



Attracting Adults to Computer Programming via Hip Hop

Douglas Lusa Krug
Department of Computer Science
Virginia Commonwealth University
Richmond, VA, USA
Instituto Federal do Paraná - IFPR
União da Vitória, PR, BR
krugdl@vcu.edu

Chrystalla Mouza
College of Education
University of Illinois
Urbana-Champaign
Champaign, IL, USA
cmouza@illinois.edu

W. Monty Jones
Department of Teaching and
Learning, School of Education
Virginia Commonwealth University
Richmond, VA, USA
cmouza@illinois.edu

Taylor Barnett
Department of Music
Virginia Commonwealth University
Richmond, VA, USA
barnettt@vcu.edu

David C. Shepherd
Department of Computer Science
Virginia Commonwealth University
Richmond, VA, USA
shepherd@vcu.edu

ABSTRACT

The demand for qualified computing professionals is high, with thousands of positions remaining unfilled each year. To create more qualified professionals, initiatives to attract and engage students in computer science have been proposed, but they tend to concentrate on primary, secondary (K-12), and post-secondary (college) levels. With many adults looking for better career opportunities, it is surprising that few computer science initiatives focus on attracting adult learners to the field. This paper presents the results of an informal computer programming course that teaches the foundational concepts of computer programming to adults as they program hip-hop beats. This course is designed to attract adult learners that otherwise might have never considered computer programming, building their confidence and skills. We conducted this course online, two nights a week, for five weeks, for about 40 participants. Afterward, we conducted a qualitative analysis of written survey data. We found that the adult learners' perception of computer programming changed during the course, with many participants planning their next step in computing education.

CCS CONCEPTS

• **Social and professional topics** → **Informal education; K-12 education**; • **Applied computing** → **Sound and music computing**.

KEYWORDS

music, TunePad, informal learning

ACM Reference Format:

Douglas Lusa Krug, Chrystalla Mouza, W. Monty Jones, Taylor Barnett, and David C. Shepherd. 2023. Attracting Adults to Computer Programming

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

SIGCSE '23, March 15–18, 2023, Toronto, ON, Canada.

© 2023 Association for Computing Machinery.

ACM ISBN 978-1-4503-9431-4/23/03...\$15.00

<https://doi.org/10.1145/3545945.3569800>

via Hip Hop. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023)*, March 15–18, 2023, Toronto, ON, Canada. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3545945.3569800>

1 INTRODUCTION

Computer programming is becoming a required skill, not only for computer programmers, as many other occupations now require basic programming knowledge [7, 26]. Unfortunately, despite the percentage of adults who are digitally literate [21], the vast majority of modern workers do not have foundational programming skills. To increase the number of workers with programming skills, there is an increasing number of initiatives designed to attract and engage people in computational learning. These initiatives vary and may focus on informal learning [2] or High School classes offered in a formal environment [5, 8]. However, most learn-to-code initiatives and associated research target some of the youngest members of society [12, 25]. Indeed, research examining the teaching-learning process of computer programming for adult learners is, except for expensive and intensive boot camps [6], scarce [30].

Teaching computer programming to adults can empower this immense and fast-growing population, improving their quality of life and financial outlook [12]. In fact, for the subset of adults who truly embrace computer programming, learning these skills can be life-changing. The field has, for years, had a considerable number of open positions that will increase for at least the next ten years. The U.S. Bureau of Labor Statistics [23] projects that the number of CS-related posts will increase by more than 667,000 positions, or 13.4%, from 2020 to 2030.

Unfortunately, it is challenging to attract adults to computer programming. They typically view computer programming as difficult and boring, something that would be impossible for them to learn at this stage of life [7]. Getting these jaded adults to even sign up for a computer programming class requires some convincing. However, if this initial reluctance can be overcome, there is hope, as even a brief, concrete learning experience with computer programming has been shown to significantly impact adults and their attitudes towards programming [7].

To attract people resistant to a particular field, they may need to be introduced to that field indirectly. Referring to the field of mathematics specifically, for instance, Seymour Papert said *“The mathophobia endemic in contemporary culture blocks many people from learning anything they recognize as ‘math,’ although they may have no trouble with mathematical knowledge they do not perceive as such.”* [24] One indirect approach to teaching computer programming, which may help people overcome their hesitation, is through music [9, 14, 19, 20]. If students are programming music, it turns out that much of their focus is on musical creation [11], and they learn the necessary computational concepts almost by accident. When culturally relevant music, such as hip hop, is used, it creates an even more powerful platform for students to learn computational concepts by engaging in the seemingly unrelated but extremely engaging task of creating beats [10, 17].

