

Scratch Encore: Creating and Sustaining Culturally Responsive Computer Science Education

Rasha Alkhateeb¹, J. Elisabeth Kasner², David Weintrop¹, Jennifer Palmer³, Merijke Coenraad⁴, Minh Tran³, and Diana Franklin³

¹University of Maryland, ²Florida State University, ³University of Chicago, ⁴Digital Promise

OVERVIEW

Scratch Encore (Canon Lab, n.d.) is a culturally relevant, student-centered, 14 module, computer science curriculum for 4th to 8th-grade learners that introduces foundational computing topics using the Scratch environment. It employs three key design goals: (a) supporting teachers, (b) supporting learners, and (c) using culturally responsive practices to address long standing inequities in computing. The curriculum offers equitable and effective learning experiences for students who have historically not had equal opportunities to fully participate in computing while providing a wide array of supports for educators who may be inexperienced with Scratch and/or programming. This article features a high-level overview of Scratch Encore and the first 6 modules in greater detail to help teachers understand the content and pacing of the curriculum.

Topics: Computer Science Education, Culturally Responsive Pedagogies, Programming

Time: 14 modules. Each module is approximately 2-4 lessons. Each lesson takes 60-120 minutes.

MATERIALS

- [Scratch Encore](#) (Canon Lab, n.d.) registration
 - o Materials provided in Scratch Encore (module lesson plans, student worksheets, worksheet answer keys, [Scratch Encore assessments](#), videos introducing focal concepts). Refer to Support materials at the end of this article for additional sources.
- Student devices with keyboards (1:1 or 1:2)
- Student Scratch accounts
- Projector/Screen

CONTEXT-AT-A-GLANCE

Setting

4th-8th grade classrooms with a focus on computer science and/or programming.

Modality

Face-to-face, hybrid, or online (synchronous or asynchronous)

Class Structure

14 modules with 2-4 lessons each. Lessons vary from 60-120 minutes.

Organizational Norms

Modules have a common, Use→Modify→Create gradual release structure. The content is cumulative so modules should be completed in order. Students should work individually or in pairs.

Learner Characteristics

Scratch Encore's modules, specifically designed to support learners from groups underrepresented in computing, can be used to address the challenges of bringing high-quality CS instruction to historically excluded and minoritized students in computing.

Instructor Characteristics

Scratch Encore is designed for 4th-8th grade teachers. The curriculum offers numerous teacher supports, including teacher guides, assessments, and videos.

Development Rationale

This work supports teachers in teaching CS in a culturally responsive way. It provides a deep dive into computing concepts in the Scratch Environment.

Design Framework

Constructionist Design Theory; Leverages the Use→Modify→Create Pedagogical Approach, blending structured and open-ended activities.

SETUP

Scratch Encore (Canon Lab, n.d.) was designed to allow for teacher choice as they work to meet the needs of their students (Franklin et al., 2020). In the first 6 lessons, teachers choose which thematic strand they want to use (multicultural, youth culture, or gaming) and then use the provided materials in their classroom.

Across the Scratch Encore curriculum, teachers will introduce computing concepts by situating them in themes that resonate with students. Students will also have the opportunity to create Scratch projects that reflect their own interests and ideas.

Teachers can choose between digital and print versions of the student resources. Scratch Encore is designed to be used in the classroom but can be used for remote, hybrid, and asynchronous instruction.

To participate, students need a device with a keyboard that can run [Scratch](#). Teachers may find that a projector or large screen can be helpful to display curriculum materials to the class.

STANDARDS

Scratch Encore supports the following standards:

- Computer Science Teachers Association (CSTA, 2017) standards:
 - 1B-AP-10; 1B-AP-11; 1B-AP-12; 1B-AP-13; 1B-AP-14; 1B-AP-15; 1B-AP-17; 2-AP-10; 2-AP-13; 2-AP-16; 2-AP-19; 1B-1C-20; 1B-1C-21
- International Society for Technology in Education (ISTE, 2016) standards for students:
 - 1c; 1d; 2b; 4a; 4c; 4d; 6b; 6d; 7a

CONTEXT AND SETTING

Scratch Encore is aligned with [CSforAll](#) (n.d.), a national movement to bring high-quality computer science education to all students. The Scratch Encore curriculum fulfills a specific need in upper-elementary and middle school education for culturally responsive computer science education.

Scratch Encore is designed to live between first-touch computer science experiences and more

advanced curricula that use text-based programming languages. As such, the curriculum is designed for 4th-8th grade students learning computer science. The curriculum was initially designed for face-to-face classroom instruction but, in response to the COVID-19 pandemic, the curriculum can now also be taught fully virtually as all materials (e.g., activities, assessments) can be completed through online forms. Each module has a common structure relying on various, research-backed pedagogical approaches. The details of the curriculum are more fully described in the Learning Representation section.

Scratch Encore consists of 14 modules with 2-4 lessons per module. Lessons vary from 60-120 minutes depending on the prior experience of the students and the pace the teacher sets. While the full curriculum may be completed across multiple school years, many teachers use only the first few modules due to time constraints. The modules are cumulative, with later modules building on the concepts and skills introduced in earlier modules.

Scratch Encore was created with three key design goals in mind: (a) supporting teachers, (b) supporting learners, and (c) being culturally responsive to address long-standing inequities in computing.

To support teachers, the curriculum is fully featured, meaning it includes detailed lesson plans, worksheets with answer keys, automated assessments, supporting videos, and additional resources related to multicultural topics. Further, there is an online, asynchronous Scratch Encore professional development course that teachers can elect to join (see edX, n.d.). Collectively, these scaffolds make it possible for teachers with little prior computer science or Scratch experience to successfully teach the curriculum in their classrooms.

To support students, Scratch Encore provides a scaffolded introduction to the Scratch environment, programming concepts, and computer science more generally. Students have the opportunity to engage with computing content and create personalized projects that reflect their ideas and interests. The curriculum was designed using research backed pedagogical strategies including TIPP&SEE (Salac et al., 2021), Use→Modify→Create (Franklin et al., 2020), and structured planning sheets to help students create their own projects (Tsan et al., 2022).

