Improving Computational Thinking through Collaborative Music-Leadership Based Learning

FATEMEH JAMSHIDI and DANIELA MARGHITU*, Auburn University, USA
OLIVIA COOK, Auburn University, USA

The gender gap in technical fields is likely to persist for generations. Potential reasons include gender differences in aptitude, interest, and academic environment. Prior research at the high school level has shown remarkable impacts on student engagement and intention to participate in computing, especially for female students. Computer music is one way to engage students in computer science (CS) by prioritizing personal expression, creativity, and aesthetics. This paper aims to introduce a mixed-methods study to determine ways to promote CS among underrepresented groups. It describes an adaptation of EarSketch and Scratch for use in an secondary-school-level introductory programming course at an open-access summer camp at Auburn University. American girls aged 12–16 participated in pre-, post-, and follow-up surveys while attending the CS camp. The paper explains a quasi-experimental study suggesting that a combination of a leadership-authentic learning environment predicts increased intentions to persist via identity and creativity. To evaluate students' performance, we conducted path analysis exploring factors related to student engagement and intention to persist in CS. We suggest that providing an entertaining CS programming experience environment in K-12 classrooms is crucial for boosting underrepresented groups' confidence and interest in CS.

Additional Key Words and Phrases: Computer Education, Music Technology, Music Computing, Leadership, Scratch, EarSketch

ACM Reference Format:
Fatemeh Jamshidi, Daniela Marghitu, and Olivia Cook. 2024. Improving Computational Thinking through Collaborative Music-Leadership Based Learning. 1, 1 (March 2024), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

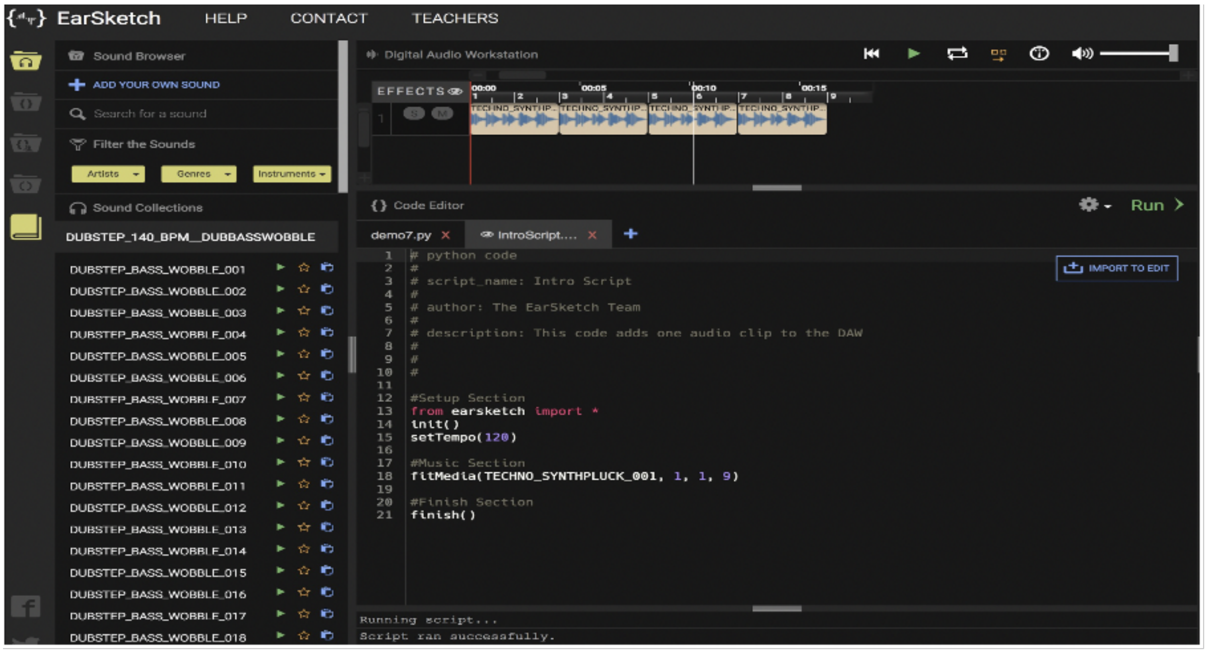
In this research, we aim to broaden the participation of underrepresented populations in high school computing courses and future careers. Various approaches have sought to increase underrepresented recruitment and retention in CS education. Some ideas include making CS more relevant or engaging through culturally specific technology [3]. These technologies include computer games and digital media or integrating artistic creativity within pedagogical practices as an "in" for motivating students [18, 19].