This paper reports the results of an informal computer programming course that presented foundational computer programming concepts to an audience formed by adult learners, coding hip-hop beats. This course was held virtually for five weeks, with one hour class twice a week. During this course, we investigated the impact of this indirect approach, already known to be effective with young learners, on the engagement and motivation of adult learners.

2 RELATED WORK

To date, most research that creates and investigates ways to motivate and engage students in learning foundational concepts of computer programming targets young learners. This body of work typically analyses the effects of tools and methods of programming among students from primary and secondary educational or, at most, introductory college-level classes. While an increasing number of for-profit coding boot camps target adults, these courses are expensive and require full-time engagement for about twelve weeks [6]. Unfortunately, most adults are either unwilling or unable to dedicate much of their resources to what they perceive as a high-risk endeavor [31]. Furthermore, some boot camp students faced the same informal community boundaries (e.g., race, gender, previous experience, and stereotypes) as in other informal and formal learning environments [32].

Some studies have begun to investigate interventions with adults. For instance, Charters et al. [7] report an experiment that uses an educational video game to provide adult participants with programming experience. In this work, the authors found that, after a positive exposure to programming, through the educational video game, attitudes such as the belief that programming is difficult, boring, and something that they could not learn, changed positively [7].

Using an online survey, Guo [12] investigated older adults’ motivation to learn computer programming and the frustrations they experienced. In terms of motivation, three important categories were found: (a) Age-related Motivations: respondents wanted to make up for missed opportunities during their youth; (b) Enrichment-Related Motivations: for instance, learning programming to implement a specific hobby project idea; and (c) Job-Related Motivations: for instance, respondents wanted to learn programming as continuing education that is relevant to their current job or to improve their future job prospects. Finally, in terms of frustrations, findings were categorized into (a) Age-Related Frustrations: for instance,

cognitive limitations such as bad memory; (b) Pedagogy-related Frustrations: for instance, lack of instructional scaffolding; and (c) Technology-related Frustrations: for instance, debugging syntax and run-time errors.

Krafft, Fraser, and Walkinshaw [16] investigated the effects of using Scratch, a block-based language already known as an approach that motivates young learners with adult learners. They observed a positive effect on students’ self-perception and motivation to continue learning programming.

In a longer study, during a six-month-long course with textual and visual programming languages, Sayago and Bergantiños [30] examined the computer programming learning experience of a group of older adults and active computer users with low levels of formal education and no previous experience with computer programming. By the course’s end, participants learned and understood how to write simple programs.

There are several methods and tools to engage and attract people to computer programming, which vary from unplugged activities [3] to the use of robots [15] to the use of block-based environments [13, 28, 29] to the use of video-games development [27] and the use of music creation [1, 11, 20, 22]. Using music as a motivational tool for teaching computer programming has already attracted attention among educators and researchers, resulting in the development of various platforms, such as EarSketch [20], JythonMusic [22], Sonic Pi [1], and TunePad [11], each one with its musical and coding characteristics.

The use of music to teach code has already demonstrated promise with young learners; Using EarSketch, Magerko et al. [20] showed that the use of this tool increased the “Computing Confidence”, “Motivation to Succeed in Computing” and “Creativity” in students that participated in their workshop. In addition, Freeman et al. [9] showed that using music through EarSketch can effectively teach introductory computing concepts. Similarly, using TunePad, Horn et al. [14] showed significant attitudinal gains in students’ interest, self-confidence, enjoyment, and intention to persist in CS. Finally, using a similar tool called Sonic Pi, Krug et al. [19] reported a statistically significant difference in engagement towards computer science in the context of a summer camp, where students were able to create their own song.

3 CONTEXT

Code Beats uses an engaging and innovative approach to teach foundational computer programming concepts through hip-hop beats. Participants developed hip-hop beats created entirely by code, using the music concepts to introduce the computer programming constructs. In previous work with young learners, this strategy has increased students’ motivation and engagement toward computer programming during a summer camp [19].

As software tooling for this learning experience, we use the music coding platform called TunePad [11]. TunePad is delivered as a website, using a computational notebook approach, with a different code snippet for each track (or instrument) that can be played individually or simultaneously. Each track can be relatively simple, but by combining many tracks, realistic-sounding songs can be composed. TunePad has a user-friendly, interactive interface that offers the chance to experiment, playing notes simply by selecting

an instrument and clicking on a virtual keyboard or set of drums. At the same time, if the user wants to create an actual track, they must write Python ¹ code, as a textual programming language.