Finally, towards the goal of making a culturally responsive curriculum, Scratch Encore includes three thematic strands. Each strand situates the content in a different theme, allowing the teacher to choose how to present the content and providing opportunities for them to enrich the context based on their own knowledge and experiences. These themes were co-designed with educators, students, and parents (Coenraad et al., 2019). When explaining concepts, examples can be drawn from students' current knowledge and existing prior experiences (e.g., Scott et al., 2015).

LEARNING REPRESENTATION

In this section, a high-level overview of Scratch Encore (Canon Lab, n.d.), including its core ideas, its structure, and how it supports students and teachers, is presented.

Each Scratch Encore module is situated within a specific youth-oriented theme. The themes used in the curriculum were designed in collaboration with educators, students, and parents to ensure that they resonate with middle school learners (Coenraad et al., 2019). Modules 1-6 contain three thematic strands that situate CS content: multicultural, youth culture, and gaming.

The curriculum draws on constructionist learning theory, which emphasizes hands-on learning where students construct personally meaningful artifacts (Papert, 1980), which, in this case, are Scratch programs. Each module utilizes a Use→Modify→Create structure to scaffold knowledge of that concept.

Scratch Encore's unique stranded structure empowers teachers to select the version of the module that they think will resonate with their students most. The content and learning goals are the same across the thematic strands to ensure students gain content mastery regardless of which strand the teacher selects. In the section, how each of these design goals are achieved is presented.

THE SCRATCH ENCORE CURRICULUM

Scratch Encore comprises 14 modules, each covering a different topic of growing conceptual complexity. Each lesson is situated within a specific theme, with

the first 6 lessons having 3 themes the teacher can choose from: multicultural, youth culture, and gaming.

Each strand situates the content of the activities within a specific context designed to relate to learners while covering the exact same CS material (see Figure 1). In teaching the first 6 modules, teachers can decide which thematic strand they wish to use based on the theme and their students. Given the content is the same, teachers can move between strands as they progress from module to module.

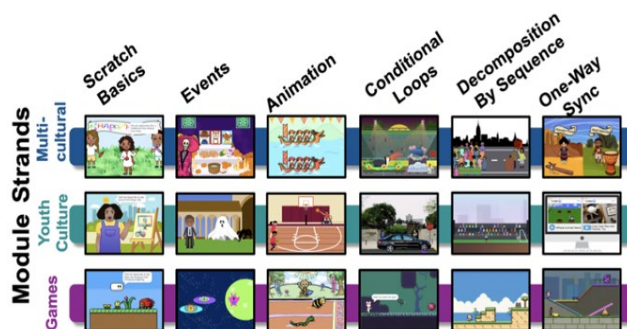


Figure 1: Scratch Encore module display of strands.

A feature of programming is that its concepts and practices are not tied to a specific context. Thus, it is possible to teach a concept, such as conditional loops, using different themes, such as waiting for a ride (transportation), Brazil's Carnival celebration, or a side-scrolling video game. Situating computing content in different themes can help students see their broad applicability and help them better relate to the content (Ladson-Billings, 2009). In this way, Scratch Encore draws on students' prior knowledge, cultural practices, and interests to help them connect with the content and feel a greater sense of belonging.

In terms of an overall pedagogical design approach, Scratch Encore's design draws from constructionist design principles which emphasize hands-on activities and give learners the opportunity to personalize the learning experience. To accomplish this, each module of the curriculum employs the Use→Modify→Create pedagogical approach. Use→Modify→Create is a research-backed, highly scaffolded progression of activities used to introduce learners to foundational computer science concepts (Lytle et al., 2019).

Learners begin by *Using* the focal CS concept before being asked to *Modify* a program that includes that

concept. Students do this by interacting with Scratch projects provided by the curriculum. Having seen the concept and modified its behavior, students then *Create* their own programs using the concept. Use→Modify lessons utilize TIPP&SEE, a novel learning strategy that helps students navigate the Scratch interface while learning from example projects.

The structure and thematic design of the *Use* and *Modify* projects provide a means for integrating cultural ideas into the curriculum. For example, students can be introduced to Events by using and modifying a Scratch project about creating an ofrenda for Día De Los Muertos or learning how to decompose a program by working on a project about soccer. This design provides a way to blend cultural and youth interest topics with computing content.

To facilitate the final, *Create*, projects, students are provided with a worksheet to help them brainstorm a topic, think through how they will use the focal concepts, and begin to plan out exactly what their scratch project will look like (see Figure 2). For example, to facilitate the *Create* project, students are provided creative prompts, like “What is your favorite sport?,” that empower them to tailor projects to their own interests and allow them to choose how to connect to their personal culture (Ryoo et al., 2013).

Creating with Events - Lesson 2

Objective: Today, I will create a Scratch project where sprites talk/think, change size, and move based on a topic of my choice.

Create a project about a topic you choose! Circle or highlight your topic choice or brainstorm your own.

- Favorite Holiday Christmas / Halloween
- or
- Family celebration We decorate our x-mas tree
- or
- Favorite place around your city Park
- or
- My topic Halloween

🎯 Planning Your Project:				
Use the Five W's to plan your project. Write your answers in the space provided. You may not need to use all five for your project.				Done
Who will be in the project (sprites)?	#1: <u>Witch</u>	#2: <u>Bat</u>	#3: <u>Frank</u>	✓
What are they doing? Say, Move, Change Size by ___ blocks	<u>save the bat from Frank</u>	<u>gets trapped by Frank</u>	<u>traps Bat to eat him</u>	✓
When? The events this sprite will respond to are: Choose at least two for each sprite. All three events need to be chosen at least once	<u>when clicked</u> <u>when this sprite clicked</u> <u>when key pressed</u>	<u>when clicked</u> <u>when this sprite clicked</u> <u>when key pressed</u>	<u>when clicked</u> <u>when this sprite clicked</u> <u>when key pressed</u>	✓
Where (Choose your Stage/Backdrop)?	<u>A haunted house</u>			✓
Why did you choose this? Say blocks	<u>I love halloween</u>	<u>I was a witch for my first halloween.</u>	<u>I love Frankinstien</u>	✓

Figure 2. Example Module 2 create worksheet.

