Making Noise: Using Sound-Art to Explore Technological Fluency

Erik Brunvand
School of Computing
University of Utah
elb@cs.utah.edu

Nina McCurdy
School of Computing
University of Utah
nina@cs.utah.edu

ABSTRACT

We describe our experience designing and delivering a general education technological fluency course that frames the discussion of computer science and engineering technology (electronics and programming) in the context of sound-art: art that uses sound as its medium. This course is aimed at undergraduate students from a wide variety of backgrounds and is designed to fit into the "Intellectual Explorations" area of a general undergraduate program. The goal is to introduce computer engineering and computational principles to non-CS students through an exploration of sound-art, experimental and electronic music, noise-making circuits, hardware hacking, and circuit bending.

1. INTRODUCTION

In 1997 the National Science Foundation (NSF) asked the National Research Council (NRC) to study the question of what Americans should know about information technology. The NRC released a landmark report in 1999 entitled "Being Fluent with Information Technology" [15]. An equally influential follow-on report was issued by the NRC in 2002 entitled "Technically Speaking: Why All Americans Need to Know More About Technology" [16]. These reports stress that technological *literacy* does not suffice in modern times. Literacy implies only basic knowledge of a subject. The 1999 report adopted the term *fluency* to describe "[intellectual] capabilities [to] empower people to manipulate the medium to their advantage and to handle unintended and unexpected problems when they arise" [15]. The 2002 report continues this theme, proposing to broaden technological fluency to include basic engineering knowledge, and the nature and limitations of the engineering design process [16]. This theme has echoed through the NSF resulting in calls for education to include computational thinking throughout the curriculum [4, 42]. As prices fall and access increases for "maker" materials (e.g., open-source hobbyist computing platforms like Arduino [1]), computer integration with physical devices (sensors and actuators) has emerged as a new and important facet of technological fluency.

In spite of the rise of technological tools, and the reports describing the critical nature of technology in a general educational setting, general education requirements for undergraduates are slow to change. At the University of Utah, for example, undergraduate students are required to take six classes in "intellectual exploration" of areas that are not in their major. The areas they can choose from are Fine Arts, Humanities, Social & Behavioral Science, and Applied Science. The Applied Science category, which could contain CS and engineering courses, is focused instead on experimental science in areas such as chemistry, biology, etc.

To position this course as an interesting choice for a wide variety of non-CS undergraduate students, we developed the course specifically to introduce computer science and engineering technology related to electronics and programming through the lens of experimental and electronic music and sound-art projects. Essentially, this is a way to increase the students' technological fluency but through digital media projects rather than engineering projects. It is also a way to expand students' ideas about technology in the arts and how arts and technology interact in our modern world.

One higher level goal of the project is to expand significantly the dialog on campus related to the intersection of arts and technology, and how *creative design thinking* and *engineering problem solving* are complementary skills that all students need. We hope it will also serve as a catalyst for additional cross-disciplinary collaborations both by us and by other faculty members across campus. The 15-week project-based undergraduate course is called *Making Noise: Sound Art and Digital Media.* [10]

2. BACKGROUND

There is a rich body of literature on using authentic framing contexts to encourage learning in CS and technology areas. For example, a variety of courses use media computation to introduce CS principles and techniques (e.g., [25,40]). These media computation courses typically use a wide variety of media including images, video, and sound as frameworks in which to explore computation. Other proposed classes use more specific media contexts such as art [11, 12, 43] and literature [7, 8], to name just two. There have also been proposals to use music and sound generation such as [6, 27, 32, 33, 37], and using computational techniques to manipulate music as data or in performance (e.g., [22, 26, 31, 38]). Our course inherits from these approaches a focus on audio media as a compelling context. Refining this context,

SIGCSE '17, March 08-11, 2017, Seattle, WA, USA
 ISBN 978-1-4503-4698-6/17/03...$15.00
DOI: http://dx.doi.org/10.1145/3017680.3017714

Figure 1: An inductive pickup used to make electromagnetic (EM) recordings. This is a telephone tap designed to pick up the EM field of the speaker coil on a wired telephone.