We chose hip hop as the musical genre to teach computer programming for two main reasons: (1) hip hop is one of the most popular musical genres, especially for African-Americans and Latino-Americans who are underrepresented in computer science; (2) how a hip-hop beat is created is well suited to teaching computational concepts. Because its creation does not require extensive harmonic knowledge, focusing instead on complex rhythms, it tends to support the more technical concepts present in CS.

We based our approach on the Use-Modify-Create framework [18]. During the course, students worked on coding activities that are actual hip-hop songs transcribed to TunePad (Use). These projects are formed by multiple musical tracks coded in Python, where each track plays one part of the song and, when playing simultaneously, constitutes the whole song. Students were asked to create one track in each activity that would complement the song, usually mimicking the original version (Modify). By the end of the course, students were encouraged to explore their creativity in the final course project, creating their own beats (Create).

4 RESEARCH QUESTION

This study's objective is to analyze this learning experience's influence on adult learners' perceptions of computer programming. In line with that, the research question we are investigating is: How does the use of hip-hop in teaching coding impact adult learners' perceptions of computer programming?

5 METHODS

5.1 Study Design and Data

This paper reports on data collected during a 5-week virtual course. During that period, participants attended streamed 1-hour lessons offered twice a week. The classes were live-streamed using the YouTube ² platform, where the participants could interact with the instructors and each other using the platform's chat. Each class consisted of a mix of interactive live sections and pre-recorded coding and music lessons, with two hands-on activities completed during class and one activity completed after class. Every activity was designed to explore the concept taught that day but could also include previously introduced concepts, adding more complexity to the activities each day. The background to these activities was a realistic-sounding beat based on an actual hip-hop song. College musicians transcribed that into TunePad. At the end of the course, participants engaged in a beats exhibition and contest, where they created their beats to show what they learned. After the course, we asked them through an online survey to answer a set of open-ended questions:

Q1. What did you think about computer programming before *Code Beats*?

Q2. What do you think about computer programming now?

Q3. After *Code Beats*, are you interested in learning more about computer programming? Why or why not?

Q4. After *Code Beats*, are you interested in a career in computer science? Why or why not?

Q5. Did the use of music in *Code Beats* change what you thought about computer programming? If so, how?

5.2 Demographics

To reach participants, this session of *Code Beats* was organized and publicized by Computer CORE ³. This organization prepares under-served adults in Virginia to realize career aspirations with foundational digital and professional skills. During the five-week course, an average of 40 adult learners attended each class. From this group, 32 participants answered the post-survey, which was not mandatory. This population was from a wide range of ages, where the youngest participant was 20 years old, and the oldest participant was 74 years old (average age was 43.8). Of those, 37.5% identified as male, and 62.5% identified as female. In addition, 59.4% were self-declared minorities, and 43.8% were African-American. In addition, 34.4% of the participants knew how to read music, and 18.8% had previously taken a programming class. Hip-hop is the music genre preferred by 31.25% of the participants.

5.3 Data Analysis

To answer the research question, we analyzed the responses from the five questions stated above using a Reflexive Thematic Analysis, also known as Braun & Clarke [4], in an inductive and semantic way, where the coding and theme development reflects and were directed by the content of the data.

First, all the responses were grouped by question in a spreadsheet and were de-identified, where a pseudo-identifier was attributed to each learner. Then, this spreadsheet was shared with the first author of this paper and a research assistant, so each could start creating codes independently. In the first cycle, an open coding process was used, where every answer was analyzed, and one or more codes were attributed when applicable. After that, a new code was created whenever necessary, as determined by the answer's content.

Next, the two independent researchers met to merge and agree upon common codes in the second cycle. At this point, the following actions were taken: (a) For exactly the same codes, the code was maintained; (b) For codes with different words but with the same meaning, the codes were merged; (c) For codes identified by only one of the researchers, the question and the answer were re-analyzed by both, and after discussion, they decided whether the code should be deleted or kept. After finishing this process, the remaining codes were analyzed to identify themes emerging from the codes.

6 RESULTS

In this section, we present the results of the reflexive thematic analysis, separated by question from our survey (Q1 to Q5). For each question, we present the themes from the open coding process and how many times that theme appeared in the data.

¹<https://www.python.org/>

²<https://www.youtube.com/>

³<https://www.computercore.org/>

6.1 Q1. What did you think about computer programming before *Code Beats*?

The responses to this question were grouped in two themes, *Theme 1* (left): *Coding would be difficult/impossible/boring to learn* and *Theme 2* (right): *I was curious about computer programming*. Figure 1 presents all of the codes that contributed to these themes and the number of times each code occurred.