We describe our experience based on the paper "Engaging underrepresented groups in high school introductory computing through computational remixing with EarSketch" [5] with a focus on high school female students. According to the paper, "the cultural relevance of an artistic domain relates to how central the artistic practice is to the target student culture". Therefore, after a great deal of research on the camp population culture, we designed a Science, Technology, Engineering, Art, and Mathematics (STEAM) curriculum considering artistic practices that have the closest connection to students' background and culture.

Authors' addresses: Fatemeh Jamshidi, fzj0007@auburn.edu; Daniela Marghitu, marghda@auburn.edu, Auburn University, Computer Science and Software Engineering, Auburn, Alabama, USA, 36830; Olivia Cook, Auburn University, Auburn, USA, ojc0001@auburn.edu.

EarSketch (as visualized in Figure 1) is an authentic learning environment for engaging students' interest in CS concepts through the remixing of musical sounds using the Python programming language and a Digital Audio Workstation. It includes a programming environment, digital audio workstation, structured curriculum, an audio loop library, and collaboration tools, allowing students an opportunity to experience both JavaScript and Python programming [23].

Scratch is a block-based platform developed to encourage young students to develop programming concepts and develop multimedia communication skills. It includes a visual system of "Tiles" that contain blocks where students can connect to create programs. These programs direct the characters and objects in the game. Scratch can generate and play sounds using various components within the Sound category [20].



```python
# python code
#
# script_name: Intro Script
#
# author: The EarSketch Team
#
# description: This code adds one audio clip to the DAW
#
#
#

#Setup Section
from earsketch import *
init()
setTempo(120)

#Music Section
fitMedia(TECHNO_SYNTHPLUCK_001, 1, 1, 9)

#Finish Section
finish()
```

```
Running script...
Script ran successfully.
```

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

This research defines the most critical difficulties in integrating the arts into introductory CS courses (and approaches to address the challenges); it presents evidence to suggest the significant impact of artistic STEAM learning on student content knowledge and interest in computing.

In contrast to other similar programs [9, 11, 23], which mostly focus on teaching musical note structures and music remixing through computational thinking, this investigation goes one step further: improving students' creativity and imagination by teaching them how to construct beats with EarSketch for the games they have developed using Scratch. The remaining sections of this research summarize work related to this theory, present the methods, findings, and results from the high-school pilot study, and discuss the results and future work.

2 RELATED WORK

Computers nowadays have become part of everyday life; however, interest in programming remains relatively low, especially in underrepresented groups. Integrating visual and interactive art and music with CS is an approachable domain, and music production relies on several essential concepts in computer science [1, 14, 15]. The MediaCom course at Georgia Tech aims to engage women in computing practices by manipulating different media forms (images and sounds) with code [6, 7, 12].

Other attempts have been made to reinforce programming concepts through music education to enhance recruitment and retention in computing fields. In a four-year curriculum study on material linking CS and digital audio, students implemented a boost to their understanding of the science of sound [2]. It was also found that students benefited most from practical projects that they felt were relevant to themselves and others. Another project called Musicomputation, an advanced computer music/science course, “found success in crafting coursework that was relevant to the students’ areas of interest and linking those back to computer science” [16]. Other curriculum have investigated other realms of music creation. For example, in the Squeak-based curriculum Sound Thinking, students are asked to design and program novel musical instruments [8].

Several projects have attempted to bring a new approach to CS and mathematics by adding an artistic or cultural context. For instance, a web-based software application, Culturally Situated Design Tool (CSDTs), allows students to create cultural arts simulations such as Native American beadwork, African American cornrow hairstyles, and urban graffiti, and so forth using underlying mathematical principles [4].

