# Engaging Underrepresented Groups in High School Introductory Computing through Computational Remixing with EarSketch

Jason Freeman[1], Brian Magerko[1], Tom McKlin[3], Mike Reilly[2], Justin Permar[1], Cameron Summers[1], and Eric Fruchter[1]

[1] School of Music, Digital Media Program
Georgia Institute of Technology
Atlanta, GA 30332
001-404-385-7257
{jason.freeman, magerko, jpermar, csummers3, efruchter}@gatech.edu

[2]Lanier High School
Gwinnett County Public Schools
Sugar Hill, GA 30518
001-678-765-4040
Michael_Reilly@Gwinnett.k12.ga.us

[3]The Findings Group
1201 Clairmont Road, Suite 305
Decatur, GA 30030
001-404-633-9091
tom@theFindingsGroup.com

## ABSTRACT
In this paper, we describe a pilot study of EarSketch, a computational remixing approach to introductory computer science, in a formal academic computing course at the high school level. The EarSketch project provides an integrated curriculum, Python API, digital audio workstation (DAW), audio loop library, and social sharing site. The goal for EarSketch is to broaden participation in computing, particularly by traditionally underrepresented groups, through a thickly authentic learning environment that has personal and industry relevance in both computational and artistic domains. The pilot results show statistically significant gains in computing attitudes across multiple constructs, with particularly strong results for female and minority participants.

## Categories and Subject Descriptors
J.5 [**Arts and Humanities**], K.3.2 [**Computer Education**]: Computers and Information Science Education – *computer science education, curriculum*.

## Keywords
Remixing; music composition; broadening participation; CS education; computing principles; music production

## 1. INTRODUCTION
Our research seeks to address the problem of recruitment and retention of underrepresented populations in high school computing courses [1]. STEAM (science, technology, engineering, arts, and math) approaches to computing education have attempted to engage such students by integrating artistic creativity within pedagogical practices [2], [3]. However, we contend such approaches rarely satisfy two criteria crucial for success – *cultural relevance in the artistic domain* and *authentic learning in both computing and the arts.*


*SIGCSE'14*, March 5–8, 2014, Atlanta, Georgia, USA.

ACM 978-1-4503-2605-6/14/03…$15.00.
http://dx.doi.org/10.1145/2538862.2538906

The cultural relevance of an artistic domain relates to how central the artistic practice is to the target student culture. If an artistic domain does not effectively engage a large enough portion of the target student population (e.g. square dancing), then its integration into a computational curriculum may have little positive effect or even negative repercussions. We therefore assert that STEAM designers must consider artistic practices that have broad appeal across ethnicities, socio-economic backgrounds, and gender. To further support broad and effective adoption, the integrated artistic practices must have low barriers of entry, 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).

The integration of artistic practices into computing pedagogy can also be considered from the perspective of authenticity. The authenticity of a learning experience, according to Lee and Butler [4], is based on how well it focuses on the interrelated authentic learning practices of: a) having personally meaningful learning experiences; b) learning that relates to the world outside of the learning context; c) learning that encourages thinking within a particular discipline (e.g. music composition); and d) allowing for assessment that reflects the learning process. *Thick authenticity*, according to Shaffer and Resnick [5], meets all of these requirements in a single approach / system. We contend that a STEAM approach is most effective when both the technical and artistic elements of the learning experience are thickly authentic. For example, in a computational music context, thick authenticity suggests that students learn to code in an industry-relevant language, use music production paradigms common in the recording industry, and create music in popular, personally relevant styles and genres. Because of music's near-universal appeal and cross-cultural relevance [6], we believe it is a particularly compelling domain for thickly authentic STEAM learning experiences.

In our research, we hypothesize that a thickly authentic music and CS learning environment increases engagement in computer science, particularly among groups traditionally underrepresented in the discipline. We have developed EarSketch for introductory computing courses in high school and especially the new Computer Science Principles course that will be an Advanced Placement course by the 2016-2017 school year.

