# CS Pathways: A Culturally Responsive Computer Science Curriculum for Middle School

Garima Jain, Fred Martin, Bernardo Feliciano, Hsien-Yuan Hsu
*University of Massachusetts Lowell*
Garima_Jain@student.uml.edu, Fred_Martin@uml.edu,
Bernardo_Feliciano@student.uml.edu, HsienYuan_Hsu@uml.edu

Barbara Fauvel-Campbell
*Daley Middle School*
Lowell Public Schools, Lowell MA
bftech0802@gmail.com

Gillian Bausch, Lijun Ni, Elizabeth Thomas-Cappello
*University at Albany, SUNY, Albany*
gyu@albany.edu, ethomas-cappello@albany.edu,
lni@albany.edu

***Abstract*—In this Research-to-Practice Full Paper, we report on CS Pathways, a middle school computer science (CS) curriculum developed as part of a researcher-practitioner partnership among two public universities and three urban school districts in the Northeast USA. The curriculum serves middle school students to develop apps for social impact. The partnership focuses on bridging the gap between STEM (Science, Technology, Engineering, and Mathematics) and community and gender gap within STEM. The project is based on Culturally Responsive Pedagogy (CRP), which includes the importance of recognizing students' culture in all facets of learning. The project employs a researcher-practitioner partnership (RPP) model, which recognizes that transformational change can occur in educational ecosystems that connect research, policy, practice, and community work.**

**The project curriculum was collaboratively developed by CS researchers, teacher-practitioners, and school administrators. Middle school students develop their own apps that support socially relevant activities in their communities. Using the RPP process, continuous feedback from researchers, teacher-practitioners, and students shaped the curriculum design. Key feedback was collected via one-on-one meetings with the teacher-practitioners, which bridged the visions and knowledge among different groups of the project partners.**

**The curriculum includes the areas of computing and society, digital tools and collaboration, computing systems, and computational thinking. The curriculum helps students develop a critical consciousness of the role they can play in affecting their communities through computing, and empower them to move beyond simply learning to code [1]. The curriculum strives to demonstrate how to integrate computing across middle school subjects in a culturally-responsive way and spread a powerful message of Computer Science for All. This paper advocates for the need for culturally-responsive computing, describes how it is integrated into teachers' instruction, and presents the CS Pathways curriculum design.**

## I. Introduction

As computer science has become more and more important in our world, learning CS is of increasing importance [2]. It is important to make computational education more "inclusive, motivating and empowering for young learners" [1]. While learning computing, young people should also have the opportunity to create using computing that has a direct impact on their lives and communities [1].

An approach that has emerged in the field is "culturally responsive pedagogy," which means aligning CS curriculum and instruction with students' cultural backgrounds and experiences to promote equitable learning outcomes [3]. There is a consensus in the field that it is important to teach computer science in culturally responsive ways [3]. In this paper, we introduce the foundations of the CS Pathways project: its use of culturally responsive pedagogy, the literature of researcher-practitioner partnerships, and Code.org's App Lab, the technology platform used by the project.

### A. Culturally Responsive Pedagogy

Culturally responsive pedagogy (CRP) "values students' identities, backgrounds, and cultural references as key tools for building meaningful learning environments" [4]. It has been proved to be an effective teaching strategy that use diverse students' prior experience and knowledge to make leanings more relevant and equitable [5], [6]. The goal of applying the pedagogy is to support, encourage students and make them learn to love learning, become independent learners, and excel academically [5], [7]. Teachers who implement the pedagogy in their classroom aim to encourage students, give them a sense of belonging, and help creating a space where they feel safe, respected, heard, and challenged. The CS Pathways program is, in similar fashion, a culturally responsive program. It ensures inclusion of diverse points, experiences, and examples.

*1) Dimensions of CRP:* To appropriately implement CRP, teachers are required to be fully equipped with the skills of bridging the gap between instructional delivery and diverse learning styles [8]. Researchers have made many efforts to help teachers enhance their CRP skills [9]–[11]. Per multiple sources, Richards et al. discussed three functional dimensions of CRP which guide educators' practice, namely: the institutional dimension, the personal dimension, and the instructional dimension [9]–[11]. The first dimension, institutional dimension, "emphasizes the reform of cultural factors affecting the organization of schools, school policies and procedures (including allocation of funds and resources), and community involvement" [12].

The second dimension, personal dimension, refers to the "process by which teachers learn to become culturally responsive" [12]. Self-reflection as a teacher is an important part of the personal dimension [10]. Since teachers' values affect relationships with students and their families, teachers must reflect on negative feelings toward any cultural, linguistic, or ethnic group, and when teachers are able to unlearn these prejudices, they help create an atmosphere of trust and acceptance for students and their families, resulting in increase students' chances of success [10].

The third dimension, instructional dimension, refers to "practices and challenges associated with implementing cultural responsiveness in the classroom" [12]. It is not race that matters in culturally responsive education, but prejudice that ignores the cultural tools that students bring to their classes and hinders healthy student-teacher relationships and narrow interpretations of learning [13].

It is important to teach children at an early age, as they are digital natives, and are very adaptable to new technologies. It is essential to change education programs according to the developments in CS, as the very nature of CS is dynamic, flexible, and innovative [14]. To promote this mindset, the tool adopted by the project is App Lab from Code.org. The tool was used in the program to encourage students to choose their own project which further helps to identify their community.

