CBIE 2023
CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO
Uma escola para o futuro: Tecnologia e conectividade a serviço da educação

Inspiring Interest in Computing using Music: A Case Study on Students Lacking Prior Music Education

Douglas Lusa Krug[1], Reuel J. Kulibaba[1], Denis A. Linzmeyer, Chrystalla Mouza[2], David C. Shepherd[3]

[1]Instituto Federal do Paraná (IFPR) – Campus União da Vitória
União da Vitória – PR – Brazil

[2]College of Education – University of Illinois Urbana-Champaign
Champaign, IL, USA

[3]School of Electrical Engineering and Computer Science
– Louisiana State University
Baton Rouge, LA, USA

douglas.krug@ifpr.edu.br, cmouza@illinois.edu, dshepherd@lsu.edu

Abstract. *Broadening interest in computer science is a major research goal of the 21st century. Many initiatives use traditional "hooks" to attract students, such as video games and robotics. Unfortunately, this tends to attract only those already interested in computer science. One alternative domain gaining momentum in computer science education research is music, which is showing interesting results with participants that have previous music knowledge. This paper presents a case study of teaching computer programming with music, in Brazil, to students with limited formal music experience. Through data collected in surveys, focus groups, and researchers' observations, we show that in this context students can still learn and thrive as musical programmers.*

1. Introduction

Broadening interest in Computer Science (CS) is a major research goal of the 21st century [Scott et al. 2016], and rightly so given the extreme lack of diversity in the field today [Newsome 2022]. Some approaches use well-known contexts to attract and engage students in CS (e.g., robotics [Miller et al. 2018] and games [Kafai and Burke 2015]). The challenge is that these approaches tend to attract those students already interested in the area, not only failing to address issues of diversity but, in some ways, actually perpetuating them by recruiting from a population already known to lack diversity. Alternatively, an approach that targets a naturally diverse population, those who engage with music, has shown promise in attracting, motivating, and engaging students [Magerko et al. 2013, Freeman et al. 2014, Horn et al. 2020, Lusa Krug et al. 2021] and has the potential to increase diversity in the field.

While countless efforts to broaden CS education exist within high-income and developed nations [Coenraad et al. 2021], few efforts directly target regions outside of North America and Europe. Many of these regions, such as South America and Africa, could benefit the most from these educational initiatives, as there are significant population bases that, as the cost of computers drops, are quickly starting to gain access to the necessary infrastructure to enact innovative technology education programs [Zhang et al. 2021]. Therefore, the CS education community needs to prepare to serve the needs of CS education for *all*, specifically for *all* countries [Apiola et al. 2022].

One clear challenge for CS education research is that approaches may not hold across cultures, and, in some cases, the interventions do not follow a systematic and rigorous research plan [Gutierrez et al. 2018]. When focusing on Brazil, the largest country in South America, one of the challenges of transferring approaches across cultural boundaries quickly becomes clear: There are few schools with computer labs [Branco et al. 2021].

This paper presents a case study using music to teach computer programming to students without previous music education. The study was conducted in Brazil, where formal music education is not generally offered in the primary and secondary education system. The course was offered in partnership with a public school during after-school hours. Students from 6^{th} to 9^{th} grades volunteered to participate in the study. The curriculum and activities were adapted from previous experiences of *Code Beats* conducted in the United States, which had shown encouraging results in attracting, motivating, and engaging students toward CS [Lusa Krug et al. 2021]. The data collected through surveys, focus groups, and direct observation demonstrated that even in settings where music knowledge is limited, using music to teach coding is an opportunity to attract and motivate students to study computer programming.