This structured approach and the associated scaffolds (e.g., worksheets, provided Scratch projects) support teachers who are new to computer science. Additionally, the lessons are designed to be flexible so they can be modified based on available classroom time, the speed at which students are learning the material, and whether they are encountering the curriculum for the first time or reviewing it in a later year.

SCRATCH ENCORE’S STUDENT SUPPORTS

The Scratch Encore curriculum includes various resources and strategies to support students in learning to program and having a positive computer science learning experience. These supports include mnemonic devices to help with authoring, debugging, and submitting Scratch projects as well as worksheets to help them plan and execute their ideas for Scratch projects. These resources also support students in personalizing the learning experience and encourage them to bring their own interests, ideas, and values into the programs they are creating.

STUDENT WORKSHEETS

Writing programs is difficult. This includes both technical and conceptual challenges and deciding what the program will do and figuring out how to write the program to accomplish the envisioned goal. Scratch Encore provides worksheets to help students with each phase of the Use→Modify→Create process. Each Scratch Encore worksheet is available as a Google Doc, PDF, or Google Form.

In the *Use* phase, the Scratch Encore curriculum offers worksheets to help students with the TIPP&SEE process (discussed in the next section). This includes reminding students of each step in the process and asking specific questions about the project that require students to attend to different aspects of the project (e.g., the sprites, behaviors when different events occur). These worksheets help students learn how to *read* Scratch projects and make sense of their behaviors.

The *Modify* worksheets (see Figure 3) help students plan how they are going to modify the provided Scratch project. This includes having them think through the sprites they want to add/modify and the behaviors they want those sprites to carry out. Additionally, they include clearly defined tasks that

students mark as complete, so students can make sure they accomplish all they set out to do.

For students that need additional scaffolding, the curriculum also includes step-by-step instructions that walk students through each step of the *Modifying* process, allowing students to track progress to make sure they have done everything that needs to be done.

The *Create* worksheets (see Figure 2) provide planning prompts for students as they design a new project from scratch, defined tasks for programming, and reflection prompts. This worksheet helps students decide on a topic for their *Create* project and then think through how they are going to use the focal concept in their program. Additionally, it helps to make sure the resulting project includes the focal concept of the module.

Events - Modify Project

Strand: Multicultural

Objective: Today, I will modify a Scratch project to use different events to trigger actions.

🎯 Planning Your Project:	
Pick three special people who have passed away to be on your Ofrenda. Traditionally, people on Ofrendas are from our families, but you can also choose a famous person who has passed away.	Done <input type="checkbox"/>
Special Person 1: _____ Favorite Memory: _____	<input type="checkbox"/>
Special Person 2: _____ Favorite Memory: _____	<input type="checkbox"/>
Special Person 3: _____ Favorite Memory: _____	<input type="checkbox"/>

🔧 Modify Tasks:		
Setup:	Done	
<ul style="list-style-type: none"> • Reload, Remix, Share, and +Add to Studio (RoAR and hiSS) from the Modify project: https://scratch.mit.edu/projects/323737554/. • Choose costumes for Special Person 1, Special Person 2, and Special Person 3. 	<input type="checkbox"/>	
Now implement your plan for the <u>Left</u> Sprite:	Coded	Tested
<ul style="list-style-type: none"> • When this sprite clicked, it gets larger, says Special Person 1's name and favorite memory, then gets smaller. 	<input type="checkbox"/>	<input type="checkbox"/>
Now implement your plan for the <u>Middle</u> Sprite:		
<ul style="list-style-type: none"> • When this sprite clicked, it gets larger, says Special Person 2's name and favorite memory, then gets smaller. 	<input type="checkbox"/>	<input type="checkbox"/>
Now implement your plan for the <u>Right</u> Sprite:		
<ul style="list-style-type: none"> • Copy the 2 scripts from the Middle sprite. • When this sprite clicked, it gets larger, says Special Person 3's name and favorite memory, then gets smaller. 	<input type="checkbox"/>	<input type="checkbox"/>

Figure 3. Module 2 multicultural modify worksheet.

TIPP&SEE STRATEGY

TIPP&SEE is a scaffolding approach that uses reading comprehension strategies to help learners break down a Scratch project to understand how it behaves and why (see Figure 4). The approach is based on research teaching students how to read and is a key feature of the *Use* portion of the module

where students are being introduced to a new topic. Research on the TIPP&SEE strategy shows that it improves student learning, especially among students who have historically not succeeded in computing contexts, including students from racial groups historically excluded from computing contexts, students performing below grade level, and students with learning disabilities (Salac et al., 2021).

The TIPP portion guides students in previewing various components of a Scratch project before they examine any code. Students begin by finding key project features: *Title*, *Instructions*, and *Purpose*. Then they run the code while carefully observing events and actions. Students are encouraged to *Play* the project and accompanying code to see what it does. As students engage in purposeful play, they are prompted to interact with specific elements of a Scratch project to observe what happens after various events, such as 'click the green flag' or 'click on a sprite.'

Start with TIPP&SEE!				
Get a TIPP from the Project Page:				
Title: What is the title of the project? Does it tell you something about the project?				
Instructions: What do the instructions tell you to do?				
Purpose: What is the purpose of this activity? What will this code teach you?				
Play: Run the project and see what it does! Look at which sprites are doing the actions.				
What happened when you played the project? Circle or highlight the action(s) that happened for each event.				
When I clicked the green flag:				
talked waved did nothing	talked changed size did nothing	talked changed size did nothing	talked changed size did nothing	talked changed size did nothing
When I pressed the space bar:				
talked waved did nothing	talked changed size did nothing	talked changed size did nothing	talked changed size did nothing	talked changed size did nothing
When I clicked on the left picture :				
talked waved did nothing	talked changed size did nothing	talked changed size did nothing	talked changed size did nothing	talked changed size did nothing
When I clicked on the middle picture :				
talked waved did nothing	talked changed size did nothing	talked changed size did nothing	talked changed size did nothing	talked changed size did nothing

Figure 4. Module 2 TIPP&SEE worksheet.