## II. Research-Practitioner partnership

The Research-Practitioner Partnership (RPP) approach is a long-term collaboration between practitioners and researchers organized to investigate practical problems and solutions to improve schools and school districts [15]. RPPs are a very specific form of partnership. They are long-term—instead of focusing on one study, researchers and system leaders share an open commitment to building and sustaining working collaborations across multiple projects. Instead of conducting research to address gaps in existing theory or research, they focus on practical problems—key dilemmas and challenges faced by practitioners. They are mutually beneficial—the focus of the work is negotiated together, and there is a shared authority. RPP employs intentional strategies to facilitate partnerships, building interactions through carefully designed rules, roles, routines, and protocols. The CS field has sought to translate equity-focused research into more relevant and practical applications through the Black, Latinx, Indigenous, low-income, female, and disabled students who are being left out of CS education and career opportunities [16]. Finally, they involve raw analysis of the data—participants collect their own, sometimes sophisticated analytical techniques to answer district questions using administrative data [15].

RPP is "a promising strategy for better aligning the research and practice communities" [17]. When equity is operationalized intentionally in an RPP, both practitioners and researchers feel that their input and interests are equally valued [16]. Researchers who partner with practitioners "understand local contexts, address pressing questions, and produce informative and actionable findings " [17]. Likewise, collaborating with researchers can help practitioners access, interpret, and use research evidence more easily. [17]. When using RPPs, practitioners can help elevate their own needs and experiences in order to produce more relevant research and outcomes. This can help researchers critically examine how power and culture can impact research and education implementations [18].

Given the significance of RPP, researchers and educators at CS Pathways have adapted a coaching approach called One-on-One (O3) [19], which involves RPP members—researchers and teachers. The meeting provided opportunity for researchers and teachers to discuss the design of the project curriculum and help address and any challenges teachers encountered while implementing the curriculum. The O3 meetings as adopted by the CS Pathways have three main features: frequent, scheduled, semi-structured. O3 meetings were scheduled once every week among all teachers. The duration of the meeting ranged from 15 to 30 minutes long. Researchers developed an semi-structured interviews, which aims to inquire teachers' curriculum implementation status. The interviews were recorded with the permission of the teachers and later on were analysed for research purposes.

The O3 process brought two major benefits to the project. First, O3 engages teachers and researchers in developing collaborative CRP-embedded curriculum, which shared exploration of feedback to build and sustain partnerships [19]. By applying the approach and working under the RPP framework, teachers and researchers collaborated as co-designers of the curriculum. The teachers developed, implemented, and tested the materials. The researchers shared research concepts, discussed implementation, and provided feedback on the process and materials developed. The final product was sent to district for further improvement. Second, O3 serves to provide on-time support for teachers to address challenges and difficulties. In return, this provides researchers a chance to learn better of teachers' and their students' needs, which informs our project further design.

## III. What is Code.org?

The CS Pathways selected Code.org's App Lab as the computational design environment for students because of its versatility, power, ability to run on Chromebooks. App Lab allows students to create browser-based applications and share their apps simply by exchanging URLs. In App Lab students code their apps in blocks or JavaScript.

The free and comprehensive curriculum of Code.org includes detailed lesson plans, videos, handouts, offline classes, and online tutorials. Each lesson is connected to standards of computer science, and assignments can be shared with Google Classroom. [20].

Studies show block-based programming interfaces makes it easier for beginners to program with little or no prior experience, and using this approach can help support students from underrepresented groups as they develop their computing skills [21].

In a highly cited review, Kalelioğlu describes how Code.org materials are designed to motivate students and educators

Table I
DISTRICT DEMOGRAPHICS

| **Characteristic** | **District 1** | **District 2** | **District 3** |
|---|---|---|---|
| total district enrollment (count) | 14,436 | 6935 | 9251 |
| gr 6-8 enrollment (count) | 3350 | 1719 | 2068 |
| **Enrollment by Race/Ethnicity** | **% of district** | **% of district** | **% of district** |
| African American | 7.9 | 1.2 | 31.7 |
| Asian, Native Hawaiian, Pacific Islander | 28.9 | 3.8 | 17.2 |
| Hispanic | 31.9 | 39.4 | 20.5 |
| Native American | 0.1 | 0.1 | 0.2 |
| White | 27.4 | 50.5 | 24.2 |
| Multi-Race, Non-Hispanic | 3.9 | 4.9 | 6.1 |
| **Economically Disadvantaged** | 55.9 | 34.8 | 83.9 |
| **English Language Learner (ELL)** | 24 | 11.7 | 4.9 |

to continue learning CS to improve or resolve real-world problems [22]. The lessons use a blended learning approach to teach CS, which means that students learn from a mix of online, self-guided activities and unplugged activities—teacher-led activities that use no computer at all.

Kalelioğlu draws the following conclusions: A slight increase in reflective thinking and skills in problem-solving skills and techniques of female users was observed as compared to male users. The platform is user-friendly and easy to use. It has incorporated playful and colorful elements and characters, making the platform engaging to use and learn. Additionally, Students improved their math, geometry, computing and cognitive skills, and users showed greater interest in programming after using the platform.

Code.org's tutorial videos capture a diverse population of people who practice or pursue a career in CS. Some CS teachers believe that a Geek Gene (or something similar) is necessary to succeed in CS, and not everyone has it [23]. However, Code.org's methods state that a student does not need to be a genius to pursue a career in CS or STEM. The key ingredient to this recipe is dedication [24].