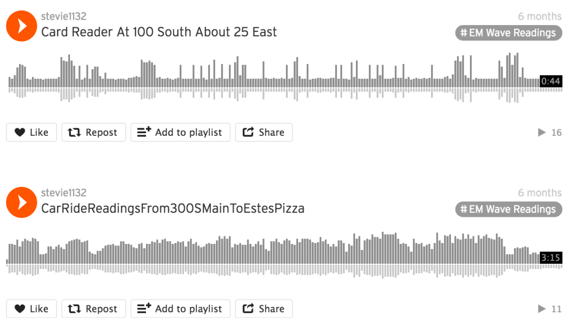

Figure 2: These field recordings using the inductive pickup were captured by a student and uploaded to SoundCloud.

we consider sound-art and experimental and electronic music as distinguishable sub-domains of audio. This allows us to delve into electronics-related computer engineering topics such as circuits and electronics in addition to pure CS coding material.

There is also a rich literature on CS courses for general education audiences including computational thinking courses, e.g., [20, 35], CS Principles courses, e.g., [2, 3], and computing as a general education topic, e.g., [19, 21, 28, 30]. Our course is targeted at technological fluency for a broad class of undergraduate students. It is specifically not designed as a lead-in course for potential CS majors. Rather it is designed as a technological component of a general education curriculum. It is also not designed to promote diversity of student demographics from a CS program perspective. The diversity we seek is students from diverse majors with interests that may be far afield from those of a prospective CS student.

3. COURSE CONTENT

Our course has been delivered twice at the University of Utah: once in Spring 2015 (19 students) and again in Spring 2016 (24 students). Students enrolled in the course have had widely varying backgrounds in terms of majors (e.g., business, fine arts, communication, chemistry, sociology, etc.). It is a project/lab-based course with a set of project assignments (Figures 1 - 6) leading up to a student-defined final project (Figures 7 - 9). Curricular material and computer science and engineering topics are detailed in Table 1.

Our framing context is sound-art: defined broadly as art that uses sound as its medium. While experimental music

```
int speakerPin = 8; // digital pin: speaker
int sensorPin = A0; // analog pin: light sensor
int Duration = 10;  // ms before re-sense
void setup() { // initial setup
  // speaker connection is an output pin
  pinMode(speakerPin, OUTPUT);
}
void loop() { // main Arduino loop
   // get a sensor reading from light sensor
   int sensorVal = analogRead(sensorPin);
   // map results from the sensor's range
   // to the desired pitch range (in Hertz):
   int freq = map(sensorVal,200,900,100,1000);
   // change the pitch, play for Duration ms:
   tone(speakerPin, freq);
   delay(Duration);
}
```

Figure 3: Simple code for Arduino using the tone library to generate sound, a CdS light sensor as input, and producing a simple "light Theremin."

certainly fits in this category, sound-art has been defined more recently in a context that sets it apart from music performance and positions it as a separate fine-art genre [34, 36]. Given that this is a relatively new art category, we also explore experimental electronic music as a precursor to sound-art [17, 29]. The electronic aspects of both areas contribute to our exploration of circuit and system aspects of computer engineering technology.

Because our sound-art context is likely to be little known by our students, we provide a set of readings, one per week, from pioneers of electronic and experimental music (e.g., Luigi Russalo, Edgar Varèse, John Cage, Iannis Xenakis, Vladimir Ussachevsky, Karlhienz Stockhausen, and Brian Eno) and sound-art (e.g., Christian Marclay, Janet Cardiff, Zimoun, Tristan Perich, Susan Philipsz, and Richard Garet). Short listening assignments, derived from [39], are also given once a week, typically focused on careful listening within the students' daily environments. Each class period (80min, twice a week) typically starts with a discussion of a reading, along with listening to works by the artist, or a discussion of a listening assignment with examples given by students. Students are issued a soft-covered sketchbook (plain or gridded) in which they can record their reading responses, listening assignment responses, and designs for their projects. Images from their sketchbook pages are turned in online for grading. The sketchbooks are used to encourage an arts context mindset among the students.

