The Use of DJing Tasks as a Pedagogical Bridge to Learning Data Structures

David James
Computer and Information Sciences Department
Spelman College
Atlanta,GA
djames18@spelman.edu

ABSTRACT

Computing is a field that needs to make more strides to have workforce that matches the racial and ethnic diversity found in the population of the United States. The author of the study aims to address this problem, by using course materials that incorporate the culture of students taking computer science courses. Geneva Gay [5] mentions in her theory of culturally responsive pedagogy that it's critical for instructors to embed the students' culture in their course materials, so students can use their own experiences to understand concepts. The study explores how DJing tasks (such as creating, editing and iterating through playlists) can be used to teach students about data structures. Participants in the study were drawn from a convenience sample in a 15-week sophomore course in data structures at a small liberal arts college. 16 of the 24 students fully participated in the study, and all of the respondents were Black women pursuing a major or a minor in computer science. The students were taught with examples that used Black music as a context throughout the course, and were tasked with creating software for a drum machine or a DJ controller as a final project. Study participants were asked via a web survey to compare example problems presented in a music based context to examples they had seen in the course text. Students were also asked to reflect on the experiences they had engaging in the final project.

CCS CONCEPTS

• Social and professional topics~Computing education

KEYWORDS

STEAM, culturally responsive pedagogy

ITiCSE '20, June 15–19, 2020, Trondheim, Norway
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6874-2/20/06…$15.00
https://doi.org/10.1145/3341525.3387427

ACM Reference format:

David James. 2020. The use of DJIng tasks as a Pedagogical Bridge to Learning Data Structures. In *Proceedings of ACM ITiCSE conference (ITiCSE '20)., June 15-19, 2020, Trondheim, Norway. ACM, New York, NY, 5 pages.* https://doi.org/10.1145/3341525.3387427

Introduction

Culturally relevant and responsive pedagogies and their use within the field of computer science is not a new phenomenon. Scott [8] offers a literature review of studies where authors have used these teaching techniques to educate their students. Instructors have used a variety of activities and curricular materials to ensure that their students can use their own experience to learn course content. Instructors have used contexts as varied as textile making and needlework [6, 9],visual programming in scratch [1–3] and music making [4] to teach concepts in computer science and programming. However many of these studies take place in K-12 schools or after school programs. Investigators have published some studies that take place in undergraduate classes, but mainly use music making as a tool to teach programming in introductory courses [7, 10]. The purpose of this study is to explore how Black music and music making hardware can be used as tools to teach concepts in a data structures course.

Methods

The author sent a web survey to students during week 12 of the course to ask for their feed back on the use of Black music in the course. Black music is defined in this study as any genre of music created by people from the African Diaspora, such as Hip Hop, R&B, Jazz, Soul, Blues, Gospel etc. All 24 students answered the survey which had the following questions:

- Talk about your experience working on problems/examples that use Black music as a context versus problems/examples you've seen in the course text.
- Would you like to have more or less problems that use Black music used as context in future classes? Why?
- Would you like to do projects or problems in the future that incorporate music hardware (e.g.: turntables, DJ controllers and drum machines)? Why?

During the final week of the course (week 15) the author sent another web survey asking students to respond to the following prompt:

- Please reflect on your experience using music hardware and Black Music for the final project.

16 of the 24 students responded to the final survey and submitted reflections on their projects.

Description of the course and tools

Our data structures class serves two main purposes. We aim to teach students the weaknesses and strengths of linear data structures while introducing students to the C++ programming language. We are not expecting students to be "fluent" in the language, but to be comfortable enough to be independently learn features of the language not covered in the course.

The Rhythm and Bytes API

Given the aims of the course it was a struggle to find a C++ audio API that would be appropriate for students who are new to the language. Audio API's can also be quite heavy with respect to overhead, in configuring parameters to control the source(s) and quality of audio for your project. The author created the Rhythm and Bytes API to give students a tool to play prerecorded audio, and manipulate audio with sound effects, by creating instances of objects, and making method calls using those objects. The final project also gave students the option to create software for a drum machine or a DJ Controller. Using the API students could also connect with music hardware via MIDI (Musical Instrument Digital Interface), a standard music data protocol, and receive data from musical interfaces on mobile phones and other devices via Open Sound Control (OSC).