Besides, another Culturally Situated Design Tool, Musical Rhythms, allows students to simulate traditional musical rhythms and create new rhythms of their own. Both Scratch and Alice teach programming through the creation of interactive media, improving students’ creativity while lowering the technical barriers-to-entry [13, 17]. These platforms support a diverse range of projects, giving students the tools to create something that captivates them. Music is a unique approach with broad applications, and its production has significant concepts in computer science. There have been some attempts to teach programming concepts through music education, hoping to draw a broad range of students’ attention. The results from the four-year curriculum study on material linking CS and digital audio illustrate that students benefited most from practical projects that they felt were relevant to themselves and others. While the above approaches have reported success, it is unclear how authentic the learning experience is in both target domains. For example, the MediaCom course [4] uses audio at a deficient 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. Therefore, in this research, we aim to go one step further: writing computer programs to create beats for the games that students will develop using EarSketch and Scratch.

3 METHODS

3.1 Pilot Population and Learning Context

Case one of the research is called Mentoring Alabama Girls in Computing (MAGIC) camp which is piloted with 24 middle school and high school students as a computing and music camp at Auburn University. The 1-week pilot took place from June 16th to June 20th, 2019.

The MAGIC curriculum includes EarSketch and Scratch, with the new addition of leadership concepts, is the first curriculum in a three-camp computing pathway - Robotics, App Development and CS for all girls programs in Auburn and surrounding communities. Its curriculum includes some programming and CS principles and modules on topics such as leadership and music remixing. The EarSketch and Scratch pilot served as the programming module of the course.

Many programs have aimed to increase awareness about STEAM education for female students across the globe, such as the "Girls, Music and Computer Science" and "Music Education Meets Computer Science and Engineering Education" [21, 22]. In our efforts to attract underrepresented female students towards future careers in CS and Music, we developed the MAGIC + M program - a curriculum combining music, CS, and leadership. In this project’s context, we believe that Perfomatics [3] is a potential area that may attract students to CS. The main steps of the curriculum are:

- 1) introducing students to CS concepts and skills using Scratch music-related games [4].

- 2) introducing students to music-based storytelling.
- 3) teaching the basics of music programming using EarSketch
- 4) introducing students to basic concepts of leadership
- 5) teaching how to generate their musical composition, and
- 6) providing an opportunity for student presentations in a music composition competition.

The MAGIC teaching methodology is founded upon a project-based, "flipped classroom" learning model, which resembles a studio-based learning approach that has seen practical application in CS curricula [10].

The pilot instructors created a curriculum derived from the openly accessible EarSketch and Scratch online curriculum and leadership concepts.

The students enrolled in the MAGIC pilot did not self-select for a course with EarSketch, Blockly programming, and leadership or even a course in CS. Their interest was in the broader camp program at Auburn. Students did not begin the class with substantial computer and computing experience, as proven by their responses to our engagement survey.

3.2 Curriculum and Teaching Strategies

During the pilot, instructors mostly focused on:

- 1) maximizing student motivation,
- 2) interweaving the teaching of programming, music, and leadership concepts and techniques,
- 3) encouraging creative applications of the acquired knowledge via individual and group projects, and
- 4) benchmarking progress via quizzes.

3.2.1 Maximizing Student Motivation: MAGIC project is a collaborative project between experts in their academic areas. Instructors employed several complementary strategies to motivate students. Students' immediate goal of producing personally expressive, innovative music to share with the class encouraged their fellow peers to learn to code more entertainingly. In particular, the course emphasized how programming (instead of creating music manually in the DAW) enables students to work more efficiently. We approached this by automating repetitive tasks, to more rapidly experiment with different musical possibilities and variations, and to create unique musical structures and sounds to make their music and game stand out. However, grasping the basics of CS concepts and applying them in composing songs at the beginning seemed complicated for students. Therefore, we first familiarized students with block programming using Scratch, where students could define their game story and implement their game with instructors' help. Students showed a great interest in making their own game and were motivated to compose their songs for each component of their game after fully understanding CS concepts and their connection with music. Instructors also emphasized the importance of programming in the music industry's future and the opportunities to become industry leaders by developing new tools for making, performing, and distributing music.