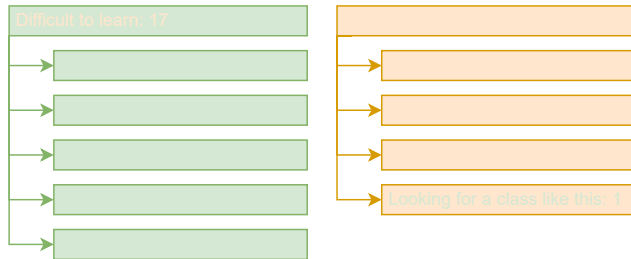


Figure 1: Q1 - What did you think before?

Theme 1: Difficult to learn (n=17) - Participants were initially intimidated by the prospect of learning programming, believing that it was, at best, difficult and sometimes impossible for them to learn. Most adult learners had reservations about their ability to learn programming before taking *Code Beats*, as shown by this group of codes that appeared 17 times in the responses for Q1. Participants' responses included: "I thought [coding] was something **impossible** and that I could never do it" (S12). "I thought [coding] was **difficult and boring**" (S21). "[Coding] was **complicated** and beyond my ability" (S24). "I thought [coding] was fairly **difficult to learn**. It also seemed like it would involve a lot terminology I wouldn't understand and prerequisite skills." (S29).

Theme 2: Curious about coding (n=6) - While most adults were intimidated by computer programming, a few adult learners were curious about computer science. For many of these curious adult learners, the fact that *Code Beats* was centered around music seemed to make it less intimidating, or at least potentially fun, to them, with relevant responses appearing a total of 6 times for Q1. Participants said: "Had no idea how to do it. Just sounded like a **cool thing to check out**" (S2). "I did not understand how to code, and make music out of Python. I am a music producer, and would **like to better understand** computers, so I figured that this is the best fit for me" (S28). "...then I saw my child to simple coding games and toys and programs and *Code Beats* seemed like a fun way to **try it out**. I really enjoyed it and my big break through moment was when I realized that the errors were my friends and showed me exactly what to change. Before that I saw it as me getting it wrong. That little shift in mindset made it so much more fun to **try out things**" (S12).

6.2 Q2. What do you think about computer programming now?

The responses to this question were grouped into four themes. The codes are shown in Figure 2, from left to right, and from top to bottom: "I'm confident that I can learn computer programming" (10); "I've improved my knowledge about computer programming"

(9); "I'm interested in learn more and use it in the future" (9) and "Programming classes are not for me / it is difficult to learn" (5).

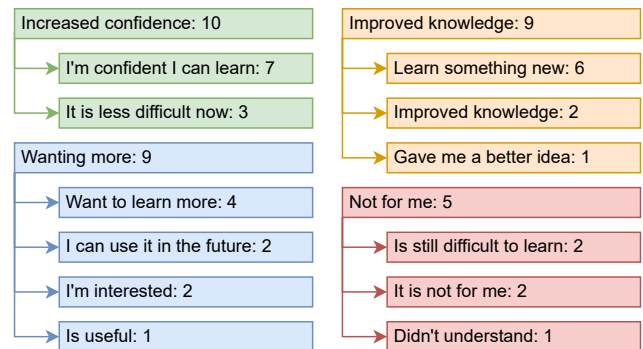


Figure 2: Q2 - What do you think now?

Theme 3: Increased confidence (n=10) - It seems that taking the class increased the learners' confidence and helped them to understand that, while it might take practice, computer programming was a skill they could learn. This theme emerged from a group of codes that appeared 10 times in the responses for Q2. Some examples of responses that we received for this question and were grouped in this theme are: "I think if I put the time to learn into it, **I can do it**" (S7). "**I think it is possible to get good at it** with lots of practice. I think I've found my inner geek!" (S12). "I think that **I am capable of doing computer programming**. It doesn't seem as hard as 1 would think and *Code Beats* taught me that!" (S19). "It's still a bit complicated but seems **less daunting**. I think **the basics are doable** with continual practice" (S29).

Theme 4: Improved knowledge (n=9) - This theme emerged from a group of codes that appeared 9 times in the responses for Q2. Participants said: "I see **how to use it better** and how it applies to so many things. I loved using TunePad to make a song" (S2). "Honestly I don't see Myself as an Older Job Seeker being a Computer Programmer. The Course Has **Added a More In depth layer to my core skill set**. The Easier YouTube Platform makes For Much Easier Access" (S16). "I can understand and **know the vocabulary of coding** which helps in my next computer coding class" (S18). They used many growth phrases (e.g., "use it better", "added to my skill set", etc.) to describe how they felt the class had changed their knowledge of coding.