The SEE portion helps students locate specific code related to the focal CS concept within the Scratch project (*Sprites* and *Events*). They click the See Inside button to explore the code that implements what they observed in the TIPP section. Then, students *Explore* the code to make suggested changes to the example code and note the project's behavior after the changes. The *Explore* questions on the TIPP&SEE

worksheets are designed to highlight specific things that students should notice before they start the Modify tasks.

TIPP&SEE outlines a systematic process for students to explore and understand a Scratch project and how the code functions through intentional experimentation and meaning-making.

WHAT?!? A MESS DEBUGGING STRATEGY

Debugging is an essential aspect of programming but is often overlooked in formal instruction. To support students in debugging their projects, Scratch Encore developed the *WHAT?!? A MESS* debugging strategy. The acronym outlines strategies for identifying what the bug is as well as potential causes and steps to resolve it.

Students start by identifying *What* the programmer intended the project to do and then find *How* the program is behaving differently than expected. Next, students *Analyze* sprite behavior and code. If a student cannot debug on their own, they are encouraged to *Talk* to their peers to find a solution.

While analyzing code, students look for A MESS that might be causing an issue in their code:

- *Arguments*: A number in a white circle is incorrect.
- *Missing*: Block(s) are missing.
- *Extra*: Extra block(s).
- *Scrambled*: Blocks are out of order.
- *Substitute*: Used the wrong block(s).

SCRATCH ENCORE’S TEACHER SUPPORTS

In an effort to be easily adoptable by teachers, Scratch Encore includes numerous resources to help teachers. All of the Scratch Encore resources and materials are organized into a single page for each module. Figure 5 shows the three Module 2 strands (depicted by the three colored columns) and the full suite of supporting materials, including videos, teacher resources, worksheets, and assessments. Each module has a page like this with all instructional materials in one place. These pages can be found at [Canon Lab](#) (n.d.) after completing a brief, free registration. The registration is used to keep track of who has reviewed the materials to share this information with stakeholders.

Module 02 - Events (M2)	Multicultural			Youth Culture			Gaming		
Unit Plans	Unit Plan Google Doc	Unit Plan pdf		Unit Plan Google Doc	Unit Plan pdf		Unit Plan Google Doc	Unit Plan pdf	
Lesson 1 (M2L1) - Exploring Events (60-120 minutes)	M2L1 Engage Video	YouTube		M2L1 Engage Video	YouTube		M2L1 Engage Video	YouTube	
	Teacher/Demo Project			Teacher/Demo Project			Teacher/Demo Project		
	Student Project			Student Project			Student Project		
	TIPP&SEE	pdf	Google Form	TIPP&SEE	pdf	Google Form	TIPP&SEE	pdf	Google Form
	TIPP&SEE Answer Key			TIPP&SEE Answer Key			TIPP&SEE Answer Key		
	L1 Modify Sheet	pdf	Google Form	L1 Modify Sheet	pdf	Google Form	L1 Modify Sheet	pdf	Google Form
	L1 Step-by-Step	pdf		L1 Step-by-Step	pdf		L1 Step-by-Step	pdf	
Dia de los Muertos Information Sheet			N/A			N/A			
Lesson 2 (M2L2)- Creating with Events (120 minutes)	L2 Create Sheet	pdf	Google Form	L2 Create Sheet	pdf	Google Form	L2 Create Sheet	pdf	Google Form
	Assessment			Assessment			Assessment		
	Google Doc			pdf			Google Form		
						Answer Key			

Figure 5. Module 2 strands and events.

LESSON PLANS AND CLASS RESOURCES

Each module in Scratch Encore includes a comprehensive unit plan, covering the focal computing content of the module, discussion prompts, details about the specific Scratch projects and activities students will work through. The lesson is also mapped to ISTE and CSTA standards (see Figure 5). The curriculum also includes posters related to topics like TIPP&SEE and WHAT?!? A MESS to help students as they work through the curriculum.

Each lesson plan begins with a high-level overview of the focal content of the lesson before outlining how each lesson in the module progresses. While not intended to serve as a script, the Unit plan does include prompts for opening discussion and closing reflections. It also includes links to the materials that are used in each lesson along with helpful pointers for the teacher to be aware of as they teach the lesson, including common questions students ask and challenges they face.

For each of the modules in the multicultural strand, teachers are provided with an information sheet about the specific theme used. For example, the multicultural theme used in Module 3 is dragon boat racing. The Dragon Boat Information Sheet provides a description of dragon boat races, their cultural significance, their history, and links to additional resources about the topic. Finally, the curriculum includes formative and summative assessments, including automated project assessments—discussed in greater detail in the next sections.

CONTENT VIDEOS

Each module includes a short (usually around 4 minutes) video introducing the core computing concept of the lesson. These videos can be shown by the teacher in class or shared with the students for them to review throughout the lesson if they need additional support. The videos use Scratch animations to define and demonstrate the core concept of each module.

ASSESSMENTS

The Scratch Encore curriculum provides both formative and summative assessments. For formative assessments, the *Use* and *Modify* worksheets (discussed in the Student Resources section) include multiple-choice and short-answer questions for students to answer. Answer keys are provided for both sets of worksheets.

For summative assessments, each module includes a quiz that can be administered at the end of the module asking students questions about the focal topic of the lesson. The quizzes are multiple-choice and often include images of Scratch projects and scripts that students have to interpret (see Figure 6). Like the worksheets, the summative quizzes are available in Google Doc, Google Form, and PDF formats.