EarSketch focuses on computational music remixing, which is a popular composition form in hip hop music [7]. Music remixing is similar to creating a collage of pictures from magazines. EarSketch attempts to connect to students in a culturally relevant fashion that spans gender, ethnicity, and socioeconomic status. It does so by explicitly connecting the learning environment to popular music production software (Figure 1), by drawing from industry practice in the features it supports, and by visualizing computational output within standard industry paradigms. It also focuses on the level of beats, loops, and effects more than individual notes, enabling students with no background in music theory or composition to begin creating personally relevant music immediately. Unlike more graphically oriented environments such as Scratch, EarSketch builds directly on professional development techniques using an industry-relevant, text-based programming language (Python).

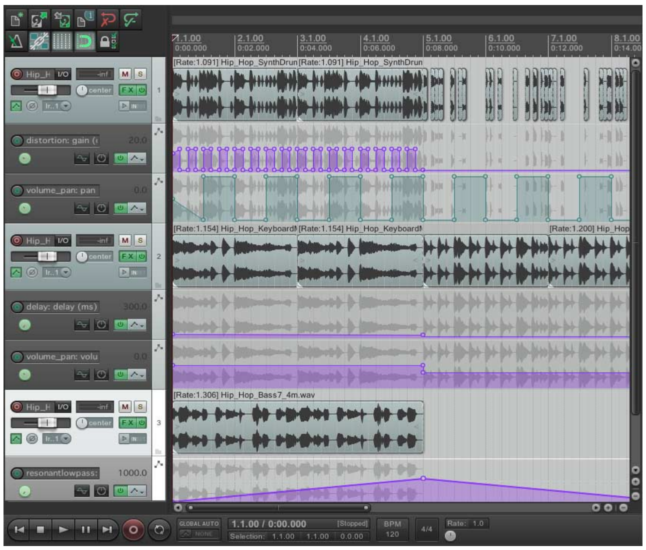

**Figure 1: The Reaper digital audio workstation (DAW) music production software integrated into EarSketch.**

EarSketch has been iteratively tested in afterschool programs at the 4th and 5th grade level, in individual user tests at Georgia Tech with high school students, and in summer workshops for high school students. The results of our 2012 summer pilot suggested that students' attitudes positively and statistically significantly increased across three constructs at $p<.01$: Computing Confidence, Motivation to Succeed in Computing, and Creativity [8]. Students indicated that they are statistically significantly more likely to see themselves in a computing field in the future as a result of the workshop. To further investigate how EarSketch impacted students' intent to persist in computing, a correlation analysis was conducted whereby the change in responses from Before to Now across our survey constructs were entered into a Pearson's correlation. The results indicated that to the extent that students gained in confidence, belongingness and creativity, their intent to persist in computing increased [8]. Interestingly, growth in creativity was statistically significantly correlated with growth in confidence, motivation and belongingness.

While our preliminary work has shown statistically significant findings, it was done primarily within the context of a week-long summer workshop, which is a very different learning environment from an academic course. To better understand EarSketch's role in this more formal learning environment, we conducted a 10 week pilot EarSketch program in a high school introductory computing course (see Section 3 for details). The larger number of participants in this pilot, as compared to our prior studies, also enabled us to analyze differences in outcomes for female and male and majority and minority participants, helping us to better understand EarSketch's potential to reach traditionally underrepresented populations. The remaining sections of this article summarize work related to this research, present the methods, findings, and results from the high-school pilot study, and discuss the results and future work.

## 2. RELATED WORK

Although computers are an important part of everyday life, interest in programming remains relatively low for many demographics [1]. Common introductory pedagogy lacks relevance, dating back to a time when computers could only manipulate text in a terminal. Leveraging the ability of computers to render complex audio and visuals to craft media-centric curricula is a promising way of teaching computer science and keeping students engaged in coursework [9].

Several projects have attempted to frame mathematics and computer science education in an artistic or cultural context, focusing mainly on visuals and interactivity. For example, Eglash's Culturally Situated Design Tools allow students to digitally simulate art from various cultures, increasing engagement for students who might typically find mathematics uninviting or feel culturally excluded from the subjects [10]. The eTextiles project blends programmable circuits with fabrics, engaging students' creativity while lowering the technical barriers-to-entry [11]. Both Scratch and Alice teach programming through the creation of interactive media [12], [13], [14]. All of these platforms support a diverse range of projects, giving students the tools to create something that resonates with and captivates them.