2. Related Work

2.1. Teaching Coding with Music

The use of music to teach coding has already attracted educators' and researchers' attention. As a result, various platforms were developed, such as EarSketch [Magerko et al. 2013], JythonMusic [Manaris et al. 2016], Sonic Pi [Aaron and Blackwell 2013], and TunePad [Gorson et al. 2017]. Perhaps the primary motivation to teach coding using music is to engage students not yet interested in CS, increasing their motivation and engagement in CS classes. Several studies have shown promising results in improving motivation, confidence, and intention to persist in CS. For example, using EarSketch, Magerko et al. [Magerko et al. 2013] showed results that increased the "Computing Confidence", "Motivation to Succeed in Computing" and "Creativity" in students that participated in their workshop. Using TunePad, Horn et al. [Horn et al. 2020] showed results where students had significant attitudinal gains in interest, self-confidence, enjoyment, and intention to persist in CS. Finally, Lusa Krug et al. [Lusa Krug et al. 2021], using Sonic Pi, reported a statistically significant difference in engagement towards CS, with students able to create their own song at the end of the reported summer camp. Furthermore, in terms of CS knowledge, Freeman et al. [Freeman et al. 2014], using EarSketch, reported that music could effectively teach introductory computing concepts.

When analyzing the prior music experience of students in existing studies, it is evident that most students had some prior knowledge. While some studies do not report this data[Magerko et al. 2013, Magerko et al. 2016, Freeman et al. 2014, Aaron and Blackwell 2013, Horn et al. 2020], they were held in countries, where historically, students have had access to music classes in primary education. The studies that provide this information indicate that students had previous experience with music. For instance, Zhang et al. [Zhang et al. 2022] indicated that 60% of students had prior music knowledge, while Koppe [Köppe 2020] found that 30 to 50 percent of participants in their workshop had previous experience with music. Similarly, Lusa Krug et al. [Lusa Krug et al. 2021] found that 64.7% of participants knew how to read music. Finally, Petrie [Petrie 2022] found that only 8 out of 22 participants were completely new to music and programming.

Further, most of these studies were held in the United States [Magerko et al. 2013, Magerko et al. 2016, Freeman et al. 2014, Zhang et al. 2022, Lusa Krug et al. 2021, Horn et al. 2020], in Europe [Aaron and Blackwell 2013, Köppe 2020] and one study in New Zealand [Petrie 2022]. In South America, and specifically in Brazil, to the best of our knowledge, this is the first study reporting the use of music to teach applied computer programming, writing code effectively.

2.2. Computer Science Education in Brazil - Primary and Secondary Levels

Brazil does not yet have formal CS education at the primary and secondary levels. Except for the career preparation integrated into the high school curriculum offered in some institutions in Brazil [Brasil and da Educação 2016], there is only a reference guideline about implementing CS at the primary and secondary levels [Ribeiro et al. 2019]. The National Learning Standards [1] for primary education refer to computational thinking as a cross-cutting theme related to mathematics and not specifically to CS [Falcão 2021]. Fortunately, the number of studies in CS education at the primary and secondary levels is growing in Brazil, with studies in primary education receiving more attention than studies in secondary education [Santos et al. 2018]. Also, initiatives to broaden diversity in CS are in place [Maciel et al. 2018].

One mapping study of computational thinking and programming in Brazil [Santos et al. 2018] indicated that games (e.g., Scratch) and robotics (e.g., Lego Robots and Arduino) are the most prevalent domains to teach computational thinking and that the least explored domain is media. Unfortunately, these efforts are held back by a lack of access to equipment in Brazilian schools, where students have access to desk computers in only 54% of schools, laptops in 35% of schools, and tablets in 15% of schools [de Informação e Coordenação do Ponto BR 2021]. To overcome this situation, studies often try to use alternatives, such as unplugged activities [Branco et al. 2021] and more affordable equipment [SantClair et al. 2021].

2.3. Music Education in Brazil

Music education in primary and secondary education in Brazil has been historically neglected. When it exists, it is focused primarily on singing and, in many schools, even this has ceased to exist [Hentschke 2013]. Legislators attempted to make music education mandatory in 2008 [2], yet in 2016 [3], this effort was reversed, and music became an optional part of arts education that also contains visual arts, dance, and theatre.