Finally, Scratch Encore provides teachers with an automated Scratch project checker that allows them to check to see if a student's *Create* project includes the required content. To use the [Scratch Encore Student Project Checker](#) (n.d.), you input the URL for the Scratch project and it reports which required aspects of the *Create* project were implemented. This tool can also be shared with students so they can check their own progress as they work on their *Create* projects. Additionally, teachers can use the [Scratch Encore Studio Grader](#) (n.d.) to monitor multiple students' work at a time (see Figure 7).

For this tool, the teacher inputs the Studio URL where student projects are stored. The tool runs the Student Project Checker on each program, providing results for the whole class. It is recommended to double-check students' work if the checker thinks something is missing because students can use creative solutions the checker does not anticipate.

Use the script below to answer questions 3a, 3b, 3c, and 3d.

3a. Circle or highlight: What do you do to make the script run?

- Click the green flag
- Click the sprite
- Press the space key

What does the sprite do when the script runs? Circle or highlight your answers.

3b. First, the sprite _____.

- moves 10 steps
- changes costume
- says "Let's play!"

3c. Next, the sprite _____.

- moves 10 steps
- changes costume
- says "Let's play!"

3d. Last, the sprite _____.

- moves 10 steps
- changes costume
- says "Let's play!"

Figure 6. A Module 2 summative assessment section.

results:

6 done	7 almost done	5 need time or help
User: awang21 Project ID: 238947322 ✓ - Changed car sprite. ✓ - Car stops upon collision. ✓ - Car says something. ✓ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✗ - Car makes a sound.	User: awang21 Project ID: 238947438 ✗ - Changed car sprite. ✓ - Car stops upon collision. ✗ - Car says something. ✗ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✗ - Car makes a sound.	User: chopcraft Project ID: 238469984 ✓ - Changed car sprite. ✓ - Car stops upon collision. ✓ - Car says something. ✓ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✗ - Car makes a sound.
User: chopcraft Project ID: 238470041 ✗ - Changed car sprite. ✓ - Car stops upon collision. ✗ - Car says something. ✗ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✓ - Car makes a sound.	User: chopcraft Project ID: 238470132 ✗ - Changed car sprite. ✗ - Car stops upon collision. ✗ - Car says something. ✗ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✗ - Car makes a sound.	User: chopcraft Project ID: 238469635 ✓ - Changed car sprite. ✓ - Car stops upon collision. ✓ - Car says something. ✓ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✓ - Car makes a sound.
User: chopcraft Project ID: 238469748 ✓ - Changed car sprite. ✓ - Car stops upon collision. ✓ - Car says something. ✓ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✓ - Car makes a sound.	User: chopcraft Project ID: 238469228 ✓ - Changed car sprite. ✓ - Car stops upon collision. ✓ - Car says something. ✓ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✗ - Car makes a sound.	User: chopcraft Project ID: 238470103 ✗ - Changed car sprite. ✗ - Car stops upon collision. ✗ - Car says something. ✗ - Changed car speed. Extensions: ✓ - Other sprites perform actions. ✓ - Car makes a sound.

Figure 7. Studio grader example.

SCRATCH ENCORE MODULES 1-6 OVERVIEW

In this section, the first 6 modules are presented in greater detail to help teachers understand the content and pacing of the curriculum. As a reminder, each of these first 6 lessons can be taught using three different themes: multicultural, youth culture, or gaming.

MODULE 1 - SCRATCH BASICS

Module 1 has three lessons.

Length: Lesson 1: Introducing Scratch (60 minutes), Lesson 2: Exploring Scratch (60 minutes), Lesson 3: Creating with Scratch (60 minutes)

Overview: This module orients students to the Scratch programming environment by introducing color-coded block categories, block shapes, and the basic components of a Scratch project (e.g., sprites, backdrops). Students explore the Scratch environment in a scavenger hunt activity, modify an existing project, and create a new project of their own design.

Objectives: Students will be able to:

- Reload, remix, share, and add Scratch Projects to a Studio.
- Add a backdrop to a blank Scratch project.
- Create and delete sprites.
- Start projects with the green flag.
- Stop projects with the stop sign.
- Add text to a sprite using say__for__secs.
- Add movement to a sprite using move__steps.
- Place a sprite in a specific location using go to x:_y:_.
- Create a project that includes text and step movement.

Module Themes:

- Multicultural: Holi celebration
- Youth culture: Communities
- Gaming: Helen the Hedgehog

MODULE 2 - EVENTS

Module 2 has two lessons.

Length: Lesson 1: Use/Modify (60-120 minutes), Lesson 2: Create (120 minutes)

Overview: In event-based programming, scripts are triggered when specific events occur (e.g., a sprite is clicked, a key is pressed). In this module, students learn to use a variety of events to trigger scripts that resize sprites or cause a sprite to say something.

Objectives: Students will be able to:

- Define an event.
- Write scripts to run when a specific key is pressed.
- Write scripts to run when a specific sprite is clicked.
- Change and set the size of sprites.
- Copy a script from one sprite to another.

Module Themes:

- Multicultural: Día De Los Muertos Ofrenda
- Youth culture: Fantasy - Wizard and Ghost
- Gaming: Race in Space

MODULE 3 - ANIMATION

Module 3 has two lessons.

Length: Lesson 1: Use/Modify (120 minutes), Lesson 2: Create (120 minutes)

Overview: A sprite can be animated by repeating sets of blocks that alternate between switching costumes or moving to the next costume along with repeated step movement. Adjusting the wait times of these repeated actions will increase or decrease the speed of the animation. Naming each of the costumes and changing the size of the sprites can assist in the creation of the animation.

Objectives: Students will be able to:

- Animate a sprite at one location using a repeat loop and multiple costumes.
- Animate a sprite with movement using a repeat loop.
- Distinguish between the functionality of switch costume and next costume blocks.

Module Themes:

- Multicultural: Dragon Boat Festival
- Youth culture: Basketball
- Gaming: Animal Races

MODULE 4 - CONDITIONAL LOOPS

Module 4 has two lessons.