Like visual and interactive art, music is an approachable domain with broad appeal, and its production relies on several important concepts in computer science [8]. The MediaComp course at Georgia Tech has successfully engaged women in computing practices through the manipulation of different media forms (images, sound, etc.) with code [15], [16]. Other attempts have been made to reinforce programming concepts through music education in the hopes of appealing to a broad range of students. In a four-year curriculum study on material linking computer science and digital audio, students demonstrated a boost to their perceived aptitude of the science of sound [17]. It was also found that students benefited most from practical projects that they felt were relevant to themselves and to others. Musicomputation, an advanced computer music/science course, also found success in crafting coursework that was relevant to the students' areas of interest, and linking those back to computer science [18]. Other curriculums have focused on other areas of music creation. The Squeak-based curriculum Sound Thinking, for example, has students design and program novel musical instruments [19].

While the above approaches have reported success, it is unclear how authentic the learning experience is in both target domains. For example, the MediaComp course uses audio at a very low level (e.g. writing an algorithm to compute reverb) while Scratch presents a coding environment that is visually oriented rather than using a typical IDE. It is unclear how these approaches in both the artistic domain and technical domain offer thick authenticity (as mentioned in the Introduction), and therefore may not be as effective as approaches that tackle both domains authentically.

# 3. METHODS

## 3.1 Pilot Population and Learning Context

EarSketch was piloted with sixty-nine 9th graders (first-year high school students) as a module in the Computing in the Modern World course at Lanier High School. The 10-week pilot took place from January to March 2013. Lanier High School is a public high school in Sugar Hill, Georgia with 1605 students, 44% of whom are on free or reduced lunches (suggesting a diverse range of socioeconomic backgrounds). The 69 students enrolled in the pilot course were 30% female, 7% Hispanic, and 16% African-American.

Computing in the Modern World is the first course in a three course computing pathway in the Career, Technical and Agricultural Education (CTAE) program in Georgia. [20]. Its curriculum includes some programming and computer science principles along with modules on topics such as web design, networking, and hardware. The EarSketch pilot served as the programming module of the course.

Lanier High School teaches Computing in the Modern World within its Center for Design and Technology (CDAT). Ninth graders in CDAT take four courses: 1) science (biology or chemistry), 2) language arts, 3) digital media (Computing in the Modern World), and 4) Physical Education/Health [21]. The CDAT teaching methodology is founded upon a project-based, "flipped classroom" learning model, which resembles a studio-based learning approach that has seen effective application in computer science curricula [22]. Students within CDAT are expected to be hands-on, interdisciplinary and independent; projects are commonly utilized as a means to demonstrate knowledge and comprehension [23].

The pilot teacher, Mike Reilly, created a curriculum derived from the openly accessible EarSketch online curriculum. He taught two separate sections of 30-35 students and installed the EarSketch software in standard PC labs equipped with high fidelity headphones. Mr. Reilly was an experienced software developer and computer science teacher but had little prior experience with music, which aligns with our expectations for the background of most EarSketch teachers.

The students enrolled in the EarSketch pilot did not self-select for a course with EarSketch or even a course in computer science; their interest was in the broader CDAT program at Lanier. Students did not begin the class with strong experience in computers and computing, as evidenced by their responses to an engagement survey (see Evaluation below).

## 3.2 Curriculum and Teaching Strategies

During the pilot, the teacher focused on 1) maximizing student motivation, 2) interweaving the teaching of programming and music concepts and techniques, 3) encouraging creative application of the acquired knowledge via projects, and 4) benchmarking progress via quizzes. The remainder of this section details each of these four areas of focus.

### *3.2.1 Motivating Students*

The teacher employed a number of complementary strategies to motivate students. The immediate goal of producing personally expressive, creative music motivated students to learn programming. In particular, the course emphasized how programming (as opposed to creating music manually in the DAW) enables students to work more efficiently by automating repetitive tasks, to more rapidly experiment with different musical possibilities and variations, and to create more unique musical structures and sounds to make their music stand out. The teacher also emphasized the importance of programming to the future of the music industry and the opportunities to become industry leaders through developing new tools for making, performing, and distributing music. Finally, he noted the broader applicability of programming skills and the demand for computer scientists and software engineers across many different industries.

These points were reinforced by references to famous musicians, such as Will.I.Am [24], who have discussed the importance of programming expertise to the music industry's future. Near the end of the pilot, Young Guru [25], a professional audio engineer and DJ best known for his collaborations with Jay Z, visited the class (Figure 2) to personally communicate his excitement about programming within the field of music (Young Guru also created much of the audio loop library content for EarSketch.)