[1] Free translation of *Base Nacional Comum Curricular - BNCC*

[2] http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/l11769.htm

[3] https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2016/lei/l13278.htm

Despite the laws and regulations, some challenges were faced in implementing music as a mandatory school component. For instance, align the universities' formation roadmap with the schools' curriculum [Hentschke 2013]. One example of the art component diversity in primary and secondary education in Brazil is the data reported by Araújo and Lima [de Araújo and Lima 2022], which shows how students responded when asked which activities the art teachers usually developed in the classes. Specifically, 31% of the students said drawing, 22% painting, 31% writing, and only 16% music. In summary, despite the attempts to make music part of primary and secondary education in Brazil, it is clear that music is not present in these levels of education. Therefore, students interested in music, such as music theory, singing, or learning how to play an instrument, must seek opportunities outside of the formal primary and secondary education environment.

3. Background

This paper reports on a study conducted using the approach and curriculum from *Code Beats*. This approach has already been shown to increase motivation, engagement, and interest in CS in North America [Lusa Krug et al. 2021], where most students who attended the course had previous knowledge of music. We ran this same course in Brazil to test its effectiveness in a different cultural setting where most students do not have previous education in music. Here we first introduce the main concepts of *Code Beats*, then describe the adaptations necessary to use this approach in the reported experiment.

3.1. *Code Beats*

Code Beats is an approach that uses music to teach foundational computer programming concepts. It was first implemented in the United States to broaden participation in CS and create interest in populations underrepresented in CS (e.g., African-Americans and Latino-Americans). It uses hip-hop as its primary musical genre, as it is an important part of the culture of these populations. Hip-hop beats usually do not require extensive harmonic knowledge, as the main focus is on the complexity of the rhythm. Thus it is well suited to teaching many fundamental programming concepts.

The *Code Beats* curriculum is built to teach the foundational concepts of computer programming, such as sequencing, constants, variables, lists, and repetitions (see Table 1, column "Programming Concept"). In addition, it includes musical concepts, such as melody, rhythm, and chords (see Table 1, column "Music Concept"). The curriculum combines the concepts from both areas so that learning the musical concept reinforces the computational concept and vice versa.

After learning these concepts, students can apply their learning on hands-on activities that are actual hip-hop songs transcribed into a music-coding platform called TunePad [Gorson et al. 2017]. TunePad uses a computation notebook approach and allows the use of cells to create songs. Each cell is an instrument that can be coded in Python [4] and will play a sound according to the instructions. It is possible to use different instrument types, such as a keyboard, drums, bass, and guitar. Each cell may look and sound simple, but by combining multiple cells, realistic-sounding songs can be coded. And this is how the activities are built in *Code Beats*, a group of individual cells that mimic an actual song. Students receive a project with all but one coding cell complete, and their activity involves editing that cell to complete the song.

[4] https://www.python.org/

3.2. Contextual Adaptations

Some adaptations were necessary to implement *Code Beats* in Brazil. The first is translating all the activities and instructions into Portuguese without changing the activities' objectives and content. The translation occurred only on the instructions, not in the TunePad interface, TunePad documentation, or built-in functions (e.g., *playNote*, *rest*). Also, adaptations were needed in the curriculum, where not all concepts from the original format were taught, and, in some cases, the concept was approached in more than one class. The original curriculum distribution is shown in Table 1, and the curriculum distribution used in the experience reported in this paper is shown in Table 2.

Day	Programming Concept	Music Concept
1	Sequencing	Melody
2	Variables and Constants	System of Musical Notes
3	Functions (with single parameter)	Rhythm
4	Functions (with multiple parameters)	Rhythm and Melody
5	Lists	Chords
6	Repetitions (numeric controlled)	Repetition
7	Repetitions (list iteration)	Repetition
8	Repetitions (nested lists)	Chord progression
9	Modularization	Orchestration
10	Parallelism	Orchestration