Code.org's site includes ample of open-source projects which allows a student to walk through a repository of ideas to learn and implement. One of the important tabs in the library is called The Remix. It is a feature that allows one to take an already published project and remix it with their own ideas and take it to next level.

Teachers have historically had some concerns about Code.org [20]. Owing to the way the programming units are designed, students may work in isolation, with teachers lacking clarity in their role, and classrooms arriving at the end of the unit lacking the skills to take on major independent projects [20]. To address these concerns, CS Pathways project uses an Research-Practitioner Partnership (RPP) approach to fully engage teachers in supporting their students' learning.

## IV. PROFESSIONAL LEARNING PROCESS

One of the challenges of teaching computer science is choosing the right curricula and programming languages and environments for your classroom [25]. This is difficult for teachers who have little or no experience with coding, as the programming environment and the curriculum both play a role in shaping the way students understand CS concepts and the programming strategies they develop [25]. As part of the RPP partnership, the university partners organize professional learning (PL) meetings for the teachers. All the teachers from the same district gather and work on how all of the members of the project can work together to achieve a goal. The project's PL series starts by introducing the new teachers to the other members of the project. After the introduction, the meeting leads to discussion of project goals, approach, outputs and outcomes. The members assist the team of teachers to get started by providing and walking them through the resources like Google Classroom Curriculum, Connected Code Book and Code.org. The Google classroom is structured in way which has all the resources to teach all the lessons. It is a combination of unplugged activities, lesson plans on CS concepts, example apps and tutorials. The Connected Code book is a book that focuses on Why every child needs to learn to code: the shift from "computational thinking" to computational participation.

Once the teachers are acquainted with the Google Classroom, they are introduced to App Lab. A few unplugged activities that are used to teach CS concepts without using programming or computers are introduced to the teachers. Each unit of the curriculum is described to the teachers. A project based on CS concepts like events, multi-screen, variables, conditionals is explained and worked on with the teachers. By the end of the PL, the teachers have a total of three apps ready by themselves. The first app created is an informational app, has home screen with buttons which links to multiple other screens. Each of those other screens links back to home. The app gives information about a renowned academic or professional identity. It gives information about the person in a succinct way. The second app quiz app that keeps score with a variable. The app is designed for the use of the school community. Lastly the third app is an event driven app which uses conditionals, functions and variables. The apps include all the features explained while serving the goal of the project to serve the community.

## V. Data Sources and Methodology

Building on the RPP framework, the project has gathered data from a variety of sources, including: (1) the aforementioned "O3" interviews with teachers; (2) notes from monthly project meetings with teachers; (3) pre- and post-surveys completed by teachers; (4) analysis of instructional materials prepared by teachers; and (5) analysis of student work products (i.e., their apps). The results presented in this paper are based on use of a grounded theory approach to make sense of these data [26].

## VI. CS Pathways Curriculum

The CS Pathways curriculum is designed to fit in teaching practice of three urban school districts in the Northeast USA. The demographics of the three school districts are shown in Table I. The curriculum aims to serve both Massachusetts' Digital Literacy and Computer Science [27] and New York's Computer Science and Digital Fluency [28] standards. By aligning the curriculum with computer science and literacy standards, providing language scaffolding, integrating inquiry-based approaches, and providing culturally responsive pedagogies and materials, the curriculum is tailored to meet the needs of culturally diverse students in the region. The knowledge and associated skills included in the curriculum are "essential both to prepare students for personal and civic efficacy in the twenty-first century and to prepare and inspire a much larger and more diverse number of students to pursue the innovative and creative careers of the future." [27]

The curriculum was developed in collaboration among practicing teachers and CS Ed researchers, which included teachers implementing activities in their classrooms. The team organized the curriculum by putting their expertise in CS and trying to make it culturally responsive. Researchers and faculty aligned materials with Common Core State Standards and the State Standards. Culturally responsive stories depicting diverse characters that pioneered the fields of computer science and engineering were chosen to make the content student-friendly. In addition, researchers and teachers developed a module in the curriculum as a speaker series where professionals with diverse backgrounds visit the classroom (often virtually) to describe their career journey and day-to-day work.

The final product of the project curriculum is an integration of computer science with other subjects including Technology, Social Studies, ELA, Mathematics, and Sciences. The full CS Pathways curriculum has five units and is designed to be completed in 20 hours (four hours per unit). In each unit, we integrated an unplugged activity module and a culturally responsive module. It is designed to be flexible so teachers can integrated it into their existing curricula. Table II presents the five major units of the CS Pathways curriculum.

### A. Unit I: Why Computer Science? & CS for All

Unit I is designed for students to learn about the world of CS and how it can relate to students' lives, including future careers, connecting students' awareness of issues, activities, products, media, and people to the relationship with CS and begin engaging in the collaborative and social experience of CS. Students identify what coding is and how it relates to the real world.

Through this unit students learn about the world of CS and the connections between CS and issues, activities, products and media that are relevant in their lives. They learn that they can become producers of CS products as well as consumers by creating apps. Students are introduced to the idea that CS is a collaborative and social experience in the way that people create applications using CS and how CS affects communities. In the module students define identity and community within their environment. They understand CS as a form of digital media and define how CS is used as a communication medium.

The unplugged activity module introduce student's to pair-programming through using Code.org videos. It is an approach to writing code where two learners work side-by-side. One person is the "driver" and controls the keyboard and mouse and does the programming. Their partner is the "navigator" and follows all the work and makes suggestions and thinks ahead. The pairs exchange roles at regular intervals.