**Figure 2: A student (sitting) shares his project with Young Guru (standing) at Lanier High School.**

### *3.2.2 Interweaving Programming and Music*

Throughout the pilot, the curriculum taught music, music technology, and computing concepts together, always linking new computational concepts to a musical application. Weeks 1 and 2 provided an introduction to fundamental music and music production concepts, including tracks, tempo, instrument pairings, and sound effects; relevant programming concepts included variables, function calls, and core EarSketch API functions for placing sounds and effects on tracks. In addition, the course addressed sharing both music and code, open licensing models in both domains, and sharing practices on the EarSketch social media site. Throughout the course, but especially early on, the teacher played popular music to demonstrate specific concepts.

Week 3 and 4 introduced rhythms and repetitions; relevant programming concepts included strings and iteration. Weeks 5 and 6 focused on the arrangement of musical compositions as a series of musical sections, with students learning to define their own functions to encapsulate and re-use musical sections. In the remainder of the pilot, students reviewed and reinforced previous material, deepened their understanding of computing concepts, API functions, and musical techniques, and developed individual projects that served as final projects for the EarSketch module.

### 3.2.3 Quizzes and Projects

To ensure that students effectively acquired and internalized the EarSketch curriculum, the teacher created two types of assessments: 1) quizzes, which were not multiple-choice exams but instead asked for answers in the form of handwritten Python code; and 2) projects that demonstrated integration and application of music and programming concepts and techniques, including utilization of the EarSketch API.

The quizzes all asked the students to write Python scripts to make music with stated computational and musical requirements. The requirements incorporated learned programming techniques (such as variables, constants, strings, loops, functions, and lists) and EarSketch API functions (to perform tasks such as placing audio files on the timeline, creating rhythmic beats, and controlling track effects and effect envelopes). Quiz 1, for example, asked the students to write a Python script that creates a minimum of three tracks of varying lengths, uses multiple sound files, and applies volume level adjustments to the tracks. Projects were similar to the quizzes, as they required the development of working code to create music given a set of requirements, but were more open-ended in order to provide students with more opportunities for artistic creativity. Projects typically required students to incorporate a particular set of API functions, computing concepts, and musical structures but left most of the choices about how they should be incorporated into the music up to the student. Projects were completed on the computer rather than written on paper.

Both quizzes and projects were commonly "graded by interview", a technique similar to pedagogical code review processes [22]. The teacher engaged in one-on-one discussions to peruse each student's code and composition. This grading format allowed the teacher to identify errors that an individual student was making, such as incorrect API usage or improper application of iteration, which resulted in personalized curriculum customization in the hope of obtaining improved student comprehension.

## 4. EVALUATION

Students in the pilot were measured on attitudes changes and content knowledge. A student engagement survey was given to students as a pre/post retrospective at the end of the pilot. The approach asks students to consider how they felt "before" the course and to indicate their perceptions "now" at the end of the course. This format was chosen to diminish response shift bias.

The instrument measures seven psychosocial constructs: 1) computing confidence (e.g. "I can get good grades in computing"); 2) computing enjoyment (e.g. "I feel comfortable working with a computer"); 3) computing importance/perceived usefulness (e.g. "I believe that it is very important for me to learn how to use a computer."); 4) motivation to succeed (e.g. "I like solving computer problems"); 5) identity and belonging in computing (e.g. "I take pride in my computer abilities"); 6) intent to persist in computing (e.g. "I can see myself working in the field of computing"); and 7) creativity in computing (e.g. "I am able to be very expressive and creative while doing computing"). The forty-one question survey, which included negated formulations, was designed to address whether students report increases in these seven pyschosocial constructs as a result of participating in program activities and was based on scales from [26] and [27]. Cronbach's Alpha for each construct ranged from 0.691 to 0.893.

Literature reviews suggest that there is no standard for reporting responses to individual likert-scaled responses; the literature reveals significant *disagreement* among psychostatisticians and others about whether a likert scale is nominal (appropriate for n, % reporting) or ordinal (appropriate for mean, standard deviation). However, psychostatisticians agree that when several likert questions are summed to compute an average construct, then the data may be treated as interval data measuring a latent variable. Parametric statistical tests such as ANOVA or t-test can be applied. Our analysis consisted of a paired samples t-test whereby statistically significant changes from before to now were assessed.