Table 1. Curriculum Distribution - Original Version

Day	Programming Concept	Music Concept
1	Sequencing	Melody
2	Sequencing	Melody
3	Variables and Constants	Musical Notes System
4	Functions (with single parameter)	Rhythm
5	Functions (with single parameter)	Rhythm
6	Lists	Chords
7	Lists	Chords
8	Repetitions (numeric controlled)	Rhythm
9	Repetitions (numeric controlled)	Rhythm
10	Modularization	Orchestration

Table 2. Curriculum Distribution - Adapted

Another difference from the original approach is the number of hands-on activities during classes. Due to time constraints of the after-school session and students' music knowledge, students had one short and one long activity in this implementation, instead of two short and one long activities from the original format. While two short activities were prepared for this class, during the first days of the course, it became evident that this group would need significantly more time to complete activities, as students were taking much more time than expected to understand the music requirements for the activity. Thus the second short activity was removed from each day, and, on some days, only one activity was applied, leaving the other for the following class [5].

[5] `https://bit.ly/3oLyMmj`

Another change is in the way the music content was taught. In the original approach, even though connections were made between music and coding in the instructions, in all lessons, the CS instructor taught coding, and the music instructor taught music. In the implementation reported in this paper, it was impossible to have the music instructor in every day of class, so the coding instructor (having witnessed the music instruction many times as part of other classes) gave the music lesson on days the music instructor was absent.

Finally, when implemented in the United States, students composed their own beats (from scratch) at the end of the course to participate in a beat contest. When implemented in Brazil, students did not engage with the beats contest. There were two main reasons for this. First, not all students had computer access outside of class time, and second, students' lack of previous music experience limited their ability to create their own compositions.

4. Research Questions

This study aims to understand what motivates a population without previous music experience to attend a coding course that uses music and the challenges and opportunities that arise from this context. Thus, the research questions being investigated are:

RQ1: What motivated students to attend, continue to attend, and engage in this course?

RQ2: What challenges and opportunities arise from teaching coding using music to a population that lacks a musical background?

5. Methods

5.1. Study Design

This paper reports a case study that took place in a southern Brazilian city, home to around 58,000 residents. The classes were organized in partnership with a public school (with around 350 students from 6th to 9th grades), allowing the researchers to advertise the course to the students and use the school computer lab. The students were divided into two groups due to the limited size of the available computer lab. Both groups had the same content in the same order. Each group consisted of ten classes, over two weeks, with an hour-long class each weekday after school. The classes included computer programming lessons, music theory lessons, hands-on activities, and interactive quizzes. A typical day of classes had two hands-on activities based on an actual hip-hop song transcribed to TunePad. Students had to do their activities in one TunePad cell, which would complement the transcribed music. Students were asked to complete a voluntary pre and post-course survey at the beginning and end of the course. On the last day of the course, students were invited to participate in a focus group session.

5.2. Participants Demographics

A total of 55 students participated in at least one day of classes. From this total, 25 students answered the pre and post-course surveys. They were, on average, 13.5 years old (min 12 - max 15). 44% of the students self-identified as Girls, 40% as Boys, and 16% of the students preferred not to say. 60% of the students self-identified as White, 36%

as Mixed Races[6], and 4% as Black. 52% of the students said they had computer classes in the past. However, for 60% of the students, this course was their first experience with computer programming, and 32% had less than one year of experience in computer programming. 80% of the students did not play a musical instrument, and 88% said they had never had music classes. The students' preferred musical genres are funk (52%), Brazilian country music (44%), pop (40%), rap (40%), rock (36%), and hip-hop (32%).

5.3. Data Collection and Analysis

A pre and post-course survey was administered to document the changes between pre and post-course and to provide data to answer the research question. This survey is the same one used in previous experiences of *Code Beats*, and it is adapted from Mouza et al. [Mouza et al. 2016], inspired by Ericson and McKlin [Ericson and McKlin 2012]. The survey includes 26 questions and asks students to rate their agreement or disagreement using a Likert scale format, from 1 - "Strongly Disagree" to 5 - "Strongly Agree." The questions were grouped into four categories, (a) confidence in CS, with 16 questions (e.g., "I have a lot of self-confidence when it comes to computing."); (b) belonging to CS, with three questions (e.g., "I feel like I "belong" in computer science."); (c) gender equality in CS, with three questions (e.g., "Girls can do just as well as boys in computing."); and (d) students' future in CS, with four questions (e.g., "Someday, I would like to have a career in computing.").