Length: Lesson 1: Use/Modify (120 minutes), Lesson 2: Create (120 minutes)

Overview: Within a conditional loop, repeated actions will continue until a specific condition becomes true. In Scratch, the condition that is fulfilled can relate to another sprite or a specific color. In this module, students will implement conditional loops that control the repeated actions of sprites. The sprites will stop moving when the condition of the loop becomes true due to a sprite reaching a color or another sprite.

Objectives: Students will be able to:

- Create an animation with one costume and repeated movement.
- Write a script using a conditional loop that repeats an action until a condition is true.

Module Themes:

- Multicultural: Carnival Parade
- Youth culture: Waiting for a Ride
- Gaming: Ninja Cat & Gem

MODULE 5 - DECOMPOSITION BY SEQUENCE

Module 5 has two lessons.

Length: Lesson 1: Use/Modify (120 minutes), Lesson 2: Create (120 minutes)

Overview: A series of actions by multiple sprites can be decomposed based on the order in which the actions occur and the conditions that cause one action to stop and another to start. To decompose a project into a sequence of actions, it is necessary to identify the actions that occur based on certain events. In this module, students identify a set of events based on sensing conditions (e.g., when one sprite touches another sprite) and the actions that occur as a result of those sensing conditions.

Objectives: Students will be able to:

- Decompose a sequence of events.
- Create scripts that will trigger the action of one sprite dependent on the action of another sprite.
- Use sensing blocks to stop and start actions.

- Plan and create an animation based on a set of events and actions.

Module Themes:

- Multicultural: Protest Marchers
- Youth culture: Soccer
- Gaming: Fix the Game

MODULE 6 - ONE-WAY SYNCHRONIZATION

Module 6 has two lessons.

Length: Lesson 1: Use/Modify (120 minutes), Lesson 2: Create (120 minutes)

Overview: Synchronization is the coordination and timing of actions *between sprites*. Previously, events were all directly from the user. However, a single sprite is limited in what events it can sense. For example, a sprite cannot sense when a user clicks on a different sprite. In one-way synchronization, one sprite sends an invisible message to other sprites, thereby initiating an event. This is called message passing - a single communication event between two or more sprites takes place in which one sprite passes a message, and one or more sprites receive the message and respond accordingly.

Objectives: Students will be able to:

- Create scripts that use message passing for one-way synchronization between two sprites.
- Create scripts that use message passing for one-way synchronization that initiate simultaneous actions on multiple sprites.

Module Themes:

- Multicultural: Music, Navajo flute and Djembe
- Youth culture: Encore TV
- Gaming: Remote Control Cars

DEEP DIVE - MODULE 2: EVENTS

During this section, italic text identifies questions or prompts for the learners.

Having presented the content of the first 6 modules, a more detailed look into one module is provided. The goal here is to provide a greater sense of exactly what it is like to teach Scratch Encore. For this

section, Module 2: Events was chosen. In this module, students are introduced to event-based programming. They identify the events that sprites should respond to and program the actions that should be carried out if that event occurs.

There are two lessons in the Events module: Lesson 1: Exploring Events (60-120 minutes) and Lesson 2: Creating with Events (120 minutes). In both Lessons 1 and 2, teachers can select from three themes to use in their classroom: multicultural, youth culture, or gaming. Following instruction, the module assessment uses multiple-choice questions to gauge students' mastery of using events to trigger an action.

To gain access to the specific worksheets and materials for this module, register for free at [Canon Lab](#) (n.d.).

MODULE 2 THEMES

Module 2 has three themes' teachers can choose from: multicultural, youth culture, or gaming.

Multicultural Theme: Students use and modify a project that depicts a Día de los Muertos ofrenda. They begin with an ofrenda featuring grandparents, a family pet, and offerings. When picture frames are clicked their inhabitants grow and share a special memory. Students can also use the spacebar to learn more about the holiday. After exploring the events in the teacher/demo project, students modify the ofrenda to include people who are important to them and a favorite memory. Accompanying the lesson materials is an Information Sheet about Día de los Muertos, ofrendas, and the cultural significance of the holiday in Mexican culture.

Module 2 Youth Culture Theme: Students use and modify a project that depicts a wizard and a ghost. When characters are selected, they grow, and the spacebar causes the wizard to cast a spell shrinking the ghost. After exploring the events in the teacher/demo project, students modify the scene to make the ghost grow when clicked and code a previously unused cat sprite.

Module 2 Gaming Strand: Students use and modify a project that depicts two aliens, Bop and Beep, in flying saucers who are racing to save a third alien, Bork. When clicked, the saucers grow and provide directions for the race. The right arrow key causes

Bop to move and the left arrow key causes Beep to move. Students can use the arrow keys to decide who saves Bork. After exploring the events in the teacher/demo project, students modify the race to change the text for Bop and Bork. Students also program events for aliens to grow and speak using the right arrow key, mouse click, and green flag.

MODULE 2 DEEP DIVE

LESSON 1: EXPLORING EVENTS

In this lesson, the concept of Events in programming is introduced. Students create and modify scripts that handle basic events. A starter project is provided with sprites and example code that handles an event and then responds with an action. For this deep dive, the Día de los Muertos materials from the multicultural strand is featured.

ENGAGE (20-30 MINUTES)

Introduction (10-15 minutes)

The lesson kicks off with an introductory discussion (10-15 minutes) that is designed to introduce the focal concept in a way that connects with students' everyday lives. To start, pose questions like, *What is an event? What do you do if someone taps you on the shoulder?* Through this discussion, students are introduced to events in programming and how they are used in Scratch. An animated video of the [Events Engage discussion](#) (Scratch Encore, 2020) is available for students who were absent, those who benefit from multiple exposures to content, or for online/asynchronous instruction.

TIPP&SEE (10-15 minutes)

The next portion of the lesson reintroduces students to the TIPP&SEE strategy (10-15 minutes). Announce that you will model part of TIPP&SEE and students will finish the rest themselves. Ask students if they remember what TIPP&SEE is from the previous lesson and explain they will begin looking at the project as a class first. Distribute the Module 2 TIPP&SEE worksheet and, if possible, display the [Ofrenda Student Modify Project](#) (ScratchEncore, 2019a; see Figure 8).