Our content knowledge assessment (CKA) measures students' mastery of pre-defined computing standards. It was administered as a pre/post assessment with ten multiple choice items. It covers typical introductory computing topics in Python such as function definitions, calls, and arguments; iteration and loop indices; and syntax and debugging; as well as EarSketch functionality. Items were classified by difficulty based on student answers to determine if students learned the concept being tested [28].

Overall, the results of the student engagement survey (Table 1) indicate a positive and statistically significant increase in students' attitudes towards computing across all constructs at $p<0.01$. Before the course, female students were statistically significantly lower in confidence ($p=.041$) than male students; likewise, female students rated their identity and belongingness in computing lower than male students ($p=.078$). However, after the course, these gender differences disappeared. Additionally, gains in female motivation to succeed was greater than males ($p=0.045$) suggesting that the course may be particularly effective in increasing female motivations to persist on computing problems.

Table 2 shows that both minority and majority racial/ethnic groups' attitudes toward computing increased statistically significantly in all constructs. While race/ethnicity groups alone didn't show as strong a correlation (i.e. not statistically significant) as gender, after the course minority students showed slightly higher ratings than majority students in four constructs - confidence, perceived usefulness, intentions to persist, and creativity. This could suggest that EarSketch is particularly effective for minority students. Follow up surveys and qualitative studies may give more insight into the long-term impacts of EarSketch on intention to persist in computing.

Students across all gender and racial/ethnic groups rated the course as being good or excellent. Many free-response comments specifically addressed the thickly authentic aspects of the learning environment: one student "got to express my ideas and it was fun and inspiring to see that I could be good at computing;" another liked "learning how music is made and how we can learn and get good at doing things that people in the music industry do now;" and another enjoyed "making my own music tracks that people, including myself, actually liked." Though not statistically significant, female and minority students rated the course higher than their counterparts, again suggesting the possibility that EarSketch is particularly effective among underrepresented groups in computing.

The results of the content knowledge assessment (CKA) indicated statistically significant increases in computing content knowledge from pre to post. On average, students answered 23% of the items correct at the beginning of the course (pre) and 60% correct at the end of the course (post), and they exhibited statistically significant gains in 9 of the 10 CKA items. (There was an error in the formulation of the tenth question that did not exhibit gains.)

**Table 1. Constructs by Gender**

| | *Female (n=21)* | | | | *Male (n=47)* | | | |
|---|---|---|---|---|---|---|---|---|
| *Construct* | *Before* | *Now* | *t-test* | *Effect Size* | *Before* | *Now* | *t-test* | *Effect Size* |
| Computing Confidence | 2.58 | 3.88 | <0.01** | 1.95$^{L}$ | 3.00 | 3.99 | <0.01** | 1.26$^{L}$ |
| Computing Enjoyment | 3.74 | 4.29 | <0.01** | 0.93$^{L}$ | 3.79 | 4.36 | <0.01** | 1.05$^{L}$ |
| Importance/ Perceived Usefulness | 3.64 | 4.39 | <0.01** | 1.48$^{L}$ | 3.89 | 4.46 | <0.01** | 0.93$^{L}$ |
| Motivation to Succeed | 2.84 | 3.82 | <0.01** | 1.43$^{L}$ | 3.09 | 3.67 | <0.01** | 0.59$^{M}$ |
| Identity and Belongingness | 2.86 | 3.75 | <0.01** | 1.03$^{L}$ | 3.29 | 3.89 | <0.01** | 0.70$^{M}$ |
| Intent to Persist | 2.93 | 3.80 | <0.01** | 0.99$^{L}$ | 3.29 | 3.83 | <0.01** | 0.61$^{M}$ |
| Creativity in Computing | 3.19 | 3.86 | <0.01** | 0.89$^{L}$ | 3.33 | 3.93 | <0.01** | 0.73$^{M}$ |

Scale= 1, Strongly Disagree to 5, Strongly Agree. **p<.01, *p<.05, †p<.10. Negatively worded statements were reverse-coded to assess construct means. Effect size= (s) Small (0.20), (M) Medium (0.50), (L) Large (0.80).