Survey data on students' attitudes toward computing were scored, entered into a spreadsheet, and subsequently exported into statistical software, where means and standard deviations were calculated to assess changes from pre to post-administration. To test the statistical significance of the difference between pre and post-course surveys, the Two Sample t-test was performed when the data were normally distributed and the Wilcoxon Rank Sum test when the data were non-normally distributed. To test the normal distribution, the Shapiro-Wilk test was used.

In addition to survey data, focus groups were conducted to assess student experience. Focus group questions addressed student motivation for attending the course (e.g., "What made you decide to attend this course?"); experience with coding (e.g., "Tell me about your experience coding in TunePad. How did it work?"); potential surprises (e.g., "Did anything surprise you about the course?"); and the role of music in sustaining interest (e.g., "Did the fact that you were making music make you more or less interested in this course?"). Focus group data were analyzed qualitatively to identify emerging themes.

All questions in the survey and the focus group were in Portuguese, the students' native language. The answers were translated into English to perform the analysis and report the results.

This paper's first and third authors acted as instructors during the course. The first author is a CS professor in a Brazilian public institution with experience teaching computer programming to secondary and post-secondary school students. The third author has an art degree with a concentration in music. He teaches music and art classes in Brazil. Their observations during the classes and the main challenges faced during the course are reported as part of the results.

[6]Translation of *Pardo*, multi-racial Brazilians.

6. Results

To answer our two research questions, we broke the findings down into two sections. The first, related to RQ1, deals with student motivation. The second, focused on RQ2, deals with the challenges and surprises encountered when deploying this approach in a new context.

6.1. Highly Engaged and Motivated

Quantitative Survey Data: Student scores on a pre and post-survey were analyzed, and it was found that they started the class with high scores for all measures, and these scores did not change much throughout the session. We believe this speaks to the interest in both computer programming and music present in this community. Table 3 reports each category's pre and post-course responses. The first column has the category. The second and third columns contain the average of responses for pre and post-course surveys, with their standard deviation in parenthesis. The fourth column shows the difference between the pre and post-course survey averages. The last column has the p-value for each category. For all but "Gender Equality in CS," the statistical significance was tested using the Two sample t-test. For the category "Gender Equality in CS," the test performed was the Wilcoxon test. The difference between the pre and post-course survey averages is not statistically significant for any category.

Category	Pre	Post	Diff	p-value
Confidence in CS	4.0 (1.2)	4.2 (1.0)	0.2	0.0627
Belonging in CS	3.0 (1.3)	3.3 (1.3)	0.3	0.0756
Gender Equality in CS	4.7 (0.8)	4.7 (0.8)	0.0	0.1198
Future in CS	3.6 (1.3)	3.7 (1.2)	0.1	0.5999

Table 3. Pre and Post-course Survey

Qualitative Survey Data: This after-school course was entirely voluntary. However, it required students to sign up ahead of time, and each day, students had to return to school to attend, as it was held some time after school ended. When asked what made them decide to attend this course, many students cited an interest in learning music. For instance, S18 answered, *"To fill my free time and improve my music theory knowledge."* S29 stated, *"Because I thought it would be cool, and I always wanted to learn about music."* Finally, S35 answered, *"My interest in music. Learn about music theory."* An interest in learning CS was also a factor. For example, S26 answered, *"Because I'm interested in computer science, and I guess it is cool."* S16 stated, *"Because computer science is one of my interests."* And S31 answered, *"I am interested in computers, and it appeared to be an opportunity to start studying them. The music made it more fun."* S04 answered, *"I was interested in learning more about computer science and music."* demonstrating interest in music and CS.

