mar. 2008.
- [29] A. Ruthmann, J. M. Heines, G. R. Greher, P. Laidler, and C. Saulters II, "Teaching computational thinking through musical live coding in scratch," in *Proceedings of the 41st ACM technical symposium on Computer science education*, 2010, pp. 351–355.
- [30] A. Monroy-Hernández and M. Resnick, "Empowering kids to create and share programmable media," *interactions*, vol. 15, no. 2, pp. 50–53, 2008.
- [31] M. Brandao, G. Wiggins, and H. Pain, "Computers in music education," in *Proceedings of the AISB '99 Symposium on Musical Creativity*, 1999.

- [32] C. Kelleher and R. Pausch, "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers," *ACM Comput. Surv.*, vol. 37, no. 2, pp. 83–137, 2005.
- [33] M. P. Rogers, "Making music in CS I," *J. Comput. Small Coll.*, vol. 20, no. 1, pp. 98–105, 2004.
- [34] A. Misra, D. Blank, and D. Kumar, "A music context for teaching introductory computing," in *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education*, Paris, France, 2009, pp. 248–252.
- [35] B. Manaris and A. R. Brown, *Making Music with Computers: Creative Programming in Python*, vol. 13. Boca Raton, FL: CRC Press, 2014.
- [36] J. Strobel and A. van Barneveld, "When is PBL More Effective? A Meta-synthesis of Meta-analyses Comparing PBL to Conventional Classrooms," *Interdisciplinary Journal of Problem-Based Learning*, vol. 3, no. 1, 2009.
- [37] L. Darling-Hammond, S. Goldman, and M. Zielezinski, "Using Technology to Support At-Risk Students Learning." 2014.
- [38] A. Berry and J. Van Driel, "Teacher Professional Development Focusing on Pedagogical Content Knowledge," *Educational Researcher*, 2011.
- [39] M. S. Garet, B. F. Birman, A. C. Porter, L. Desimone, and J. Herman, "Designing effective professional development: Lessons from the Eisenhower program [and] technical appendices." US: District of Columbia, 1999.