**Table 2. Constructs by Race/Ethnicity**

| | *Minority (n=28)* | | | | *Majority (n=41)* | | | |
|---|---|---|---|---|---|---|---|---|
| *Construct* | *Before* | *Now* | *t-test* | *Effect Size* | *Before* | *Now* | *t-test* | *Effect Size* |
| Computing Confidence | 2.95 | 4.05 | <0.01** | 1.48$^{L}$ | 2.82 | 3.90 | <0.01** | 1.41$^{L}$ |
| Computing Enjoyment | 3.75 | 4.30 | <0.01** | 1.02$^{L}$ | 3.79 | 4.36 | <0.01** | 1.03$^{L}$ |
| Importance/ Perceived Usefulness | 3.87 | 4.46 | <0.01** | 0.97$^{L}$ | 3.78 | 4.43 | <0.01** | 1.16$^{L}$ |
| Motivation to Succeed | 3.04 | 3.72 | <0.01** | 0.75$^{M}$ | 3.02 | 3.73 | <0.01** | 0.81$^{L}$ |
| Identity and Belongingness | 3.18 | 3.80 | <0.01** | 0.66$^{M}$ | 3.14 | 3.88 | <0.01** | 0.93$^{L}$ |
| Intent to Persist | 3.16 | 3.86 | <0.01** | 0.79$^{M}$ | 3.22 | 3.81 | <0.01** | 0.67$^{M}$ |
| Creativity in Computing | 3.41 | 4.02 | <0.01** | 0.84$^{L}$ | 3.21 | 3.84 | <0.01** | 0.76$^{M}$ |

Scale= 1, Strongly Disagree to 5, Strongly Agree. Assessment: Good=Above 4.0; Attention=Below 4.0; Action=Below 3.5. **p<.01, *p<.05, †p<.10. Negatively worded statements (n) were reverse-coded to assess construct means. Effect size= (s) Small (0.20), (M) Medium (0.50), (L) Large (0.80).

## 5. DISCUSSION

These evaluation results suggest that EarSketch's thickly authentic music and computing learning environment effectively teaches introductory computing concepts to high school students in a formal academic course and that it substantially improves student attitudes towards computing. Some of the data suggests that EarSketch may be particularly effective for groups that are traditionally underrepresented in computing.

We must temper these encouraging results with a few important caveats. The pilot took place within Lanier's CDAT program, an innovative flipped classroom, studio-based setting that is still unusual within most high schools. The teacher, Mike Reilly, is a project collaborator who invested additional time and energy into customizing the curriculum to best fit his classroom and acquiring music and music technology skills important to the pedagogy. While students did not self-select to participate, they did self-select for the CDAT program, indicating a broader interest in design and technology. The pilot program concluded with a personal visit from a major hip hop figure (Young Guru, as shown in Figure 2) to critique student work and argue for the importance of computation to the music industry. These specific aspects of the pilot cannot, of course, be replicated at scale (though we are creating videos with music industry role models intended to serve similar motivational purposes to Young Guru's visit). We therefore cannot yet assume that the results of this pilot can be replicated at scale either. However, the positive outcomes of this pilot as well as our summer camp pilots (see Section 1) do together suggest EarSketch's effectiveness in multiple learning contexts.

Moving forward, we plan to deploy and evaluate EarSketch on a larger scale, to determine the effectiveness of EarSketch when the unique aspects of the Lanier pilot are absent. To this end, we: a) have made our curriculum and software available online, b) are developing formal teacher training materials and planning teacher training workshops, and c) are collaborating with a metro Atlanta school district to deploy EarSketch at multiple high schools locally. We are also teaching EarSketch as part of a MOOC on Coursera, with approximately 24,000 students enrolled. We are also beginning to create an entirely web-based version of EarSketch to support more straightforward deployment and a more integrated user experience.

We also wish to learn more about why EarSketch has been effective, developing new assessment tools and software enhancements, informed by learning sciences, to better understand the specific aspects of EarSketch that have contributed to particular student engagement and content knowledge outcomes.