The textbook for the course is *Handmade Electronic Music: The Art of Hardware Hacking* by Nicolas Collins [14], along with instructor notes on physical computing [9]. The primary text describes in detail how to engage electronics to make noise making circuits and modify existing circuits (typically noise-making toys) to make different noises, a process known as "circuit bending" [14, 23, 24]. Circuit-related material is covered in the primary text, with Arduino programming material covered in the instructor's notes. Special equipment used in the course was purchased using a grant from our Undergraduate College, but the materials (listed in Table 2) are not terribly expensive, and could be supported using student fees (especially considering that our textbook is relatively reasonably priced by textbook standards).

Table 1: **Curricular Content of Making Noise: Sound Art and Digital Media**

Project	Activities	CS and Engineering Technical Connections
Readings	Readings and in-class listening from pioneers in electronic and experimental music and sound art.	Context for projects and labs. In-class discussions promote engagement with material and classmates.
Listening	Ear training and sound awareness assignments.	Contextualized listening, in-class discussions.
EM Field Recordings (Figs 1, 2)	Recording of EM signals using an inductive pickup tuned to the audio frequency range (phone tap). A wide variety of electronic equipment emits interesting EM noise (e.g., motors, computers) [14].	Electromagnetic signals and spectrum, information as data, data manipulation using audio editors such as Audacity [5].
Arduino Music [1] (Figs 3, 4)	Programs on Arduino that make music/noise both directly from program code and using external sensor input such as light sensors [9].	Basic imperative programming: data types, variables, conditionals, loops, arrays, etc. Physical computing with sensor inputs and output actuators (speakers).
Toy Hacking (Fig 5)	Students acquire a noise-making toy from a thrift store, use circuit-bending [14, 23] to modify its sound, and re-package project into a new context.	Basic electronic circuits: wires, resistors, capacitors, RC circuits, potentiometers, voltage division. Reverse engineering and tinkering for knowledge acquisition.
Oscillator Circuits (Fig 6)	Students build oscillator circuits using Schmidt triggers and RC circuits [14]. "Instruments" are packaged into playable self-contained projects.	Basic logic gates (inverters, NAND, and NOR), schematics, circuit wiring and construction, frequency and amplitude, modulation of signals.
Final Project (Figs 7, 8, 9)	Students conceive, design, and build sound-art projects using techniques and concepts from class. Projects are judged on both technical and aesthetic aspects.	Synthesis of technological knowledge from class. Projects typically involve a combination of computer control and sensing using Arduino along with oscillator circuits and sound samples.

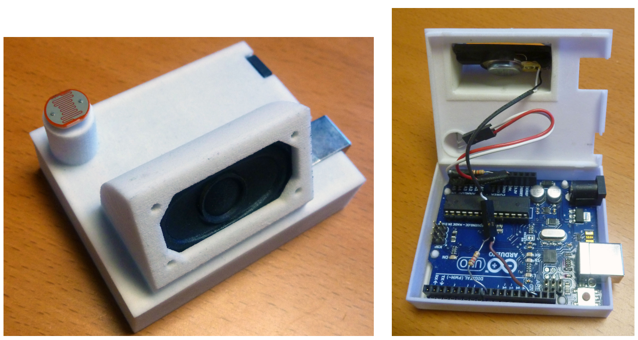

Figure 4: An example of a nicely packaged Arduino programming project: The case is 3D printed and includes a housing for the CdS light sensor and small speaker. See Figure 3 for example code.

Projects, shown in Table 1, provide contexts in which to discuss technological content, and also provide a tool box of techniques, and a library of sounds, that students can use in their final projects. Projects are documented in the students' sketchbooks and resulting sounds are uploaded to the students' SoundCloud [41] account, which is accessible to the instructors for grading (see Figure 2 for an example).

For their final project (3 weeks), students use the materials and techniques developed throughout the semester as a starting point for a project of their choice. Project ideas range from more involved hacking on toys, to electronic music compositions using the sound clips collected during the semester, to site-specific sound-art installations, to large assemblages of custom oscillator circuits, perhaps used as a live-performance instrument. The students propose their own final projects either singly or in small teams, and the final projects are presented in public demonstration at the end of the semester.

Figure 5: Noise-making toys modified by students through circuit bending / hardware hacking. The modified toys have been repackaged into new cases and new controls have been added.