3.2.2 Interweaving Programming and Music: Throughout the pilot, students learned to design and develop musical applications using their newly grasped music, game development, and leadership skills. Days 1 and 2, students described their game story using the computing concepts learned in Scratch to develop their game interfaces. In days 2 and 3, we introduced fundamental music and melody/ rhythm pattern concepts, including tracks, tempo, instrument pairings, and sound effects and filters. In the afternoon of each day, we moved to leadership development sessions, and relevant programming concepts, including variables, function calls, and core EarSketch Application programming interface (API) functions for placing sounds and effects on tracks. The course also addressed sharing both music and code, open licensing models in both domains, and sharing practices on the EarSketch and Scratch social media site. Throughout the course, especially early on, we played popular music to demonstrate specific concepts and motivate students to make their versions.

Fig. 2. Students sharing their projects.

On days 3 and 4, we introduced rhythms and repetitions in music and their relevance to programming concepts, including strings and iteration. Day 5 focused on musical compositions' arrangement as a series of musical sections, with students learning to define their functions to encapsulate and re-use musical sections. At the end of the pilot, students reviewed, and reinforced previous materials learned in the camp. They deepened their understanding of computing concepts, API functions, musical techniques, and individually developed projects that served as final group projects.

3.2.3 Quizzes and Projects: To ensure that students effectively acquired and internalized the MAGIC curriculum, the instructor created two types of assessments: 1) quizzes, which were not multiple-choice exams but instead asked for answers in the form of online Python code; and 2) Individual and group projects that demonstrated integration and application of music, programming concepts, and leadership.

The quizzes asked the students to write Python scripts to make music with specific computational and musical requirements. The requirements included programming techniques (such as variables, constants, strings, loops, functions, and lists) [5] in Scratch and EarSketch API functions (to perform tasks such as playing audio files on the timeline, creating rhythmic beats, and controlling track effects and effect envelopes). For instance, Quiz 1 included writing a Python script that creates a minimum of two tracks of varying lengths, which uses multiple sound files and applies filter level adjustments to the tracks that go alternatively through the sound files. Projects were similar to the quizzes, as they required the development of a code to create music but were more open-ended to provide students with more opportunities for artistic creativity.

Students were asked to take online quizzes during the camp and do individual and group projects where the instructor could use a technique similar to pedagogical code review processes to grade students [10] and engaged in individual discussions with students to peruse each student's code and composition. Individual discussions with students enabled the instructor to detect errors that an individual student was making, such as incorrect API usage or application of iteration, to improve the student's comprehension. At the end of Days 4 and 5, students had their presentations to share their projects with the class and get peer reviews on their projects.

4 SOFTWARE DEVELOPED BY PARTICIPANTS

This section represents two of the most innovative projects developed by the girls who participated in the camp.

4.1 Keyboard Music

Student one developed the Keyboard Music Game. It was a game developed to improve the player's musical perception of rhythmic and melodic patterns. The game has three levels of difficulty: Easy, medium, and challenging. After choosing the difficulty setting, the user needs to follow the rhythm and melody given in a sequence and then play that particular sequence simultaneously (shown in Figure 3). The game's idea was to analyze what the user played on the digital piano and give him visual feedback by showing what was played correctly in green and erroneously in red; however, the last part of the project was not implemented due to the lack of time.