## 6. ACKNOWLEDGMENTS

Thanks to Cockos, Inc., Lanier High School, the Marist School, Georgia Tech's Institute for Computing Education, and our project team (Barbara Ericsson, Ziwen Fan, Tom Jordan, Elise Livingston, Christopher Michaud, Scott McCoid, Casey Feisler, Aneesh Vartakavi, Brad Short, Alex Duncan, Laurie Marion, David Poore, Tom Jenkins, Hera Kan, Neeraj Vaidya, Shannon Yao, Jonathan Streater, Erica Richards, Charles Hancock, Arun Marsten, Juan Carlos Martinez, Anand Mahadevan, Michael Madaio, Mikhail Jacob, Young Guru, and Richard Devine). This work is funded by the National Science Foundation under CNS grant #1138469. EarSketch is available at http://earsketch.gatech.edu.

## 7. REFERENCES


[1] J. Cuny, "Address to the Computer Science Community," presented at the CE21 PI and Community Meeting, New Orleans, LA, 21-Jan-2011.

[2] J. D. Burge and T. L. Suarez, "Preliminary analysis of factors affecting women and african americans in the computing sciences," in *Proceedings of the 2005 conference on Diversity in computing*, Albuquerque, New Mexico, USA, 2005, pp. 53–56.

[3] Eglash, R. et al, "Culturally Situated Design Tools: Ethnocomputing from Field Site to Classroom," *American Anthropologist*, vol. 108, no. 2, pp. 347–362, 2006.

[4] H.-S. Lee and N. Butler, "Making authentic science accessible to students - International Journal of Science Education," 2003.. [Online]. Available: http://www.informaworld.com/smpp/content~content=a713865046&db=all. [Accessed: 07-Jun-2010].

[5] D. W. Shaffer and M. Resnick, "Thick authenticity: new media and authentic learning," *J Interact Learn Res*, vol. 10, no. 2, pp. 195–215, Dec. 1999.

[6] I. Peretz and R. J. Zatorre, "Brain organization for music processing," 2004.

[7] S. Vaidhyanathan, *Copyrights and copywrongs : the rise of intellectual property and how it threatens creativity*. New York: New York University Press, 2003.

[8] B. Magerko et al, "Tackling Engagement in Computing with Computational Music Remixing," in *Proceedings of the ACM SIGSCE Technical Symposium on Computer Science Education, Denver, CO, USA, 2013*.

[9] M. Guzdial and E. Soloway, "Teaching the Nintendo generation to program," *Commun Acm*, vol. 45, no. 4, pp. 17–21, Apr. 2002.

[10] R. Eglash, A. Bennett, C. O'donnell, S. Jennings, and M. Cintorino, "Culturally Situated Design Tools: Ethnocomputing from Field Site to Classroom," *Am. Anthr.*, vol. 108, no. 2, pp. 347–362, 2006.

[11] 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 twenty-sixth annual SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2008, pp. 423–432.

[12] 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.

[13] 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.

[14] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and others, "Scratch: programming for all," *Commun. Acm*, vol. 52, no. 11, pp. 60–67, 2009.

[15] M. Guzdial, "A media computation course for non-majors," in *ACM SIGCSE Bulletin*, 2003, vol. 35, pp. 104–108.

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

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

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

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

[20] "CTAE." [Online]. Available: http://public.doe.k12.ga.us/Curriculum-Instruction-and-Assessment/CTAE/Documents/Career_Pathway_Chart_080612.pdf. [Accessed: 31-Aug-2013].

[21] "CDAT." [Online]. Available: http://cdat.lanierhs.org/. [Accessed: 31-Aug-2013].

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

[23] D. R. Krathwohl, B. S. Bloom, and B. Masia, *Taxonomy of educational objectives: The classification of educational goals*, vol. 2. New York: David McKay Company, Inc., 1956.

[24] T. Olmstead, "Will.I.Am Says Coders Are Today's Rock Stars," 30-Aug-2012. .

[25] L. Stouffer, "P2P GA Tech Computer Engineering / Lanier High," *People 2 People*, WSB-TV, 11-Mar-2013.

[26] E. Wiebe, L. Williams, K. Yang, and C. Miller, "Computer Science Attitude Survey," *computer*, vol. 14, no. 25, pp. 0–86, 2003.

[27] G. Knezek and R. Christensen, "Validating the Computer Attitude Questionnaire (CAQ)." 1996.

[28] F. Lord, "The Relationship of the Reliability of Multiple Choice Test to the Distribution of Item Difficulties," *Psychometrika*, vol. 18, pp. 181–194, 1952.