Theme 5: Wanting more (n=9) - Adult learners who attended this class may have increased their interest in pursuing further education and even employment in the field. This theme emerged from a group of codes that appeared 9 times in the responses for Q2. Some examples of responses that we received for this question and were grouped in this theme are: "I might be **interested in it now**. In fact I might also **think about pursuing a career in it**" (S1). S20: "**Excited to do more** and do my own job as a *Code Beats* master". (S20). "**I would like to get certified** in Python and A+. I would like to also get Security+ ,and Network+" (S28).

Related to this theme, while we could not track all the participants after the class, we know 18 participants who followed up this class by attending a Python class and at least one that gained employment in the field.

Theme 6: Not for me ($n=5$) - As expected, a few adult learners decided that computer programming was not for them. This theme emerged from a group of codes that appeared 5 times in the responses for Q2. Some examples of responses that we received for this question and were grouped in this theme are: “**Confused**. Programming as I saw in *Code Beats* is very different from, say, Data Structures in C. But it seems to be fun.” (S27). “**Not in the computer programming class.**” (S32).

6.3 Q3. After *Code Beats*, are you interested in learning more about computer programming? Why or why not?

The responses to this question were grouped into two themes: “I would like to learn more about computer programming. I can do this.” and “I’m not interested in computer programming.”. The codes and their counting is presented in Figure 3.

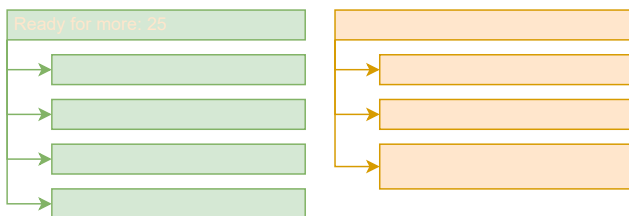


Figure 3: Q3 - Interested in learning more?

Theme 7: Ready for more ($n=25$) - Many adult learners were interested in programming and optimistic about their ability to learn it, with responses from this theme appearing 25 times for Q3. Participants said: “Yes. I think **I could enjoy having a job doing computer programming**” (S7). “**Yes very eager**” (S15). “Yes, **I was very much interested in learning more**. I thought the *Code Beats* class was fun and it was an introduction to Coding. So, it makes me **feel confident** that I could take on a Computer Programming course” (S19). “Yes **I’m more interested in coding** because I see there is a different way to learn the skills needed” (S21). “**I feel comfortable moving forward** to learn more about coding. I was very nervous at first, but not anymore” (S30). These learners expressed both a desire to learn more (e.g., eager, very much interested, more interested) and a growth in confidence (e.g., confident, not [nervous] anymore, comfortable).

Theme 8: Not interested ($n=4$) - After taking the class, a few adult learners decided they were not interested in computer programming. For these adult learners, the class did not focus enough on motivating the importance of computer science, or they failed to learn enough to gain confidence in their ability to code. This theme emerged from a group of codes that appeared 4 times in the responses for Q3. Participants said: “Maybe. It seems important but I haven’t seen a need for it yet in my career. Also **I don’t feel I am motivated** at this time to learn it” (S29). “**Not interested** because I don’t know how to code” (S32).

6.4 Q4. After *Code Beats*, are you interested in a career in computer science? Why or why not?

The responses to this question were grouped into two themes: “I can use computer programming in the future” (13) and “I’m not interested in computer programming” (4), presented in Figure 4.

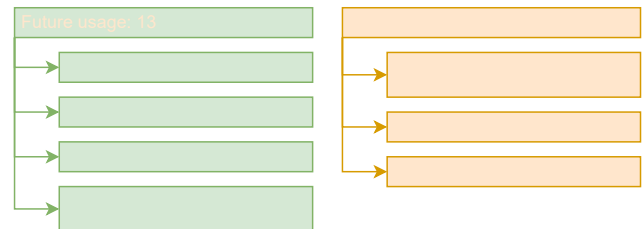


Figure 4: Q4 - Interested in a career?

Theme 9: Future usage ($n=13$) - Many adult learners believed they could and would use computer programming in the future. During completing our class, participants’ confidence in their ability to program a computer grew enough that they could imagine working in the field. This theme emerged from a group of codes that appeared 13 times in the responses for Q4. Some examples of responses that we received for this question were: “Maybe, if the coding can be used in a fun way & in a company that was creative. I am not exactly sure what the jobs would entail if I did it as a career. If I got involved with a game company, **I am sure I would love it** and learn fast” (S2). “Yes. I have a disability that makes me rely on jobs at a desk. **This could be a great fit**” (S7). “Yes. I want to do something different, **learn new technology and get paid more money**” (S10). “Yes, I love working on computers! I think that **coding is in my future** and that I am more than capable of getting the job done!” (S19). “After *Code Beats*, **I am interested in a career in computer science**, because it has so many applications, it is rampant everywhere, and very useful. I especially like how much computers can change our lives, make things more easy for us, and do things faster than without any technology” (S22).