Students also attended because they thought it would be interesting and fun. For instance, S08 said, *"I decided to attend this course because it was interesting, and my friends would attend too."* that is in line with the answer from S06, *"I decided to attend because it appeared to be fun."* Some students indicated that it was a good use of their free time. For example, S27 answered, *"I thought it would be fun and something to do* *with my free time."* S20 answered, *"I thought it was fun and I would have something to do with my free time."*

Musical Motivation: Students were asked specifically if making music made them more or less interested in this course. The vast majority pointed out that the music was crucial to their interest, motivating them to attend the course. For instance, student S01 said, *"Yes, I would not attend with no music. When the course was advertised, you* [7] *started talking about computer programming, and it did not bring my attention, but when you talked about music, I thought, yes, this I would like to do."* Students S12 and S18 stated, *"If there were no music, I would not be interested in / I would not attend the course."* Some students agreed that the music made the course more interesting, but would attend as well without music. For example, student S06 answered, *"If there were no music, I would attend, but the interest would be lower."* Student S31 stated, *"I would have been interested without music, but the music made it more fun."*

6.2. Challenges and Surprises

Unexpected Pairing: The first surprise we report is the students' surprise. Students were asked if anything surprised them during the course. Many students said they were surprised that they could create music with coding. For instance, student S12 answered, *"The TunePad. I never imagined that it was possible to create music like that."* Student S04 stated, *"The fact that it is possible to create music with coding."* Another student, S06, said, *"When we created the song, even the minimal one, I was impressed."* Additionally, one student was surprised by how the different tracks combined to make a complete song. S32 stated, *"Use all the code together. It is really interesting. I had no idea it was possible."* Finally, one student was surprised by the music instructor's ability. S31 said, *"The music instructor's ability. It is possible to see that he knows what he is doing."*

Musical Competence and Confidence: The biggest challenge we encountered was the lack of musical background. For most students, this course was their first contact with music theory; these students did not have the basic knowledge of music, for instance, musical notes and rhythm. Even though the activities do not require deep knowledge, students had to gain at least a basic understanding to complete the activities successfully. Fortunately, students seemed to be able to pick up musical concepts quickly, and they were engaged when asked to work on their activities. They did especially well with the shorter activities, with more straightforward solutions, with an example to follow and hints that guided them to the correct answer. On the other hand, students struggled with longer, more open-ended activities, seemingly due to a lack of music knowledge or perhaps confidence. When the instructor solved these longer activities with the class, students could answer specific questions about what commands to use and how to use them but struggled to do this alone. Similarly, when students were asked to create their own compositions on the last day of class, they struggled to do so. It seems that their lack of confidence in their own musical skill made them hesitate or even stop when asked to make musical choices, even though they had a good understanding of the necessary programming concepts and commands to implement a solution.

Programming Experience: Another challenge, which did not cause major problems, was the students' relatively little previous experience in programming. Students were

[7] The researcher that advertised the course.

asked to talk about their coding experience during the course. Some students said that it was easy, like S24, who said, *"It was really easy."* One student (S09) who had experience with another programming language said, *"Easy, there is no semicolon."* Another student with previous programming experience said *"It was not hard. The syntax was easy to memorize. The hard part was to know how to use the music. It is easier than Scratch."* On the other hand, some students said it was hard at the beginning of the course, but it became easy with time. For instance, student S14 said, *"In the beginning was hard, and then it started to be easier."* Likewise, student S29 stated, *"In the beginning, it was a little confusing, but now it is easy."* Additionally, student S10 explained what helped him in the course, *"It was a little difficult, but with help from the instructors, it became easier."* Finally, student S36 pointed out that it was hard to memorize the commands, *"It was a little complicated. It was hard to memorize everything. With time, we could get it."*

High Interest: Another surprise was the level of interest in this course, even during recruitment. During recruitment, the first author of this paper visited each class to describe the course to students briefly. The students' reaction was positive, with a strong appreciation of the idea of using music to teach coding. Registration forms were only given to students that explicitly asked for them. 150 students asked for this form, and 55 returned them signed by their parents, registering the student and giving the research consent. The students remained engaged and motivated during classes, paying careful attention to the music and coding lessons.