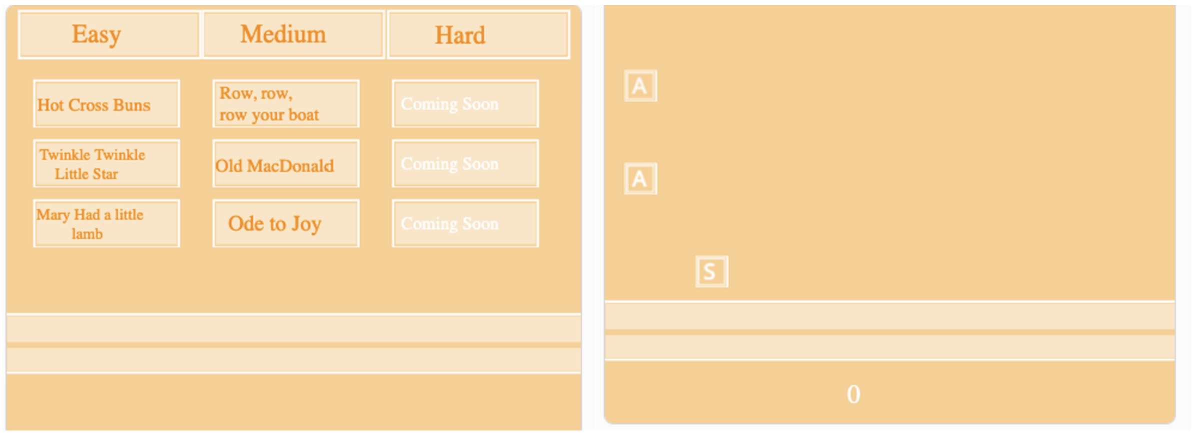


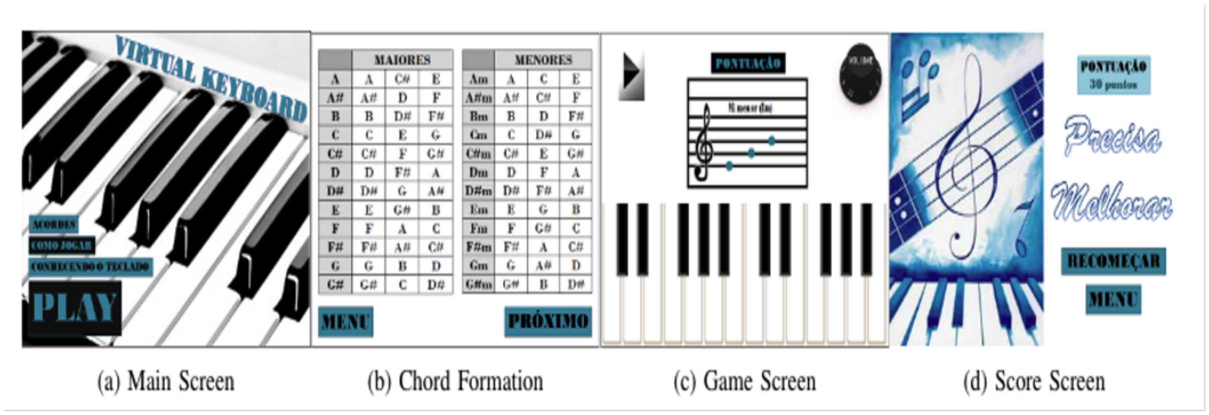
Fig. 3. Student project. Keyboard Music

4.2 Keyboard Chord Trainer

The Keyboard Trainer was developed by student two. The goal of this game was to improve the user's music score reading at first glance, focusing on major, minor and seventh Chords. From the main screen, it is possible to select to look through the student's animated videos explaining how the chords were formed and to play the game. The game shows the user a succession of chord formations that the user needs to execute on the piano simulator. After a given number of chords, the game ends, showing the score screen (Figure 4).

5 EVALUATION

In this pilot study, we developed qualitative and quantitative surveys where students were measured on attitude changes and content knowledge. A student engagement survey was given to students as a pre/post retrospective at the pilot's beginning and end. The approach asked students at the beginning of the camp to consider how they feel about the course and their background in music and computer science concepts. At the end of the camp, we asked students to indicate their perceptions of the course.



	MAIORES		
A	A	C#	E
A#	A#	D	F
B	B	D#	F#
C	C	E	G
C#	C#	F	G#
D	D	F#	A
D#	D#	G	A#
E	E	G#	B
F	F	A	C
F#	F#	A#	C#
G	G	B	D
G#	G#	C	D#

	MENORES		
Am	A	C	E
A#m	A#	C#	F
Bm	B	D	F#
Cm	C	D#	G
C#m	C#	E	G#
Dm	D	F	A
D#m	D#	F#	A#
Em	E	G	B
Fm	F	G#	C
F#m	F#	A	C#
Gm	G	A#	D
G#m	G#	B	D#