The unit consists of a culturally responsive activity module, which helps to share their culture and community to others. Each student participates and learn more about their classmates and friends. Through Vision Boards, students create a class slide deck to express who they are and get to know about their classmates as well. As students create their vision boards, they explore their identities and the concept of community in preparation for designing apps that can serve their communities. They also learn about fair use of media.

The unit also consists of Introduction to App Lab platform and a few examples are presented. This module is designed to convey the message of "CS For All." As the lectures and classes proceed students are introduced to the CS concepts of coding and programming (using blocks), and a few simple debugging techniques.

### B. Unit II: App Introduction and Creation

In this unit, students achieve some success at making a simple app to introduce CS in a way that motivates students with different levels of CS experience. The teacher introduces the whole process of creating an app. It helps the students to understand what an app is, experience an app, remix an app and go through the process of making an app on their own.

The modules in Unit II are designed to help students easily add images and sounds into an App Lab project while also teaching (or reviewing with) them about creativity, copyright, and fair use. The unit also encourages students to create or remix an app that teaches about or engages a user in a subject/topic assigned by the teacher or derived from the framing activity in unit one. Additionally, in this unit, students relate the concept of algorithms back to everyday activities. After discussing their steps, students are introduced to another unplugged activity that is to make paper planes using an algorithmic approach. The goal is to develop the skills necessary to translate real world situations into online scenarios and vice versa.

Table II
CS PATHWAYS CURRICULUM OVERVIEW

| **Units** | **Goals** | **Unplugged Activities** | **Culturally Responsive Activities** |
|---|---|---|---|
| I. Why Computer Science? & CS for All | Learn to become consumers and producers of CS.<br>Introduction to idea that CS is collaborative and social experience.<br>Introduction and application of the concepts of CS. | Practice Pair programming. | Vision Board |
| II. App Introduction and Creation | Learn app creation and interaction of different elements.<br>Resources for images, videos, and audio. | Unplugged Algorithm Drawing Instructions | Pitch your Passion |
| III. Basic Computer Science | Gain an understanding of careers in CS.<br>Knowledge of routines and simple block placement.<br>App development: Create a simple app learning about Creative Commons and approaches for remixing media. | None | Career in Computer Science - Speaker Visit (Skype a Scientist) |
| IV. More Complex Computer Science | Get hands-on experience in the CS iterative, modular and media concepts.<br>Draw on prior knowledge of creating a simple app to create a multi-function, more complex app. | Code.org's unplugged activity on variables, and functions | None |
| V. Final Project | Design, develop and showcase a culturally responsive project. | None | Student App Showcase |

The unplugged activity introduced in Unit II is about designing paper towers. The class works in groups to design paper towers that will be as tall as possible. The activity helps the students to work together and solve problems as a team. Teamwork-based projects encourage the students to use the technical knowledge to solve complex problems [29]. At the end of the lesson, the students reflect about their work on the activity and identify the types of problem solving they may be working on in the rest of the course.

The culturally responsive module in this unit, contributed by one of the teachers, is to help students to make applications that reflect their real-life interests. This not only allows students to choose their own idea, but also to embrace something they are passionate about and drive a real change in the community. "Pitch Your Passion" activities take the student apps and then allows students to showcase them to their classmates, catching their interest as well. This interactive process helps students connect with each other by sharing these personal passions and increases their interest in the CS code.

### *C. Unit III: Basic Computer Science*

Unit III is about exploring careers in CS and be able to begin an App from the beginning. This begins with teacher guidance so the students gain an understanding of independently create a simple app that uses Creative Commons approach to download media. These apps can also be adapted as final project after addition of new features.

Students are introduced to variables. They learn how to manipulate and modify variables. They integrate a variable into their existing app or start a new one. Students work together to investigate several variations of an app to understand how variables are stored and information is updated. At the end of the unit, students identify common programming patterns when using variables, understand their purpose, and modify apps that use common programming patterns with variables to improve their functionality [30].

Students design and create a 4-screen app on a topic of their choosing. Students may collaborate with a classmate as a "thought partner." Additionally, in this unit students practice their programming skills by using the different concepts they learned in the previous lesson. To begin they are introduced to the concept of debugging and how using debugging tools and techniques resolve programming issues. They are encouraged to use debugging and reflect on this practice throughout the lesson. At the end of the lesson students share their experiences debugging as well as new realizations about programming [30].

The unit includes a speaker visit as part of the CRP, called "Skype A Scientist." This activity aims to have an industry professionals who uses CS make a virtual or in-person visit to the classroom. Through their interactions, students are exposed to the vast ways in which CS is employed across many different fields. This enables students to see multiple career paths that might intersect their own interests, rather than just assuming CS coding is for one group only. These speakers also use these opportunities to promote social justice, such as the gender pay gap, by broadening the participation in CS.