4. STUDENT REACTIONS

Overall, student response was very positive to the course. Regular student evaluations are done for all courses at the University of Utah, and the score for the summary question "Overall this was an effective course" was 5.33/6.00 in Spring 2015 (compared to an average of 5.07 for other courses in the undergraduate college in that semester), and 5.56/6.00 in spring 2016 (where the college average was coincidentally also 5.07 in that semester). Representative student comments from the end-of-semester evaluations include:

[Spring 2016] "This class was absolutely the best elective I've ever taken. I think that having a basic understanding of circuits is hugely helpful in everyday life. And beyond that, this class was an incredible way to allow otherwise unartistic students to create something cool that they can be proud of. I will most likely further pursue some of the things I learned in this class as a hobby, to some degree."

[Spring 2016] "Really interesting and useful for any major, a fun applied science credit"

[Spring 2016] "This was a great learning environment and I'd take another course like this again."

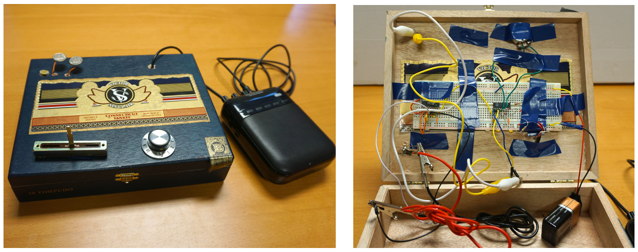

Figure 6: Oscillator circuit / noise instrument designed by a student and packaged into a cigar box. Oscillators are controlled by a knob and slider (potentiometers), light sensor, and "body contact" through the soldered coins. Oscillator sound is amplified through the amplifier/recorder.

Table 2: Partial list of specific equipment used by students in the course. Equipment can be funded using student lab fees.

Equipment	Comment
Recorder / Amplifier	5w, mp3 recording, portable guitar amplifier - $19.65 from Monoprice.com
Inductive pickup	Phone tap - $2 from surplus sources
Arduino	Ubiquitous open-source microcontroller $5 to $30 dep. on model and source
Small speakers	$0.75 - $5 each from surplus sources
Toys	Acquired by students from thrift stores. Typically around $1 - $2 each
Schmidt trigger chips	e.g., CD40106(inv) and 4093(NAND) Around $0.40/ea at surplus sources
Breadboards	Around $4 each online
Potentiometers	Both knob and slide - 500k - 5M Around $1/each from surplus sources
Cigar boxes	Great, inexpensive enclosures for projects. Sources on-line - $1-$3/ea
Basic electrical components (wire, resistors, caps)	Usually already available in on-campus labs, or easily obtained online (e.g., Mouser, Digikey, Jameco)
On-line surplus sources include: AllElectronics.com, BGMicro.com, MJPA.com, goldmine-elec-products.com	

[Spring 2015] "The way the course incorporates electrical knowledge and design as well as artistic aspects makes the course extremely effective. "

[Spring 2015] "The course was interesting, as I never really knew such topics as those covered existed. "

[Spring 2015] "What an exciting and inventive course! We need another semester on tape music!"

While we clearly cannot draw general conclusions from a small sample of students, pre- and post-surveys of students in the Spring 2016 class show a definite improvement in self-assessment of confidence in technical skills. In these surveys, of the 13 students who completed both the pre- and post-surveys, results of the question about confidence in technical fluency are shown in Table 3.

5. DISCUSSION

In this section, we discuss the successes and challenges of this course, and share insights which we hope can benefit educators embarking on similar teaching endeavors. We believe that the success of this course was due in large part to the chosen framing context. First, the interdisciplinary nature of sound-art - combining visual art, music, and electronics (three cultural cornerstones) - establishes a ground layer of familiarity, confidence, and engagement. In addition, like visual art and music, sound-art is accessible and compelling at many different levels of knowledge and expertise. This element greatly supported our diverse learning environment and, as we found, encouraged peer learning among students. Second, the radical, exploratory nature and history of experimental electronic music and sound-art, backed by a rich body of literature and multimedia, helps to create a learning environment that supports creative and objective thinking, open-ended experimentation, and hands-on learning. This environment couples nicely with the analytical thinking required to build and manipulate electronics. Third, the associated electronics are relatively friendly: electronic music and sound-art generally inhabit a low-frequency range (audio frequencies) compared to digital computer circuits, the associated circuitry is fairly straightforward, the associated high-level concepts in physics and art are relatively accessible, and the use of computing platforms such as Arduino enables a strong connection to programming skills.