Fig. 4. The main screens of the Keyboard Chord Trainer

The instrument used in this project is based on "Engaging Underrepresented Groups in High School Introductory Computing through Computational Remixing with EarSketch" in which we measure six psychosocial constructs: 1) computing confidence (e.g., "Computers can enhance the presentation of my work to a degree which justifies extra effort."); 2) computing enjoyment (e.g., "I like working with computers and programming"); 3) computing importance/perceived usefulness (e.g., "Computers help me to organize my work better"); 4) motivation to succeed (e.g., "I can make the computer do what I want it to do"); 5) identity and belonging in computing (e.g., "I take pride in my computer abilities") and 6) creativity in computing (e.g., "I can be very expressive and creative while doing computing"). The fifteen-question survey was designed to demonstrate these six psychosocial constructs' changes after attending the program activities.

Our analysis includes a paired samples t-test to address significant changes from pretest to post-test.

Our content knowledge assessment (CKA) measures students' understanding of generic computer science concepts. It was administered as a pre/post assessment with five multiple-choice items. It covers specific introductory computing topics - commands, variables, conditionals, loops, debugging, and EarSketch and Scratch functionality.

Overall, the student engagement survey (Table 1) indicate a positive and statistically significant increase in students' attitudes towards computing across all constructs at $p<0.2$. Before the course, female students were statistically significantly low in computing confidence. Additionally, gains in the female "motivation to succeed" shows that the course may be particularly useful in increasing female motivation to persist in computing problems.

Students across all racial/ethnic groups rated the course as being good or excellent. Many open-ended questions specifically addressed the thickly authentic aspects of the learning environment. One student stated, "I favored making beats with Earsketch. I say this because it felt exciting, like a breath of fresh air. Consequently, I could use this information to teach my classmates who make beats. I coherently believe that this would be useful research for them." Another student said, "I loved the part where we got to make our music using EarSketch. We got to know things that people in the music industry do nowadays," Additionally, another articulated, "I like that we used different techniques to do different things on making videos and music". Though not statistically significant, female and minority students rated the course higher than their counterparts, suggesting the addition of music to computer science through EarSketch is specifically significant among underrepresented/underserved groups in computing.

Furthermore, we designed CKAs to determine the progress of students in a more quantitative way. Our assessment results showed statistically significant increases in computing content knowledge from before to

psychosocial_constructs	Pre_survey	Post_survey
leadership_computing_confidence	3.72	3.94
leadership_computing_enjoyment	3.64	4.10
leadership_computing_usefulness/importance	4.02	4.6
motivation_to_succeed	4.16	4.43
identity_and_belonging_in_leadership_computing	3.93	4.29
creativity_in_leadership_computing	4.30	4.50

Table 1. Students Engagement Survey

after the camp. On average, students answered 15% of the items correct at the beginning of the course (pre) and more than 70% correct at the end of the course (post).

5.1 Theory of Change

We are using path analysis as an extension of multiple regression, which determines the relationships between many variables. Therefore, instead of building multiple regression models, the path analysis can analyze the theory of change with a single model. In this paper, we are using three fit indices to check the stability of the model. The Comparative Fit Index (CFI), the Root Mean Square Error of Approximation (RMSEA), and the Standardized Root Mean Square Residual (SRMR). Recommended cut-off values for the fit indices are CFI above .90, RMSEA below .08, and SRMR below .06 [6].

We used path analysis to determine the magnitude of the hypothesized factors in the theory of change model (see Figure 5).

This research mostly focuses on relationships between authenticity and the six student attitude constructs (computing confidence, enjoyment, importance and perceived usefulness, motivation to succeed, identity/belongingness, and personal creativity).