Theme 10: Not enthusiastic ($n=4$) - A few adult learners were not enthusiastic about programming. These participants’ lack of enthusiasm stemmed from a lack of interest and, in some cases, was influenced by difficulties that the adult learner had in understanding the class materials. This theme emerged from a group of codes that appeared 4 times in the responses for Q4. Participants said: “Probably not because **I don’t see myself doing anything with computer software**. I only really understand programming” (S1). “**No not interested in a career in this field**. I was so lost in this class, by trying to program music” (S32).

6.5 Q5. Did the use of music in *Code Beats* change what you thought about computer programming? If so, how?

The responses to this question were grouped into two themes: “The use of hip hop made it easier / fun to learn.” (25) and “The use of hip hop had no affect on learning” (3), presented in Figure 5.

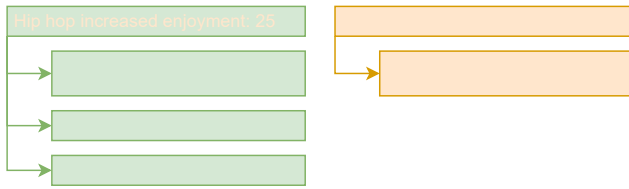


Figure 5: Q5 - Did the use of music change your attitude?

Theme 11: Hip hop increased enjoyment (n=25) - Almost all adult learners thought using hip-hop increased their class enjoyment. Hip-hop made the class more engaging, even for adult learners who struggled to learn the material. This theme emerged from a group of codes that appeared 25 times in the responses for Q5. Participants said: “Yes, it made it easy. You were able to create a finished product and you could learn from your mistakes or see on the keyboard what sounded good and try to fix the code to match it. I have no music background so it was challenging, but **I enjoyed it** and it drove me to figure out how to do it better” (S2). “Definitely! **It was instant gratification.** Basically anyone could do coding the way it was taught through music with *Code Beats*. You didn’t have to be musical but if you were, it helped. With the tracks laid down already, **it was fun to write** and change the code to make different sounds and to learn coding terms...loops, variables, lists and more” (S12). “Yes it did, I thought was a **fun and interactive way** to learn coding” (S13). “Yes, I always thought that computer programming was hard. The code itself makes no sense just by looking at it, however with *Code Beats* I understand that the coding has meaning and is formatted to give instructions to complete a task” (S19). “Absolutely. That was the best, **most fun part** (including *Dave’s dancing*). Even though I didn’t get most of the material, **it was refreshing** to see that you guys are normal and have a sense of humor. Very different from old fashioned programming in UNIX/C or FORTRAN” (S27).

Theme 12: Hip hop had no affect(n=3) - Only a few adult learners did not think the use of hip-hop affected the class. This theme emerged from a group of codes that appeared 3 times in the responses for Q5. Some examples of responses that we received for this question and were grouped in this theme are: “**No. It didn’t**” (S10). “**No, I just know the behind the scenes better!**” (S18).

7 DISCUSSION

7.1 Perception about Computer Programming - Before *Code Beats*

Before *Code Beats*, most adult learners perceived coding as difficult, boring, and something they could never do. This is consistent with past findings; previous studies found that adult learners, prior to exposure to computer programming, thought that programming was difficult, boring, and something that they could not learn [7]. It seems that adult learners are unlikely to consider computer programming as a career option without external encouragement. Furthermore, providing an initial, positive programming experience may be crucial to whether they will ever consider studying CS.

Despite this initial reluctance, adult learners sometimes changed their minds when they realized they could learn programming by

making music. Participants mentioned that *Code Beats* “seemed like a fun way to try it out” and “sounded like a cool thing to check out”, and that *Code Beats* “[was] the best fit for me” due to its focus on music. The focus on the music seemingly provided adult learners with additional motivation; they seemed to think that even though they would understand the coding, at least they would be working with music they like. Furthermore, it seemed to make an even greater difference for adults with musical talent. It made them more comfortable as they knew and understood the musical concepts, making the computational concepts less daunting. Overall, data suggest that associating programming with music is a promising way to attract adult learners.