A DJ controller is a piece of hardware that has at least the following features: two turntables, a crossfader, pitch faders, bass, middle and treble controls. The turntables allow the DJ to scratch the audio, the crossfader allows the user to control how much of each turntable partygoers are able to hear. Pushing the cross fader all the way to the left, makes it so that you hear 100% of the audio from the left turntable and none on the right. Pushing the crossfader all the way to the right, puts 100% of the audio from the right turntable through the speakers and 0% of the audio from the left. Placing the crossfader in the middle allows you hear the audio from the left and right turntables at 50% volume. Pitch faders control how fast or slow a piece of audio is playing. Bass, middle, treble controls lets users shape how loud the high, low and middle frequencies are in the music. Drum machines have buttons called pads that allow you to push a button, and play a prerecorded sound of your choice. The loudness of the sound is determined by the force with which you press the button. Drum machines also allow program a sequence of sounds that are triggered with one button press. Hip Hop and R&B music producers often use drum machines(e.g.: Akai MPC, Roland SP404) to create new pieces of music with prerecorded samples, and new recordings of musical performances.

Modeling physical instruments with algorithms

One of the benefits of creating software for a DJ controller or drum machine is that students have to devise many small models and algorithms to make their controllers work as a producer or DJ would expect them to. Students had to think about how they could use methods in the API to simulate scratching a record, or make the song play faster when a user slides the pitch control. Once they got a demonstration of how the controls worked, students had to create algorithms that transformed the data from the controllers to data values that made sense for the API methods that manipulate audio.

Before students used their controllers, they learned course concepts through examples that played or remixed Black Music throughout the semester. The following section will detail many of the examples that were used throughout the course, before students were asked to reflect on their learning in the survey

Music-based examples used to teach course concepts

Classes and Arrays

One of the first activities students do in the course is building a Song class. I tell the students to open up Spotify, Apple Music or Pandora and write down all of the pieces of information they see that describe a song. Then the students are asked to develop a class in Python that can store that data. After covering the syntax to build a class in C++ the students are asked to convert their class to C++.

After this assignment, students are asked to create a playlist of their song object using an array. As a part of the assignment students have to develop a smart shuffle function that ensures that songs aren't played twice. Many of the students used the set data structure to enforce the no repeat rule. While picking random songs is a strength of an array, inserting or deleting a song at the front of a playlist is a weakness. This provided a nice segue to talk about the structure of linked lists making it unnecessary to shift the remainder of the items in the container after a delete or an insert.

Linked Lists, Audio and Inheritance

Students were asked to create a linked list class, that will allow them to insert or delete songs at any given position in the playlist. We also talked about how doubly linked lists simplify the process of insertions and deletions, and how the tail pointer makes it more efficient to append song to the end of your list. The students were asked to create a function that selects random songs from their lists, to illustrate its relative lack of efficiency at accessing elements when compared to arrays.

After learning about pointers, memory and linked lists in C++ the students are introduced to the Sample object in the Rhythm in Bytes (RnB) API, which allowed them to load, play and manipulate audio from a variety of sound files (WAV and MP3 among others). The Sample class has all the data to play and manipulate an audio file, but has none of the descriptive data that students capture in the Song class. To introduce the topic of inheritance, students give their song classes the ability to play audio by inheriting from the sample object. The students then create an application that allows them to create real playlists that they can modify and shuffle.

Stacks and Queues

Next we cover the uses of stacks and queues in the course, using musical examples. First we talk through stacks by using a physical stack of records that a DJ would use to play songs on a turntable. I would demonstrate the DJ popping a record off of a stack and onto a turntable to be played, emphasizing that DJ's do not replay records. Since DJ's don't replay records the album popped off the stack is no longer available in the data structure. The pile of records that remain as a visual history of the songs that have already been played. I explain the queue as the data structure that would be used to preserve the order of people waiting in line to get concert tickets for a popular artist.

One of the examples students see of a stack and a queue working together is replicating "cover flow mode" in Spotify. In this mode Spotify shows the cover of the current song, a back button to access songs you've played and a forward button to advance one song at a time through your playlist. I showed the students that the songs accessible by the forward button are loaded into a queue, and a song is placed onto the stack once it plays. If the user clicks the back button the program plays the song on the top of the stack. The purpose of the example was to offer an application of a stack (backtracking through elements), and a queue (walking through elements in order) that the students see almost every day. Once students see this application of the stack and queue we walk together through the implementation of push and pop methods to understand their efficiency.

Binary Search Trees

After covering stacks and queues, the course moves on to introduce Binary Search Trees. One example they see is using the tree to order their playlist by each songs Billboard chart position. Students are also able to see how they can use their trees to play the highest charted song (right most node), or the song that has the most room to improve on the chart (left most node).