### *D. Unit IV: More Complex Computer Science*

In Unit IV students create more complex apps or remix apps to make them more complicated. Students learn concepts and skills that will let them add desired functionality to their

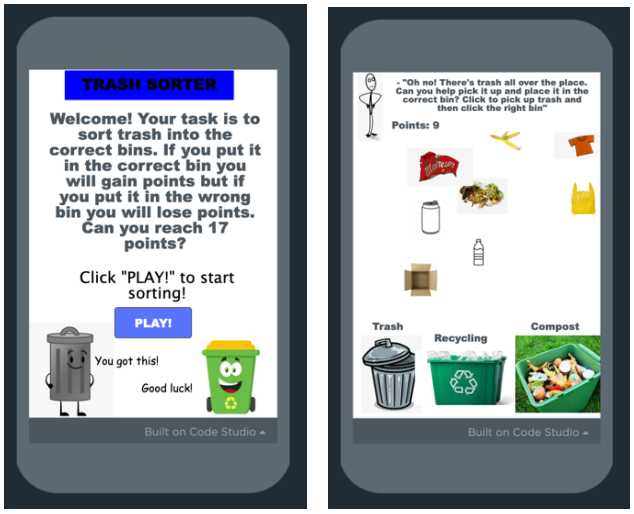


Figure 1. Trash Sorter app screens

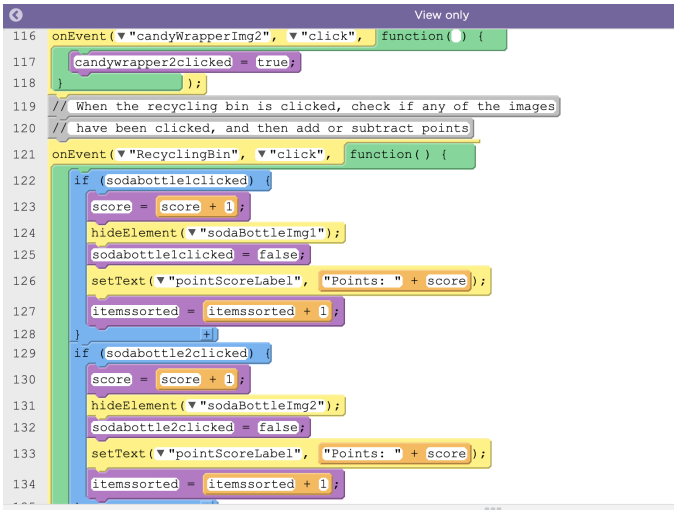


Figure 2. Trash Sorter app code excerpt

apps. This unit is designed to focus on teaching students' CS knowledge and skills.

In this Unit, students learn about conditionals by using the materials from the Conditionals Explore lesson in Code.org. Flowcharts are introduced for understanding how computers make decisions using Boolean expressions. Students work together on three different versions of an app to understand how through Boolean expressions and conditional statements programs make decisions. In each guided investigation students first watch a short video on a concept and then use the concept on the app. After adding extensions and modifying the app, students learn how to use appropriate vocabulary to describe Boolean expressions and conditional statements. They evaluate expressions that include Boolean values, comparison operators, and logical operators, and create simple programs that use these concepts [30].

Students are introduced to the concept of how functions are used in programs. As described by Code.org, "Students use appropriate vocabulary to describe the declaring, defining and calling of functions. They trace the flow of execution in programs that declare and call functions, describe the way a function call interrupts the normal flow of execution within a program and modify programs that declare and call functions to adjust their functionality." [30] The students start the lesson by discussing two ways to write the lyrics of a song: one that includes a lot of repeated text and one that does not. After exploring this example, students investigate and explore how functions can be used to remove repeated code from a program. At the end of the lesson, students discuss the concept of a function and give definitions to their journals [30].

### *E. Unit V: Final Project*

Unit V of the curriculum is where students create their final apps. They often do this by extending work done in the prior units. We present two sample student apps here.

*1) Trash Sorter App:* One of the students designed and developed a project called Trash Sorter App. The game is aimed to provide information on what kinds of trash goes to which bin. The app is designed to spread awareness and education about waste reduction.

The app consists of the four main sections: home page, gaming section, information section and an end page. The home page of the app is a welcome screen which gives a nice description of the app as shown in Figure 1. The student has used labels, images and buttons to make it visually appealing. The Play button takes to the game screen which has a small description of the game.

The Trash Sorter App has three types of bins: Trash, Recycle and Compost. The game is worth 17 points and the user's goal is to sort the trash and place it in the correct type of bin. The student uses event handling to make a working game. If the correct trash bin image is clicked in correspondence with the trash image, the score increases with one point, otherwise decreases with one point. Once all the trash is sorted by the player, the user is directed to information screens that provide information and facts, along with citations, about trash and waste recycling. Upon reading the information the user reaches the end screen which displays a message, the score of the player and encourages one to play the game again.

The code of the app uses complex concepts of CS, including conditionals, manipulation of variables, event handling and more (Figure 2). One of the best practices while writing code is to put comments to make the code easier to comprehend. The student has used comments which help the reader to understand what the next section is about.