7.2 Perception about Computer Programming - After *Code Beats*

After participation in the program, adult learners’ perception of computer programming seemed to be much more positive. They were more confident in their ability to program, were more likely to consider furthering their programming knowledge, and some even thought that *Code Beats* made coding easy. We believe that at least part of this change in perception was because we taught computational concepts by relating them to familiar musical concepts. It seems that this technique, which led to a shift in motivation for young learners [19], may also benefit adult learners.

8 LIMITATIONS

This paper report results from a virtual course where all participants volunteered for the experience and had access to it. Hence, they showed some prior interest and may not represent all adult learners. Furthermore, the surveys were limited to five questions instead of more in-depth interviews. Additionally, the survey questions were asked only after the camp. Hence, responses to questions about perceptions prior to the course demonstrate students’ reflections about how they might have thought before the course.

9 CONCLUSION

This paper presented the use of music to teach the foundational concepts of computer programming to adult learners during a five weeks course. The activities in this course used actual hip-hop songs transcribed to a programming platform, where participants could modify a part of the song to apply the computer programming concepts introduced during the classes.

By the end of the course, we documented changes in adult learners’ perception of computer programming, which they had previously perceived as boring, difficult, and impossible to learn. After *Code Beats*, they started to think of computer programming as something they could learn and do, demonstrating an interest in looking for more opportunities to learn these important skills. Finally, using hip-hop to teach computer programming was found useful, motivating adult learners and making the learning process enjoyable.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2048792 and 2048793.