Grabbing data from drum machines and DJ interfaces

Throughout the semester I also spend a few lectures walking through the process of getting data from their music hardware. Their API folders contain a sample file (midisample.cpp) that creates a connection with any piece of hardware that outputs MIDI data. Running the sample file prints all of the MIDI data coming from the controller to the console. Each knob will output three bytes of data. The first one or two bytes are used to identify the button and the third byte carries the data for that button.

Once students see the demo, they have the tools to see what data each knob, slider or button will output, and can set up logic to control how their program will behave once they get data from the controller.

Final Project Description

For the final project students were told to write software for the DJ controller that enables a user to play songs, simulate scratching, control the playback speed, mix songs, control bass, middle and treble levels in the audio. Students were also asked to use inheritance, stacks, queues and linked lists. Teams using the drum machines had to trigger sounds with its built in buttons (pads), and make it so that selected pads trigger a sequence of sounds. After the final project was completed the students were asked to reflect on their experience using music hardware and Black music.

Coding the survey responses

Gay describes the concept of a pedagogical bridge as a tool that helps students digest new knowledge by leveraging their experiences. She states that "teachers should integrate cultural diversity into fundamental and high status aspects of the instructional process on a habitual basis…a high percentage of instructional time is devoted to giving examples scenarios and vignettes to demonstrate how …concepts and skills operate in practice. These make up the pedagogical bridges that connect prior knowledge with new knowledge, the known with the unknown and abstractions with lived realities." [5]

Gay sums up the thrust of culturally responsive teaching by saying that it is the act of "using the cultural characteristics, experiences and perspectives of diverse students as conduits for teaching them more effective. It's based on the assumption that when knowledge and skills are situated within the lived experiences and frames of reference of students, they are more personally meaningful, have higher interest appeal and are learned more easily and thoroughly" [5]

The following 3 deductive themes came from the antecedents and effects of pedagogical bridges and culturally responsive teaching:

- Connecting prior knowledge with new knowledge
- Concepts and skills in practice
- Enjoyment and interest

Findings

Connecting prior knowledge with new knowledge

Students reported that Black music helped to provide common ground that helped them to relate to the material, and better understand course concepts. When students were asked to compare problems and examples placed in the context of Black music, with examples in the course text, they gave the following responses.

> “In this course I have learned how to construct a playlist. This has intrigued me because i can see the process in which most music apps are working. It has helped me to visualize the process the music apps go through which is the reason I wanted to be a computer science major to understand things that happen in my everyday life.”

> “While learning data structures through Black music, it made coding simpler and I better understood what each function does. I understood what a shuffle button should do and I could conceptualize it to code it.”

> “I really enjoyed using music when working with problems and examples because I could relate to it more which helped me to remember data structures and how they are used.”

> “When we use songs that I know and relate it to data structures, I feel that I understand the content more than when I just do the course content. This is especially when I look at a playlist as a linked list.”

Students also reported that the addition of Black music as a context and a problem space for the data structures course kept them engaged in the material. When students were asked whether they would like to see more or less problems using Black music as a context in the future, they gave the following responses.

> “I would like to see more problems that use Black music as context in future classes especially … … because it makes the students feel relatable. A lot of people use music as an outlet, so using Black music can be seen as a comfort blanket for the students and have them participate more.”

> “I would like to have more Black music because it kept me interested in the class”

> “It depends on the music; if they’re songs that I like/know, I’m more likely to be engaged.”

> “Yes because it actually gets students excited about their work. Black music is American culture and everyone should be able to relate to.”

Concepts and skills in practice

Students repeatedly emphasized that music also helped to give students a real life context, rooted in their culture, that helped them to see the usefulness of the material. The DJ is a figure that many have seen on TV and in person as the individual that keeps events fun. Designing software for a DJ imparts a desirable cultural context that allows students to quickly visualize the requirements of their application and imagine it in use. When students were asked to reflect on their experience using music hardware for their projects, they offered the following responses.

> “It was an intriguing learning experience as I have always wondered how the DJ operating system worked”

> “This project was very interesting because I was able to incorporate an important part of Black history which I think is essential to making technology accessible to all.”

Students talked about the advantage of seeing data structures concepts that resonate with real life experiences they can call upon for understanding. They gave the following responses when asked to compare music based examples to the course text.

> “In this course I have learned how to construct a playlist. This has intrigued me because i can see the process in which most music apps are working. It has helped me to visualize the process the music apps go through which is the reason I wanted to be a computer science major to understand things that happen in my everyday life”

> “I have had a good experience learning about data structures using Black music. It has helped put the new code into a real life perspective.”