The student project incorporates the aspect of being culturally responsive as the app aims to make users and players learn more about recycling and the issue caused due to littering and pollution. Waste recycling is an initiative every community and county is responsible for. This app recognizes the importance of waste recycling and henceforth, partnered with the community to spread awareness, educate people about the kinds of waste, how and why to recycle the trash. In

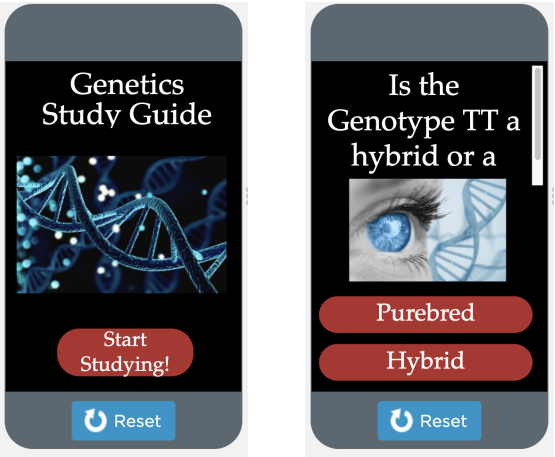


Figure 3. Genetics Study Guide app screens

```
81  onEvent("Q5_A2", "click", function() {
82    setScreen("EndGameScreen5");
83    //Play sound whenever Q.5Answer2 is clicked
84    playSound("sound://category_alerts/vibrant_wrong_action_hit_1.mp3");
85  });
86  onEvent("Q5_A3", "click", function() {
87    setScreen("FinishScreen");
88    //Play sound whenever Q.5Answer3 is clicked
89    playSound("sound://category_achievements/puzzle_game_achievement_01.mp3");
90  });
91  onEvent("EndButton", "click", function() {
92    setScreen("HomeScreen");
93    //Play sound whenever EndButton is clicked
94    playSound("sound://category_accent/puzzle_game_accent_a_01.mp3");
95  });
96  onEvent("StartOverButton5", "click", function() {
97    setScreen("HomeScreen");
98    //Play sound whenever StartOverButton5 is clicked
99    playSound("sound://category_accent/puzzle_game_accent_a_01.mp3");
100 });
```

Figure 4. Genetics Study Guide code excerpt

exemplifying how to use a given CS skill, the student designed an app that provides a solution to a real-world problem and is geared towards community service.

*2) Genetics Study Guide:* One of the students designed and developed a science project called Genetics Study Guide. The App aims to help synthesize and summarize the information in the form of asking questions. The app is useful for reviewing difficult or complex concepts or subject areas.

The app consists of the four main sections: HomeScreen, QuestionScreens, EndGameScreen and a FinishScreen. The home page of the app is a welcome screen which prompts the user to click the Start Studying! button (Figure 3). The student used labels, images and buttons to make the app visually appealing. The Start Studying! button takes to the QuestionScreen screen which asks questions about the topic.

The Genetics Study Guide App has multiple choice questions. The user's goal is to reach the FinishScreen by correctly answering all the questions related to the topic. The student uses event handling to make a working model of the game. If the correct option is clicked with respect to the question asked the user is prompted with a next question on the other screen. Each time when a button is clicked a the user hears a sound to confirm their selection. If the question is incorrectly answered, the user is directed to an EndGameScreen which provides the correct answer to the question. To proceed with new questions a user has to start over the questions from beginning. Doing this allows a user to ingrain the concept and be confident with the material. Upon successfully completing all the questions, the user is directed to the FinishScreen. The screen has a Back to home screen button which encourages the user to review the study guide again.

The code of the app uses event handling and sounds (Figure 4). The student also applied the learning of Creative Commons by adding the source of the resources that they have incorporated in their App.

The student project incorporates the aspect of being culturally responsive as the app aims to serve the school community with additional resources on the course. Creating a personalized study guide helps to review the information in a way that is most helpful and can help to improve the test scores as a result. The student used a given computer science skill to create an app that solves a real-world problem and is designed to help the community.

Table III
EXAMPLE CURRICULUM IMPLEMENTATION

| **Lesson Name** |
|---|
| 1. Coding Vocabulary |
| 2. Coding Vocab and Intro to App Lab 1-4 |
| 3. Coding Vocab Assessment |
| 4. Intro to App Lab 5-7 |
| 5. Intro to App Lab 8-15 and Women Engineers and CS |
| 6. Coding and Cybersecurity |
| 7. App Design and Pitch Your Passion |
| 8. Using App Lab |
| 9. Computer Science Careers; Creative Commons, Copyright, & Fair Use |
| 10. App Lab Get Started |
| 11. Building Your App: Remix? |
| 12. Debugging Your App |
| 13. Sharing Your App |
| 14. Computer Programming Post Assessment |

The scope of students' work is to create prototypes of apps that they share in the school context: with their classmates and sometimes in the larger school community. The students' work is intended to be introductory; in subsequent activities, teachers may be able to facilitate full partnerships with community members.

## VII. AN EXAMPLE CURRICULUM IMPLEMENTATION

This article's fifth author is one of the project's teacher-practitioner partners. This teacher implemented the curriculum in 14 lessons per Table III [31]. The curriculum begins with Coding and Computer Programming. Students identify what coding is and how it relates to the real world. Using Google Jamboard, students share their thoughts ideas about coding. This is followed by a discussion and a YouTube video lecture, "What is coding?" [32]. The video highlights that we are all surrounded by computer programs. The students also watch a small clip from diverse professionals about computer

programming [33]. The students end the lesson with a 8 minute video introducing the Peanut Butter and Jelly Sandwich and Computer Programming Analogy. In this video students learn program development through making a peanut butter and jelly sandwich. They learn that computers are very literal and only can follow instructions [34].

The Lesson 1 identifies key vocabulary words needed for coding. As reported by the teacher, her classes have a big group of linguistically diverse students. Therefore, to better engage them in learning, students practice listening to vocabulary using listen tool and activities. The teacher prepares a Code Vocabulary Quizlet and the students review and practice it for 14 minutes. She also get her students introduced to App Lab with the Getting Started tutorial on Code.org.This teaching practice reflects how the teacher applied CRP to meet the needs of her diverse students' population.