7. Limitation and Threats to Validity

This study was performed in specific settings. Therefore, its results may not be generalizable in other contexts with populations from different backgrounds. The number of students that attended the course differs from the number of students that answered the pre and post-course surveys and participated in the focus group. Unfortunately, it was impossible to collect the reason for drop-out from students that did so. The interviews and data analysis was conducted by the same researcher who developed and delivered the *Code Beats* program. Therefore, it may represent author bias.

8. Conclusion and Future Work

This paper reports a case study of teaching computer programming with music to students without music experience in a country where, predominantly, music is not part of students' formal education. Based on an approach already tested and validated in a context where most students have previous music knowledge, the original curriculum and activities were adapted to this new setting. It was impossible to see a statistically significant difference between the pre and post-course surveys. However, the pre-course survey answers were already high. According to students' interviews and researchers' observations, the use of music was a key factor in attracting and engaging students in this course, with some students relating that they would not be interested in attending the classes without music.

This case study shows that it is possible to use music to attract students in a setting where music is not part of their formal education. New musical genres can be tested in future work to see if they produce different results. Additionally, a more extensive curriculum with more classes can be tested to cover all the foundational concepts of computer programming and music.

References

Aaron, S. and Blackwell, A. F. (2013). From sonic pi to overtone: Creative musical experiences with domain-specific and functional languages. In *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design*, FARM '13, page 35–46, New York, NY, USA. Association for Computing Machinery.

Apiola, M., Saqr, M., López-Pernas, S., and Tedre, M. (2022). Computing education research compiled: Keyword trends, building blocks, creators, and dissemination. *IEEE Access*, 10:27041–27068.

Branco, A., Dutra, C., Zumpichiatti, D., Campos, F. A., SantClair, G., Mello, J., Moreira, J. a. V., Godinho, J., Marotti, J., and Gomide, J. (2021). Programming for children and teenagers in brazil: A 5-year experience of an outreach project. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, page 411–417, New York, NY, USA. Association for Computing Machinery.

Brasil and da Educação, M. (2016). Catálogo nacional de cursos técnicos.

Coenraad, M., Fofang, B. J., and Weintrop, D. (2021). Gusanos y espheros: Computing with youth in rural el salvador. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, page 404–410, New York, NY, USA. Association for Computing Machinery.

de Araújo, G. C. and Lima, I. F. (2022). Gaps in the training of arts teachers: old challenges and problems in brazilian education. *Arts Education Policy Review*, 123(4):178–193.

de Informação e Coordenação do Ponto BR, N. (2021). Survey on the use of information and communication technologies in brazilian schools : Ict in education 2020.

Ericson, B. and McKlin, T. (2012). Effective and sustainable computing summer camps. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 289–294.

Falcão, T. P. (2021). Computational thinking for all: What does it mean for teacher education in brazil? In *Anais do Simpósio Brasileiro de Educação em Computação*, pages 371–379. SBC.

Freeman, J., Magerko, B., McKlin, T., Reilly, M., Permar, J., Summers, C., and Fruchter, E. (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*, SIGCSE '14, page 85–90, New York, NY, USA. Association for Computing Machinery.

Gorson, J., Patel, N., Beheshti, E., Magerko, B., and Horn, M. (2017). Tunepad: Computational thinking through sound composition. In *Proceedings of the 2017 Conference on Interaction Design and Children*, pages 484–489.

Gutierrez, F. J., Simmonds, J., Casanova, C., Sotomayor, C., and Hitschfeld, N. (2018). Coding or hacking? exploring inaccurate views on computing and computer scientists among k-6 learners in chile. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 993–998, New York, NY, USA. Association for Computing Machinery.