Table 3: Results from pre- and post-surveys relating to self-assessment of confidence in technical skills. Students scored themselves on a scale of 1-10 (10 being high) for confidence in these technical areas.

Topic	Pre-	Post-	Diff	% increase
Electronics	5.77	7.31	1.54	26.67%
Computers	7.31	7.77	0.46	6.32%
Circuits	4.15	6.31	2.15	51.85%
Writing Code	5.00	5.62	0.62	12.31%
Modifying Code	3.62	5.31	1.69	46.81%

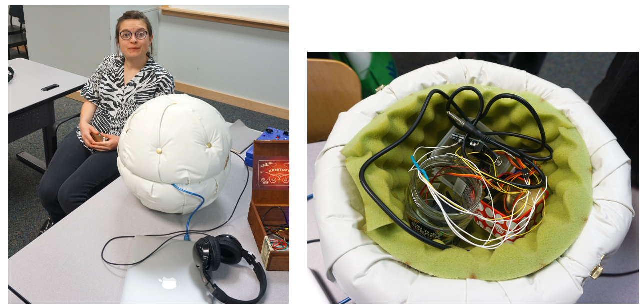

Figure 7: Final project of Madison (fine arts major) from 2016. She constructed a round padded sculpture that made electronic and physical sounds from Arduino-controlled servos, sensors, and contact microphones contained inside the sculpture.

While we deem this course an overall success, we highlight several challenges encountered throughout the course and provide some practical advice which we hope will benefit our fellow educators.

One set of challenges arose in Project four: Toy-hacking. Modifying existing circuits using Hardware-hacking is, by nature, destructive to the circuits and relies somewhat on chance. A typical first "bend" is to identify the RC timing circuit and replace the resistor with a potentiometer (knob or slider) so that the speed of the sound can be modified. Removing circuit components and replacing them is subject

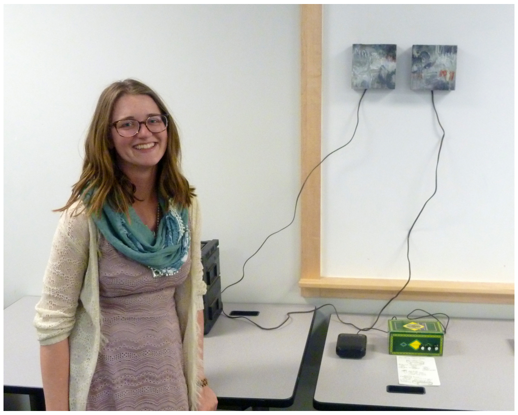

Figure 8: Final project of Kayla (fine arts major) in 2015. She painted two small paintings, installed audio drivers on the back of the paintings, and played a composition made up of sound samples she collected/generated during the semester through the paintings.

to both physical and electronic issues. This led to some frustration and disappointment within a group of relatively unlucky students whose toys did not respond well to such treatment. Modern toy circuits that use tiny surface-mount components also make toy-hacking tricky. We suspect that proper framing of the project could have helped to overcome these challenges. In particular, encouraging students to seek out older toys that likely use larger through-hole components, and to take plenty of reference photos for debugging purposes would likely improve success rates. Also helpful would be to prepare students to anticipate and not be discouraged by failures at each stage of the process, learn from such failures, and know when to move on to a new toy. It was not uncommon for students to go through three or four toys before finding one that responded to their hacks in a way that they liked.

Another challenge, somewhat obvious in retrospect, involved the programming element of the course. Students were encouraged to learn from and modify existing code, but many submissions were simply copied and pasted with no documentation of where the code originated. We suspect that students, especially non-CS students, think differently about plagiarism with respect to code than for prose. Once again, we suspect that better framing of the project, namely a clear articulation of requirements, and discussion of legitimate use of online code, could have helped to avoid this.