Ask students questions to lead them through TIPP&SEE, such as: *What does the title tell us? What*

is the purpose of looking at this code? What are the sprites' names and pictures? Share a new Scratch skill with students: copying scripts from one sprite to another with drag and drop. Select the Middle sprite. Drag the 'when green flag clicked' script from Middle to Right. Now those blocks are also in Right's scripts.

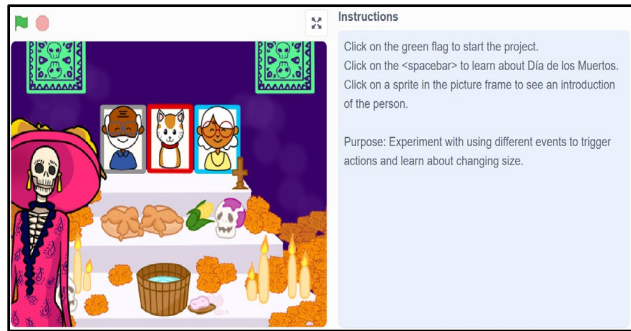


Figure 8. A Module 2 Ofrenda student modify project.

EXPLORE (30-65 MINUTES)

TIPP&SEE (10-20 minutes)

Provide students time to use the TIPP&SEE strategy alone or with a partner (10-20 minutes). They should work through the TIPP&SEE strategy in more detail on their own.

Modify Ofrenda Project (20-45 minutes)

Next, students modify the Ofrenda Student Modify Project (20-45 minutes), using the Events Modify worksheet as a guide. A step-by-step worksheet is available as a means to support differentiation for students who need additional assistance. In this activity, students will copy the script from Left sprite to Middle and Right sprites, as well as personalize the project by selecting costumes and modifying the say blocks for all three sprites. Remind students to save and share their work before exiting Scratch.

You can also offer extension challenges to students. These challenges include changing the scripts that handle events to see how the functionality changes and adding scripts to handle additional events.

REFLECT (10-25 MINUTES)

Students answer the reflection questions on the Events Modify worksheet (5-10 minutes). They will answer questions, such as: *How would you explain what an event is in computer programming? How is*

having different types of Events blocks helpful when coding?

Consider having students present their projects to the class (5-15 minutes). You can also have students participate in a gallery walk where students run each other's projects on individual computers. If there is time, consider leading a discussion about how events could be used in their projects.

LESSON 2: CREATING WITH EVENTS

During the Create portion of Lesson 2, students use events as they create a brand new Scratch project about a topic of their choice.

ENGAGE (15 MINUTES)

The lesson opens with a demonstration. Ask if anyone remembers what an event is in programming and if they remember which three event blocks were used in the last lesson. Display the [Events: Ofrenda Sample project](#) (ScratchEncore, 2019b), click See Inside, and then click on the Left sprite to reveal the blocks. Show what happens when you click the left sprite (on the stage) a few times (the sprite gets bigger, introduces itself, and shares a memory). Ask: *What block makes the sprite bigger each time the sprite is clicked?*

Click on the sprite again and stop the project using the stop sign while the sprite is talking so that it is stopped while it is big. Then, click the green flag to show that the sprite returns to its original size.

Ask the following questions to clarify the difference between: change size by ___ and set size to ___%: *What block makes the sprite return to its original size? How are change size by ___ and set size to ___% different?*

Demonstrate these things enough times that it becomes clear that set size to ___% always sets the sprite to the same size while change size by ___ continually makes a sprite bigger (or smaller) each time an event occurs.

EXPLORE (80 MINUTES)

In this portion of the lesson, students will create their own, new Scratch project using events (80 minutes). Introduce the activity by telling students they will

create a project using events. Distribute the Events Create worksheet and give students guidance throughout the project. This worksheet facilitates coming up with a Scratch project theme and planning out the sprites, events, and actions of the project.

To complete the task, students will:

- Create a new Scratch project.
- Brainstorm content for the project and plan how the project is going to be implemented.
- Add a backdrop.
- Select three sprites.
- Write scripts to handle all three types of events:
 - When green flag clicked
 - When this sprite clicked
 - When <key> Pressed
- Use say___ for___sec blocks to introduce themselves, say something about a topic, and describe what 'event' will cause them to move.
- Use change size by___ to change size, set size to ___% to initialize it when green flag clicked.

Explain to students that they can create any project of interest, but some ideas include:

- Describing a holiday you like to celebrate.
- Describing a family celebration and what makes it unique or special to your family.
- Introducing a favorite place around your city.
- A topic of your choice!

Remind students to run their project to test that it works as expected or debug and re-test as necessary. Ensure students Share and Add to Studio before exiting Scratch.

You can also offer an extension challenge to students. Have students add scripts to handle additional events (e.g., students make the sprites move when an event occurs; sprites get bigger when one event occurs but then get smaller when a different event occurs).

REFLECT (10 MINUTES)

Students should finish the reflection questions included on the Events Create worksheet (10 minutes). The reflection includes questions like, *What was challenging about this project? How did you work through the challenges you faced? What is the difference between the change size by ___ and set size to ___% blocks?*

Afterward, consider having students present their projects to the class (15 minutes). You can also have students participate in a gallery walk where students run each other's projects on individual computers. If there is time, lead a discussion about how events could be used in other projects (e.g., map movements in a game to arrow keys, map different sounds to each sprite when it is clicked).

ASSESSMENT & DIFFERENTIATION

Teachers and students can utilize the [Scratch Encore Student Project Checker](#) (n.d.) to make sure the project has all the required elements. The teacher can also use the [Scratch Encore Studio Grader](#) (n.d.) to monitor students' progress across the class. While the Automated Project Checker does a good job of reading student code for including specific tasks, it is recommended that teachers double-check submissions before use in grading, as sometimes students complete the task in a way the checker cannot anticipate.