The Lesson 2, 3, 4 and 5 aims to give hands-on learning in App Lab. The lessons provide a step-by-step guide for creating buttons, text, images, sounds, and screens using either blocks or text. Students are given the opportunity to continue working on their projects or explore new projects featured at `code.org/applab`. The students again review the coding vocabulary learned during the prior lecture videos and lessons. In Lesson 5 students also identify outstanding women in computer science through YouTube videos like Why So Few Women in Computer Science?, Hidden Figures, and more [31].

Lesson 6 introduces cybersecurity. Students learn about techniques to protect the secrecy, integrity, and availability of computer systems and data against threats. Students are introduced to strategies to keep information secure but also as accessible as possible including use of passwords, biometrics, and access privileges [35].

In Lessons 7 and 8, students identify a passion for an app and discuss and sketch its idea. The lesson also includes conditionals and functions and serves as an segue to CRP, as students create apps based on their own interests. Students take an issue affecting their community, propose a solution which uses the app to help, and follow it through to the end of the class. Students showcase their work, mentioning the community or cultural challenges being addressed, as well as the CS concepts being implemented to complete the app [31].

Lesson 9 aims to identify careers in CS and learn about Creative Commons and approaches for remixing media. The lesson is followed by a video of Amazon Future Engineer a virtual tour. Students also explore Careers in Computer Science through a short 12 minute video from YouTube. This video covers a variety of computer science careers, as well as the many different areas of computer science that are available to explore. Computer science is a vast and varied field that encompasses a range of topics, including software development, cryptography, cybersecurity, computer graphics, bioinformatics, and more. Many people imagine software developers when it comes to computer science, but there are many other positions such as security analysts, web developers, network system administrators, and so on. By employing a CRP and broadening students' view of CS careers, they can see where their own cultural interests intersect, and begin to see themselves as computer scientists. The jobs listed in the video have the highest employment and growth rate of most jobs on the bureau of labor statistics [36].

In Lesson 10 and 11, students work on their app final project proposal. Lesson 12 aims to identify and fix problems in their code through the debugging process. Students are introduced to debugging and common problems faced while programming. The students also learn a new command console.log. Students notice that it prints out text in the Debug Console below the code workspace. In lesson 13 students share their app by creating a hyperlink to App Lab project and post in discussion board on Google Classroom. Finally the lesson 15 ends with Computer Programming Post Assessment [31].

## VIII. Discussion and Conclusion

One of the goals of the project is for students to recognize that they can create an app which can do wonders. As researchers visited classrooms, they observed the enthusiasm and excitement that students had to complete their app and make an impact in the society. It kept girls involved and combined real interests (e.g. fashion) with technology. Everyone felt proud to share apps to each others' phones. With every new concept students learned, they tried to incorporate them into their work. Students felt confident with the collaborating with their peers.

As described by Tissenbaum et al. "With rapid changes happening in both computing and computing education landscapes, we have an opportunity to reconsider how students learn computing. Young learners have the capacity to develop computational products that have authentic impact in their lives from the moment they begin to code." [1] According to Tissenbaum and Ottenbreit-Leftwich "The rapid integration of CS into the current education system has challenged states, districts, and teacher preparation programs to revamp their current efforts considerably" [37]. CS Pathways has enacted a curriculum that deeply considers their students and their culture. By integrating computing with technology, science, and other subjects, it connects CS to the lives of students, young women, and underrepresented minorities. It prepares students to solve problems with CS and introduces them to understanding of the systems that surround the code they write. It serves CS for All.

## Acknowledgments

We thank all project teachers and students for their dedication to this work. We thank Foozieh Mir, project manager of CS Pathways, who contributed greatly to success of this project. We also thank Pauline White for her insights in supporting teacher professional learning, and Diane Schilder for her contributions as the project evaluator. This material is based upon work supported by the National Science Foundation under Grant Numbers 1923452 and 1923461.

## References

[1] Mike Tissenbaum, Josh Sheldon, and Hal Abelson. From computational thinking to computational action. *Communications of the ACM*, 62(3):34–36, 2019.

[2] Carsten Schulte, Malte Hornung, Sue Sentance, Valentina Dagiene, Tatjana Jevsikova, Neena Thota, Anna Eckerdal, and Anne-Kathrin Peters. Computer science at school/cs teacher education: Koli working-group report on cs at school. In *Proceedings of the 12th Koli Calling international conference on computing education research*, pages 29–38, 2012.

[3] Tia C Madkins, Alexis Martin, Jean Ryoo, Kimberly A Scott, Joanna Goode, Allison Scott, and Frieda McAlear. Culturally relevant computer science pedagogy: From theory to practice. In *2019 research on equity and sustained participation in engineering, computing, and technology (RESPECT)*, pages 1–4. IEEE, 2019.

[4] Constance L McKoy, Rebecca B MacLeod, Jennifer S Walter, and D Brett Nolker. The impact of an in-service workshop on cooperating teachers' perceptions of culturally responsive teaching. *Journal of Music Teacher Education*, 26(2):50–63, 2017.

[5] Geneva Gay. *Culturally responsive teaching: Theory, research, and practice*. teachers college press, 2018.

[6] Shuchi Grover and Roy Pea. Computational thinking in k–12: A review of the state of the field. *Educational researcher*, 42(1):38–43, 2013.