An underlying challenge throughout this course was defining a standard of creativity. In a course that relies so heavily on open-ended, exploratory, and experimental learning, we often received the reasonable yet unwelcome question "Did I do enough?" and grappled with finding an appropriate response. We thus encourage our fellow educators to look to the studio arts for inspiration and define a rubric that supports both creative and technological achievement (e.g., [13, 18]).

One positive facet of increased technological fluency is an improved sense of empowerment around basic electronics and programming. While electronics are ubiquitous in modern society, there is a general lack of understanding of how such devices actually function, and as a consequence,

Figure 9: Final project of Jordan (chemistry major) in 2016. He constructed a set of pendulums that influence the sound of a multi-oscillator circuit by passing over light sensors in the sculpture's base.

general intimidation around fixing and tinkering. As we suspect is the case with circuit bending and hardware hacking in general, we found that this course helped to demystify basic electronics and gave students a sense of empowerment, encouraging them to, for instance, go home and take apart a malfunctioning appliance (unplugged, of course!) to see what was wrong. This empowerment was reflected in the post-survey results, that showed increased levels of confidence across the board in technical areas (Table 3). We saw a similar empowerment in regards to programming, especially in terms of modifying existing code, which is not surprising given that we gave many coding examples to get students started. We suspect that a greater emphasis on programming throughout the course would have increased confidence even further in writing, rather than modifying, code.

6. CONCLUSIONS

Perhaps our favorite comment from the student evaluations is from Spring 2016: "While the course was not necessarily an electronics course it taught me a great deal about electronics and programming, as I came in with no knowledge or experience of either." We believe that the course *is* an electronics and programming course involving those aspects of technological fluency. That the student perceived it as something different - a course about sound-art and digital media - but ended up learning a great deal about electronics and programming fits exactly with our hope that a class such as this can be interesting and compelling to a wide variety of students. It also matches our belief that technological fluency can be taught such that the technological content can be seen as a natural component of the broader arts context.

7. REFERENCES

[1] Arduino. The Arduino home page. http://www.arduino.cc.

[2] A. Arpaci-Dusseau, O. Astrachan, et al. Computer science principles: Analysis of a proposed advanced placement course. SIGCSE '13. ACM, 2013.

[3] O. Astrachan and A. Briggs. The CS Principles project. *ACM Inroads*, 3(2):38–42, June 2012.

[4] O. Astrachan, S. Hambrusch, J. Peckham, and A. Settle. The present and future of computational thinking. In *SIGCSE '09*. ACM, 2009.

[5] Audacity. Free, open source, software for recording and editing sounds. http://audacity.sourceforge.net/.

[6] J. Bak, W. Verplank, and D. Gauthier. Motors, music and motion. In *Proceedings of TEI '15*. ACM, 2015.

[7] R. Bates, J. Goldsmith, et al. Science fiction in computer science education. SIGCSE '12. ACM, 2012.

[8] H. Bort, M. Czarnik, and D. Brylow. Introducing computing concepts to non-majors: A case study in gothic novels. SIGCSE '15. ACM, 2015.

[9] E. Brunvand. Lights! Speed! Action!: Fundamentals of physical computing for programmers. In *ACM SIGGRAPH '13 Courses*, 2013.

[10] E. Brunvand and N. McCurdy. Making Noise: Sound Art and Digital Media. Course web site, https://utah.instructure.com/courses/365251, 2016.

[11] E. Brunvand and P. Stout. Kinetic art and embedded systems: A natural collaboration. In *SIGCSE '11*. ACM, 2011.

[12] R. Bryant, R. Weiss, G. Orr, and K. Yerion. Using the context of algorithmic art to change attitudes in introductory programming. *J. Comput. Sci. Coll.*, 27(1):112–119, Oct. 2011.

[13] R. Clary, R. Brzuszek, and T. Fulford. Measuring creativity: A case study probing rubric effectiveness for evaluation of project-based learning solutions. *Creative Education*, 2(4), 2015.

[14] N. Collins. *Handmade Electronic Music: The Art of Hardware Hacking*. Routledge, 2 edition, 2009.

[15] N. R. C. Committee on Information Technology Literacy. *Being Fluent with Information Technology*. The National Academies Press, 1999.