REFERENCES

- [1] Samuel Aaron and Alan F. Blackwell. 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* (Boston, Massachusetts, USA) (FARM '13). Association for Computing Machinery, New York, NY, USA, 35–46. <https://doi.org/10.1145/2505341.2505346>
- [2] Andrew Begel and Amy J. Ko. 2019. *Learning Outside the Classroom*. Cambridge University Press, 749–772. <https://doi.org/10.1017/9781108654555.027>
- [3] Tim Bell and Jan Vahrenhold. 2018. *CS Unplugged—How Is It Used, and Does It Work?* Springer International Publishing, Cham, 497–521. https://doi.org/10.1007/978-3-319-98355-4_29
- [4] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. <https://doi.org/10.1191/1478088706qp063oa>
- [5] Heidi Burgiel, Philip M Sadler, and Gerhard Sonnent. 2020. The association of high school computer science content and pedagogy with students' success in college computer science. *ACM Transactions on Computing Education (TOCE)* 20, 2 (2020), 1–21.
- [6] Quinn Burke, Cinamon Bailey, Louise Ann Lyon, and Emily Green. 2018. Understanding the Software Development Industry's Perspective on Coding Boot Camps versus Traditional 4-Year Colleges. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) (SIGCSE '18). Association for Computing Machinery, New York, NY, USA, 503–508. <https://doi.org/10.1145/3159450.3159485>
- [7] Polina Charters, Michael J. Lee, Amy J. Ko, and Dastyni Loksa. 2014. Challenging Stereotypes and Changing Attitudes: The Effect of a Brief Programming Encounter on Adults' Attitudes toward Programming. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, Georgia, USA) (SIGCSE '14). Association for Computing Machinery, New York, NY, USA, 653–658. <https://doi.org/10.1145/2538862.2538938>
- [8] Code.org and CSTA. 2020. 2020 State of Computer Science Education - Illuminating Disparities. *code.org* (2020). https://advocacy.code.org/2020_state_of_cs.pdf
- [9] 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* (Atlanta, Georgia, USA) (SIGCSE '14). Association for Computing Machinery, New York, NY, USA, 85–90. <https://doi.org/10.1145/2538862.2538906>
- [10] Geneva Gay. 2013. Teaching to and through cultural diversity. *Curriculum inquiry* 43, 1 (2013), 48–70.
- [11] Jamie Gorson, Nikita Patel, Elham Beheshti, Brian Magerko, and Michael Horn. 2017. TunePad: Computational thinking through sound composition. In *Proceedings of the 2017 Conference on Interaction Design and Children*. 484–489.
- [12] Philip J. Guo. 2017. Older Adults Learning Computer Programming: Motivations, Frustrations, and Design Opportunities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 7070–7083. <https://doi.org/10.1145/3025453.3025945>
- [13] Brian Harvey. 2010. Bringing "No Ceiling" to Scratch: Can One Language Serve Kids and Computer Scientists?
- [14] Michael Horn, Amartya Banerjee, Melanie West, Nichole Pinkard, Amy Pratt, Jason Freeman, Brian Magerko, and Tom McKlin. 2020. TunePad: Engaging learners at the intersection of music and code. (2020).
- [15] Jennifer S. Kay and Janet G. Moss. 2012. Using robots to teach programming to K-12 teachers. In *2012 Frontiers in Education Conference Proceedings*. 1–6. <https://doi.org/10.1109/FIE.2012.6462375>
- [16] Maren Krafft, Gordon Fraser, and Neil Walkinshaw. 2020. Motivating Adult Learners by Introducing Programming Concepts with Scratch. In *Proceedings of the 4th European Conference on Software Engineering Education* (Seon/Bavaria, Germany) (ECSEE '20). Association for Computing Machinery, New York, NY, USA, 22–26. <https://doi.org/10.1145/3396802.3396818>
- [17] Gloria Ladson-Billings. 1995. Toward a theory of culturally relevant pedagogy. *American educational research journal* 32, 3 (1995), 465–491.
- [18] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. Computational Thinking for Youth in Practice. *ACM Inroads* 2, 1 (Feb. 2011), 32–37. <https://doi.org/10.1145/1929887.1929902>
- [19] Douglas Lusa Krug, Edtuan Bowman, Taylor Barnett, Lori Pollock, and David Shepherd. 2021. *Code Beats: A Virtual Camp for Middle Schoolers Coding Hip Hop*. Association for Computing Machinery, New York, NY, USA, 397–403. <https://doi.org/10.1145/3408877.3432424>
- [20] Brian Magerko, Jason Freeman, Tom McKlin, Scott McCoid, Tom Jenkins, and Elise Livingston. 2013. Tackling Engagement in Computing with Computational Music Remixing. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) (SIGCSE '13). Association for Computing Machinery, New York, NY, USA, 657–662. <https://doi.org/10.1145/2445196.2445390>
- [21] Saida Mamedova, Emily Pawlowski, and Lisa Hudson. 2018. A description of US adults who are not digitally literate. *Statistics in Brief* (2018).
- [22] Bill Manaris, Blake Stevens, and Andrew R Brown. 2016. JythonMusic: An environment for teaching algorithmic music composition, dynamic coding and musical performativity. *Journal of Music, Technology & Education* 9, 1 (2016), 33–56.
- [23] United States Bureau of Labor Statistics. 2020. Occupational projections and worker characteristics. Retrieved October 04, 2021 from <https://www.bls.gov/emp/tables/occupational-projections-and-characteristics.htm>
- [24] Seymour A Papert. 2020. *Mindstorms: Children, computers, and powerful ideas*. Basic books.
- [25] Joslenne Peña, Benjamin V. Hanrahan, Mary Beth Rosson, and Carmen Cole. 2021. After-Hours Learning: Workshops for Professional Women to Learn Web Development. *ACM Trans. Comput. Educ.* 21, 2, Article 15 (mar 2021), 31 pages. <https://doi.org/10.1145/3446964>
- [26] Gang Peng. 2017. Do computer skills affect worker employment? An empirical study from CPS surveys. *Computers in Human Behavior* 74 (2017), 26–34. <https://doi.org/10.1016/j.chb.2017.04.013>
- [27] Alexander Repenning, Corrina Smith, Robert Owen, and Nadia Repenning. 2012. AgentCubes: Enabling 3D Creativity by Addressing Cognitive and Affective Programming Challenges. In *Proceedings of EdMedia + Innovate Learning 2012*, Tel Amiel and Brent Wilson (Eds.). Association for the Advancement of Computing in Education (AACE), Denver, Colorado, USA, 2762–2771. <https://www.learnedtechlib.org/p/41159>
- [28] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: Programming for All. *Commun. ACM* 52, 11 (Nov. 2009), 60–67. <https://doi.org/10.1145/1592761.1592779>
- [29] Bernat Romagosa i Carrasquer. 2019. *The Snap! Programming System*. Springer International Publishing, Cham, 1–10. https://doi.org/10.1007/978-3-319-60013-0_28-2
- [30] Sergio Sayago and Ángel Bergantiños. 2021. Exploring the first experiences of computer programming of older people with low levels of formal education: A participant observational case study. *International Journal of Human-Computer Studies* 148 (2021), 102577. <https://doi.org/10.1016/j.ijhcs.2020.102577>
- [31] Sherry Seibel and Nanette Veilleux. 2019. Factors influencing women entering the software development field through coding bootcamps vs. computer science bachelor's degrees. *Journal of Computing Sciences in Colleges* 34, 6 (2019), 84–96.
- [32] Kyle Thayer and Amy J. Ko. 2017. Barriers Faced by Coding Bootcamp Students. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (Tacoma, Washington, USA) (ICER '17). Association for Computing Machinery, New York, NY, USA, 245–253. <https://doi.org/10.1145/3105726.3106176>