[7] Zaretta Hammond. *Culturally responsive teaching and the brain: Promoting authentic engagement and rigor among culturally and linguistically diverse students*. Corwin Press, 2014.

[8] Dr Yvonne Ryans. Culturally responsive teaching. 2016.

[9] Mary E Brenner. Adding cognition to the formula for culturally relevant instruction in mathematics. *Anthropology & education quarterly*, 29(2):214–244, 1998.

[10] Heraldo V Richards, Ayanna F Brown, and Timothy B Forde. Culturally responsive instruction, 2006.

[11] Heraldo V Richards, Ayanna F Brown, and Timothy B Forde. Addressing diversity in schools: Culturally responsive pedagogy. *Teaching Exceptional Children*, 39(3):64–68, 2007.

[12] Matthew Lynch. What is culturally responsive pedagogy? — https://www.theedadvocate.org/what-is-culturally-responsive-pedagogy/, May 2020.

[13] Elena Aguilar. *Making Connections: Culturally Responsive Teaching and the Brain — https://www.edutopia.org/blog/making-connections-culturally-responsive-teaching-and-brain-elena-aguilar*. George Lucas Educational Foundation, Edutopia, Feb 2015.

[14] Vic Grout and Nigel Houlden. Taking computer science and programming into schools: The glyndŵr/bcs turing project. *Procedia-Social and Behavioral Sciences*, 141:680–685, 2014.

[15] Paul Cobb, Kara Jackson, Thomas Smith, Michael Sorum, and Erin Henrick. Design research with educational systems: Investigating and supporting improvements in the quality of mathematics teaching and learning at scale. *Teachers College Record*, 115(14):320–349, 2013.

[16] Roxana Hadad, Jean Ryoo, Julie Flapan, and Steve Kong. Defining and delivering "equity" in an rpp. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, pages 1–2. IEEE, 2021.

[17] Research practice partnerships — https://rpp.wtgrantfoundation.org/about/, Oct 2019.

[18] Jill Denner, Steve Bean, Shannon Campe, Jacob Martinez, and David Torres. Negotiating trust, power, and culture in a research–practice partnership. *AERA Open*, 5(2):2332858419858635, 2019.

[19] Gillian Bausch, Pauline White, and Hsien-Yuan Hsu. One-on-one meetings as boundary practices: Managing rpp computer science curriculum co-design. Nov 2021.

[20] Christy Matte. Code.org review for teachers — https://www.commonsense.org/education/website/codeorg, Dec 2021.

[21] David Weintrop, Heather Killen, and Baker E Franke. Blocks or text? how programming language modality makes a difference in assessing underrepresented populations. International Society of the Learning Sciences, Inc.[ISLS]., 2018.

[22] Filiz Kalelioğlu. A new way of teaching programming skills to k-12 students: Code. org. *Computers in Human Behavior*, 52:200–210, 2015.

[23] Mark Guzdial. Technical perspective: Is there a geek gene? *Communications of the ACM*, 63(1):90–90, 2019.

[24] Code.org. *What Most Schools Don't Teach — https://www.youtube.com/watch?v=nKIu9yen5ncamp;t=31s*. YouTube, Feb 2013.

[25] David Weintrop, Alexandria K Hansen, Danielle B Harlow, and Diana Franklin. Starting from scratch: Outcomes of early computer science learning experiences and implications for what comes next. In *Proceedings of the 2018 ACM conference on international computing education research*, pages 142–150, 2018.

[26] Barney G Glaser and Anselm L Strauss. *The discovery of grounded theory: Strategies for qualitative research*. Routledge, 2017.

[27] Massachusetts Department of Elementary and Secondary Education. Digital literacy and computer science (dlcs) https://www.doe.mass.edu/stem/dlcs/.

[28] Computer science and digital fluency learning standards http://www.nysed.gov/curriculum-instruction/computer-science-and-digital-fluency-learning-standards.

[29] Robert Lingard and Shan Barkataki. Teaching teamwork in engineering and computer science. In *2011 Frontiers in education conference (FIE)*, pages F1C–1. IEEE, 2011.

[30] Code.org. Variables, conditionals, and functions— https://curriculum.code.org/csp-20/unit4.

[31] Barbara Campbell. App lab unit — https://sites.google.com/view/app-inventor-unit/.

[32] DK Books. *Coding for Kids 1: What is Computer Coding? — https://www.youtube.com/watch?v=THOEQ5soVpY*. YouTube, Dec 2014.

[33] Sahibzada Rizwan. *Is computer programming hard? How to become a computer programmer — https://www.youtube.com/watch?v=QERdqYoK8M4*. YouTube, Jan 2017.

[34] TotalSeminarsChannel. *Scott Jernigan on: Programming with a PBJ Sandwich (CompTIA IT Fundamentals FC0-U61) — https://www.youtube.com/watch?v=HXl5f2azATU*. YouTube, Nov 2019.

[35] CrashCourse. *Cybersecurity: Crash Course Computer Science 31 — https://www.youtube.com/watch?v=bPVaOlJ6ln0*. YouTube, Oct 2017.

[36] Zach Star. *Computer Science Careers and Subfields — https://www.youtube.com/watch?v=2VpDcjFshJI*. YouTube, 2018.

[37] Mike Tissenbaum and Anne Ottenbreit-Leftwich. A vision of k— 12 computer science education for 2030. *Communications of the ACM*, 63(5):42–44, 2020.