> “I would like to have more problems that use Black music because it helps to know what the code can be used for in real life situations.”

When students were able to connect their culture and experiences with the course concepts, they also were excited at the prospect of discovering new avenues in computer science and music. They gave the following responses when asked whether they would like more examples that use music hardware in the future.

> “I would because it shows how technology is incorporated in music, which is very interesting to me. It could also help someone find a possible career interest.”

> “Yes I would so I can use that skill and development for my platform”

> “Yes, I'd like to try using the drum machines in the future to understand more music equipment and how it can be used with code.”

Enjoyment and interest

Students seemed to genuinely enjoy working with the music for their problems throughout the course. This enjoyment seemed to push students to stay in contact with the material. The combination of cultural context that created curiosity about the DJ equipment, and the tactile and aural feedback from the software and hardware made work fun for the students.

When students were asked whether they would like to have more chances to work with Black music and hardware they gave the following responses:

> "I enjoyed learning how to use the hardware to play songs that I enjoyed, and learning how the computer interacts with said hardware."
>
> "Yes, integrating hardware with software offers more insight into software application and it's fun to play with turntables."
>
> "This project was my first experience with combining hardware and software. Figuring out how each section of the board worked and by what parameters was very enriching and made me excited for my future."
>
> "Yes, because the idea of bringing an item or a music hardware to life seems pretty cool rather than just sitting on a computer and typing a program on a command line."
>
> "Yes, it made me really feel like a dj and it was fun!"

Lessons Learned from the Project

Content has power when students can learn it through the lens of their own experiences. Throughout the semester students were motivated and engaged by music, however they wouldn't have been engaged if they couldn't choose their own songs. If you choose musical content as the cultural context for the course, you must show students how to get their songs in WAV or MP3 format. In past workshops I have prepared samples from popular songs, and students can grow tired of those selections quickly.

One student also voiced their lack of comfort working with musical material throughout the course. It is quite important that instructors emphasize to students that they will be graded on their adherence to the requirements of the assignment, not the aesthetic quality of the music that their software produces. Though some students may have been out of their comfort zones students enjoyed the physical interaction with the hardware, and benefitted from the introduction of Black music as a context for the problems in the course.

REFERENCES

[1] A., M.-H. 2012. Collaborative agency in youth online and offline creative production in Scratch. Research and Practice in Technology Enhanced Learning. 7, (2012), 63.

[2] B., … Kafai Y 2009. Scratch: Programming for everyone. Communications of the ACM. 52, (2009), 60.

[3] Fields, D., Vasudevan, V. and Kafai, Y.B. 2015. The programmers' collective: fostering participatory culture by making music videos in a high school Scratch coding workshop. Interactive Learning Environments. 23, 5 (2015), 613–633. DOI:https://doi.org/10.1080/10494820.2015.1065892.

[4] 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. Proceedings of the 45th ACM Technical Symposium on Computer Science Education (New York, NY, USA, 2014), 85–90.

[5] Gay, G. 2002. Preparing for Culturally Responsive Teaching. Journal of Teacher Education. 53, 2 (2002), 106–116. DOI:https://doi.org/10.1177/0022487102053002003.

[6] Kafai, Y., Searle, K., Martinez, C. and Brayboy, B. 2014. Ethnocomputing with Electronic Textiles: Culturally Responsive Open Design to Broaden Participation in Computing in American Indian Youth and Communities. Proceedings of the 45th ACM Technical Symposium on Computer Science Education (New York, NY, USA, 2014), 241–246.

[7] Manaris, B. and Kohn, T. 2016. Making Music with Computers: Creative Programming in Python (Abstract Only). Proceedings of the 47th ACM Technical Symposium on Computing Science Education (New York, NY, USA, 2016), 711.

[8] Scott, K.A., Sheridan, K.M. and Clark, K. 2015. Culturally responsive computing: a theory revisited. Learning, Media and Technology. 40, 4 (2015), 412–436. DOI:https://doi.org/10.1080/17439884.2014.924966.

[9] Searle, K.A. and Kafai, Y.B. 2015. Boys' Needlework: Understanding Gendered and Indigenous Perspectives on Computing and Crafting with Electronic Textiles. Proceedings of the Eleventh Annual International Conference on International Computing Education Research (New York, NY, USA, 2015), 31–39.

[10] Siva, S., Im, T., McKlin, T., Freeman, J. and Magerko, B. 2018. Using Music to Engage Students in an Introductory Undergraduate Programming Course for Non-Majors. Proceedings of the 49th ACM Technical Symposium on Computer Science Education (New York, NY, USA, 2018), 975–980.