6 DISCUSSION AND FUTURE WORK

These evaluation results suggest that EarSketch's music and computing learning environment effectively teaches introductory computing concepts to high school students in an informal program and that it substantially improves student attitudes towards computing. Some of the data suggest that EarSketch may be particularly useful for traditionally underrepresented groups in computing. Moving forward, we plan to deploy and evaluate the addition of more technical musical concepts on a larger scale and determine the effectiveness of EarSketch and other similar software. To this end, we: a) have made our curriculum available online, b) are developing online instructor training materials and planning instructor training., and c) are collaborating with the music education department at Auburn University to further expand this project to an undergraduate course for music and CS students. We are also beginning to create an entirely web-based version of our application for students with disabilities to learn CS through music. This research intends to lay the foundation so scholars can learn more about why EarSketch has been significant, develop new assessment tools and software enhancements, be

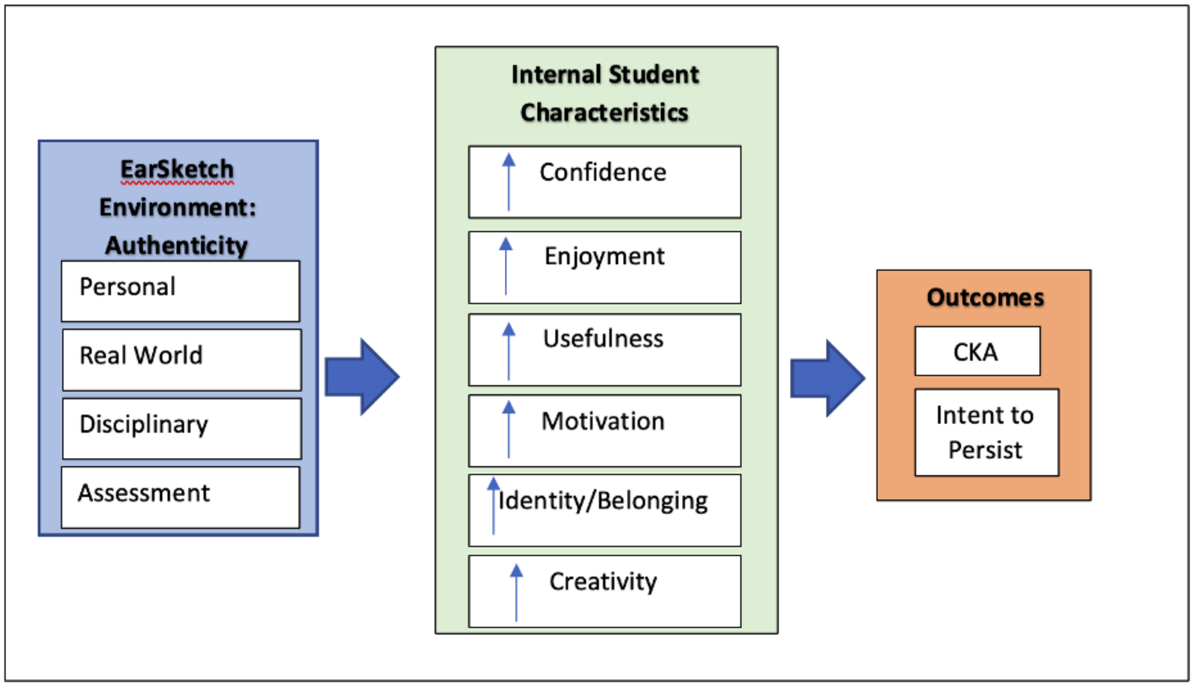


Fig. 5. Theory of change model.

informed about learning sciences, and better understand the specific aspects of EarSketch that have contributed to particular student engagement and content knowledge outcomes. Finally, as an additional part of this project, we are developing an online music learning platform where students will learn music concepts more intuitively.

REFERENCES

[1] Peggy P Apostolou and Maria D Avgerinou. 2021. The Coding Maestros Project: Blending STEAM and Non-STEAM Subjects Through Computational Thinking. In *Handbook of Research on K-12 Blended and Virtual Learning Through the i^{2}Flex Classroom Model*. IGI Global, 504–518.

[2] Jennifer Burg, Jason Romney, and Eric Schwartz. 2013. Computer science" big ideas" play well in digital sound and music. In *Proceeding of the 44th ACM technical symposium on Computer science education*. 663–668.

[3] Jamika D Burge and Tiki L Suarez. 2005. Preliminary analysis of factors affecting women and African Americans in the computing sciences. In *2005 Richard Tapia Celebration of Diversity in Computing Conference*. IEEE, 53–56.