Hentschke, L. (2013). Global policies and local needs of music education in brazil. *Arts Education Policy Review*, 114(3):119–125.

Horn, M., Banerjee, A., West, M., Pinkard, N., Pratt, A., Freeman, J., Magerko, B., and McKlin, T. (2020). Tunepad: Engaging learners at the intersection of music and code.

Kafai, Y. B. and Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist*, 50(4):313–334.

Köppe, C. (2020). Program a hit – using music as motivator for introducing programming concepts. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '20, page 266–272, New York, NY, USA. Association for Computing Machinery.

Lusa Krug, D., Bowman, E., Barnett, T., Pollock, L., and Shepherd, D. (2021). *Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop*, page 397–403. Association for Computing Machinery, New York, NY, USA.

Maciel, C., Bim, S. A., and da Silva Figueiredo, K. (2018). Digital girls program: Disseminating computer science to girls in brazil. In *Proceedings of the 1st International Workshop on Gender Equality in Software Engineering*, GE '18, page 29–32, New York, NY, USA. Association for Computing Machinery.

Magerko, B., Freeman, J., McKlin, T., McCoid, S., Jenkins, T., and Livingston, E. (2013). Tackling engagement in computing with computational music remixing. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, page 657–662, New York, NY, USA. Association for Computing Machinery.

Magerko, B., Freeman, J., Mcklin, T., Reilly, M., Livingston, E., Mccoid, S., and Crews-Brown, A. (2016). Earsketch: A steam-based approach for underrepresented populations in high school computer science education. *ACM Trans. Comput. Educ.*, 16(4).

Manaris, B., Stevens, B., and Brown, A. R. (2016). Jythonmusic: An environment for teaching algorithmic music composition, dynamic coding and musical performativity. *Journal of Music, Technology & Education*, 9(1):33–56.

Miller, K., Sonnert, G., and Sadler, P. (2018). The influence of students' participation in stem competitions on their interest in stem careers. *International Journal of Science Education, Part B*, 8(2):95–114.

Mouza, C., Marzocchi, A., Pan, Y.-C., and Pollock, L. (2016). Development, implementation, and outcomes of an equitable computer science after-school program: Findings from middle-school students. *Journal of Research on Technology in Education*, 48(2):84–104.

Newsome, M. (2022). Computer science has a racism problem: these researchers want to fix it. *Nature*, 610(7932):440–443.

Petrie, C. (2022). Programming music with sonic pi promotes positive attitudes for beginners. *Computers & Education*, 179:104409.

Ribeiro, L., Castro, A., Fröhlich, A. A., Ferraz, C. A. G., Ferreira, C. E., Serey, D., de Angelis Cordeiro, D., Aires, J., Bigolin, N., and Cavalheiro, S. (2019). Diretrizes da sociedade brasileira de computação para o ensino de computação na educação básica. *Sociedade Brasileira de Computação*.

SantClair, G., Godinho, J., and Gomide, J. (2021). Affordable robotics projects in primary schools: A course experience in brazil. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, page 66–72, New York, NY, USA. Association for Computing Machinery.

Santos, P. S. C., Araujo, L. G. J., and Bittencourt, R. A. (2018). A mapping study of computational thinking and programming in brazilian k-12 education. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–8.

Scott, A., Martin, A., McAlear, F., and Madkins, T. C. (2016). Broadening participation in computer science: Existing out-of-school initiatives and a case study. *ACM Inroads*, 7(4):84–90.

Zhang, J., Luxton-Reilly, A., Denny, P., and Whalley, J. (2021). Scientific collaboration network analysis for computing education conferences. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, ITiCSE '21, page 582–588, New York, NY, USA. Association for Computing Machinery.

Zhang, Y., Krug, D. L., Mouza, C., Shepherd, D. C., and Pollock, L. (2022). A case study of middle schoolers' use of computational thinking concepts and practices during coded music composition. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, ITiCSE '22, page 33–39, New York, NY, USA. Association for Computing Machinery.