[16] N. A. o. E. Committee on Technological Literacy. *Technically Speaking: Why All Americans Need to Know More About Technology*. The National Academies Press, 2002.

[17] C. Cox and D. Warner. *Audio Culture: Readings in Modern Music*. Continuum, 2007.

[18] M. Csikszentmihalyi. *Creativity: The Psychology of Discovery and Invention*. Harper Perennial, 1997.

[19] Q. Cutts, S. Esper, and B. Simon. Computing as the 4th "R": a general education approach to computing education. ICER '11. ACM, 2011.

[20] C. Dierbach, H. Hochheiser, et al. A model for piloting pathways for computational thinking in a general education curriculum. SIGCSE '11. ACM, 2011.

[21] J. P. Dougherty, T. Dececchi, et al. Information technology fluency in practice. ITiCSE-WGR '02. ACM, 2002.

[22] J. Freeman, B. Magerko, et al. Engaging underrepresented groups in high school introductory computing through computational remixing with EarSketch. SIGCSE '14. ACM, 2014.

[23] Q. R. Ghazala. The folk music of chance electronics: Circuit-bending the modern coconut. *Leonardo Music Journal*, 14, 2004.

[24] R. Ghazala. *Circuit-Bending: Build Your Own Alien Instruments (ExtremeTech)*. Wiley, 2005.

[25] M. Guzdial. A media computation course for non-majors. *SIGCSE Bull.*, 35(3):104–108, June 2003.

[26] J. M. Heines, G. R. Greher, and S. Kuhn. Music performamatics: Interdisciplinary interaction. SIGCSE '09. ACM, 2009.

[27] J. M. Heines, G. R. Greher, and S. A. Ruthmann. Techniques at the intersection of computing and music. ITiCSE '12. ACM, 2012.

[28] R. Hoar. Generally educated in the 21st century: The importance of computer literacy in an undergraduate curriculum. WCCCE '14. ACM, 2014.

[29] T. Holmes. *Electronic and Experimental Music: Technology, Music, and Culture*. Routledge, 2012.

[30] M. D. LeBlanc, T. Armstrong, and M. B. Gousie. Connecting across campus. SIGCSE '10. ACM, 2010.

[31] B. Magerko, J. Freeman, et al. Earsketch: A steam-based approach for underrepresented populations in high school computer science education. *Trans. Comput. Educ.*, 16(4), Sept. 2016.

[32] B. Manaris and A. Brown. *Making Music with Computers: Creative Programming in Python*. Chapman & Hall / CRC, 2014.

[33] A. Misra, D. Blank, and D. Kumar. A music context for teaching introductory computing. *SIGCSE Bull.*, 41(3):248–252, July 2009.

[34] A. H. Neset and B. London. *Soundings: A Contemporary Score*. Museum of Modern Art, 2013.

[35] L. Perković, A. Settle, S. Hwang, and J. Jones. A framework for computational thinking across the curriculum. ITiCSE '10. ACM, 2010.

[36] B. Pollack. Now hear this: Sound art has arrived. Art News (online), November 2013. http://www.artnews.com/2013/11/14/now-hear-this-sound-art-has-arrived.

[37] A. Ruthmann, J. M. Heines, et al. Teaching computational thinking through musical live coding in Scratch. SIGCSE '10. ACM, 2010.

[38] B. Sawyer, J. Forsyth, et al. Form, function and performances in a musical instrument MAKErs camp. SIGCSE '13. ACM, 2013.

[39] R. M. Schafer. *A Sound Education: 100 Exercises in Listening and Sound-Making*. Arcana Editions, Ontario, Canada, 1992.

[40] B. Simon, P. Kinnunen, L. Porter, and D. Zazkis. Experience report: CS1 for majors with media computation. ITiCSE '10. ACM, 2010.

[41] SoundCloud. "The world's leading social sound platform". https://soundcloud.com/.

[42] J. M. Wing. Computational thinking. *Commun. ACM*, 49(3), Mar. 2006.

[43] H. Yanco, H. J. Kim, F. G. Martin, and L. Silka. Artbotics: Combining art and robotics to broaden participation in computing. In *AAAI Spring Symposium*, Stanford, CA, March 2006.