[4] Ron Eglash, Audrey Bennett, Casey O'donnell, Sybillyn Jennings, and Margaret Cintorino. 2006. Culturally situated design tools: Ethnocomputing from field site to classroom. *American anthropologist* 108, 2 (2006), 347–362.

[5] Jason Freeman, Brian Magerko, Tom McKlin, Mike Reilly, Justin Permar, Cameron Summers, and Eric Fruchter. 2014. Engaging underrepresented groups in high school introductory computing through computational remixing with EarSketch. In *Proceedings of the 45th ACM technical symposium on Computer science education*. 85–90.

[6] Mark Guzdial. 2003. A media computation course for non-majors. In *Proceedings of the 8th annual conference on Innovation and technology in computer science education*. 104–108.

[7] Mark Guzdial, David Ranum, Brad Miller, Beth Simon, Barbara Ericson, Samuel A Rebelsky, Janet Davis, Kumar Deepak, and Doug Blank. 2010. Variations on a theme: role of media in motivating computing education. In *Proceedings of the 41st ACM technical symposium on Computer science education*. 66–67.

[8] Jesse M Heines, Gena R Greher, and S Alex Ruthmann. 2012. Techniques at the Intersection of Computing and Music. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*. 372–372.

[9] Jesse M Heines and Daniel A Walzer. 2018. Teaching A Computer To Sing (TACTS): Integrating Computing and Music in a Middle School, After-School Program. *Journal of computing sciences in colleges* 33, 6 (2018).

[10] Christopher D Hundhausen, N Hari Narayanan, and Martha E Crosby. 2008. Exploring studio-based instructional models for computing education. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. 392–396.

[11] Fatemeh Jamshidi and Daniela Marghitu. 2019. Using Music to Foster Engagement in Introductory Computing Courses. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 1278. https://doi.org/10.1145/3287324.3293855

[12] Yasmin Kafai, Chris Proctor, and Debora Lui. 2020. From theory bias to theory dialogue: embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. *ACM Inroads* 11, 1 (2020), 44–53.

[13] Caitlin Kelleher, Randy Pausch, and Sara Kiesler. 2007. Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1455–1464.

[14] Minji Kong. 2020. *THINKING LIKE A COMPUTER: AN EXPLORATORY STUDY OF INTRODUCTORY PROGRAMMERS'LEARNING PROCESSES IN SCRATCH*. Ph.D. Dissertation. University of Delaware.

[15] Brian Magerko, Jason Freeman, Tom McKlin, Scott McCoid, Tom Jenkins, and Elise Livingston. 2013. Tackling engagement in computing with computational music remixing. In *Proceeding of the 44th ACM technical symposium on Computer science education*. 657–662.

[16] Adam L Meyers, Marilyn C Cole, Evan Korth, and Sam Pluta. 2009. Musicomputation: teaching computer science to teenage musicians. In *Proceedings of the seventh ACM conference on Creativity and cognition*. 29–38.

[17] Isabelle Peretz and Robert J Zatorre. 2005. Brain organization for music processing. *Annu. Rev. Psychol.* 56 (2005), 89–114.

[18] Jon Preston and Briana Morrison. 2009. Entertaining education–using games-based and service-oriented learning to improve STEM education. In *Transactions on edutainment III*. Springer, 70–81.

[19] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[20] Alex Ruthmann, Jesse M Heines, Gena R Greher, Paul Laidler, and Charles Saulters. 2010. Teaching computational thinking through musical live coding in scratch. In *Proceedings of the 41st ACM technical symposium on Computer science education*. 351–355.

[21] Carlos N Silla, André L Przybysz, and Wellington V Leal. 2016. Music education meets computer science and engineering education. In *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–7.

[22] Carlos N Silla, André L Przybysz, Andriano Rivolli, Thayna Gimenez, Carolina Barroso, and Jessika Machado. 2018. Girls, music and computer science. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–6.

[23] Sebastien Siva, Tacksoo Im, Tom McKlin, Jason Freeman, and Brian Magerko. 2018. Using music to engage students in an introductory undergraduate programming course for non-majors. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 975–980.