Following Module 2, teachers can administer an end-of-module assessment. The Module 2 assessment consists of 6 multiple-choice questions related to events in Scratch. Students are required to read sample scripts to identify which event will cause a specific action, which scripts are initiated by a specific event, and what follows after a script is initiated.

The Events Module also offers various forms of differentiation. You can prepare block definition sheets for students to add to their journals. Options include: list the block names (students fill in the definitions) or provide the block names and definitions (students match block names to definitions). You can also provide a list of the type and number of blocks needed to complete each activity.

CRITICAL REFLECTIONS

LESSONS LEARNED

The Scratch Encore curriculum has been under active development since 2017. Over the last seven years, we have worked with countless teachers and students to create what we think is a high-quality,

easily adopted, and effective curriculum. Further, the curriculum has seen relatively widespread adoption, having been downloaded over 2,500 times by teachers and the Scratch projects from Module 1 having been viewed over 35,000 times and remixed over 11,000 times. Additionally, significant research has been conducted on the curriculum resulting in over 15 publications in peer-reviewed academic venues related to teacher experiences and student outcomes.

Over the course of the project, several key insights have helped make the curriculum what it is today. First, a key insight is the importance of providing a high level of support for teachers while also not being overly prescriptive. In this way, the curriculum can be taught by those new to computer science but also still provide ample room for teachers to personalize the instructional approach based on their own expertise, experience, and knowledge of the students. A good example of what this looks like in the curriculum is the inclusion of short videos that introduce the core concept of each module and specific prompts for leading class discussions. Some teachers may rely heavily on these resources while others may not use them at all.

A second key insight is the importance of providing step-by-step scaffolds for learners to help them through the Modify and Create activities. It is not enough to just help them with programming but also idea generation and mapping their ideas onto the core constructs of the lesson. Our research has shown that these worksheets helped students create more complex and correct Scratch programs.

A final insight from this work has been how valuable it is to have teachers and district leaders as partners in the design of classroom-ready curricula. Since the project began, we have had bi-weekly meetings with our district partners. Additionally, teachers and students have been close collaborators throughout the process. This has helped us create a curriculum that is easily adoptable and can work with a wide array of classroom and school contexts. This flexibility can be seen in the modular structure of the lessons, the fact that materials are available in multiple formats (e.g., PDF, Google Docs, Google Forms), and various supplementary supports and information, including standards mappings and information sheets for the multicultural strand modules.

All that being said, there are challenges associated with adopting this curriculum. Central among them is the need for dedicated classroom time to teach the materials. Scratch Encore has a fixed, regular structure and cumulative design that incrementally builds students' computer science knowledge and skills. In contexts where teachers have little time for computing instruction or instructional times come at irregular intervals, teaching Scratch Encore has been found to be challenging.

CULTURAL RELEVANCE

One of the three focal design goals of this curriculum was to be culturally responsive. Several design decisions were made to accomplish this goal and support a diversity of students. First, the inclusion of multiple thematic choices for each module allows the teacher to tailor their instruction to the cohort of students they have. These learning modules are drawn from several sources, including projects that speak to students of specific cultural backgrounds as well as ideas generated by several participatory co-design sessions involving parents, students, and educators.

Second, we provide opportunities for students to express themselves through their projects. Our techniques range from customizing minor elements of projects (choosing what sprites say in modified projects) to the Create projects which have many dimensions on which students can be creative.

Finally, care was taken to make sure the sprites used in the shared Scratch projects reflect the racial, cultural, and gender diversity of the students we hope to present in classrooms. For example, the multicultural strand projects highlight cultural practices and celebrations from around the world.

Different students have different learning needs. Therefore, we have included support for a variety of learners. We provide differentiation through extension exercises and open-ended create activities. We also provide scaffolding through graphic organizers and activity sheets that support student planning through fill-in-the-blank diagrams. Finally, we include a new strategy we developed called TIPP&SEE to help students navigate the complex Scratch interface and focus on the learning goals of each activity.

FUTURE DIRECTIONS

HARMONIZING SCRATCH ENCORE

While the Scratch Encore curriculum is complete, we are still actively working on it as a research project. Our current work is to explore ways to empower teachers to create their own versions of Scratch modules that draw on the prior knowledge and cultural practices present in their specific classrooms, schools, and communities. We call this process *Harmonizing* (Tran et al., 2024).

To *Harmonize* a lesson, teachers create a new Scratch project and accompanying student worksheets. Like with the curriculum itself, we have developed a series of supports to facilitate both coming up with theme ideas and then creating the materials for the *Harmonized* module. To support this process, we are also developing a generative AI tool, called Conjurer.

USING GENERATIVE AI TO SUPPORT HARMONIZING

To facilitate the *Harmonizing* process, we have explored using publicly available generative AI to support teachers in the project ideation phase of the *Harmonizing* process. With a simple prompt template, a teacher can interact with GPT-3 to brainstorm multiple *Harmonized* project ideas. Many GPT-generated ideas can be efficiently remixed from a sample Scratch Encore project without major changes in the project description (Tran et al., 2025). Further, we are developing a streamlined, generative-AI-powered tool to help with every step of the *Harmonizing* process: from brainstorming project ideas to implementing complete instructional materials, including Scratch projects (code, images) and student-facing materials (Figure 9). The goal is to enable teachers to *Harmonize* a full Scratch Encore module in 20 minutes or less. These *Harmonized* materials can be taught in place of existing projects in the Scratch Encore curriculum, allowing a teacher to leverage the existing materials for other lessons and continue to use the auto-grading features: the [Scratch Encore Student Project Checker](#) (n.d.) and the [Scratch Encore Studio Grader](#) (n.d.).

PROFESSIONAL DEVELOPMENT

To introduce teachers to the *Harmonizing* process, we are creating a new week-long professional development workshop focused on culturally responsive pedagogy, design strategies to support it, and an introduction to customizing Scratch Encore modules. The PD covers culturally relevant design principles in general as well as several common customization categories (e.g. literature, science, community), followed by instruction on how to match a theme to a particular activity and the process of customizing a Scratch Encore module. Teachers will leave the PD with at least two customized projects and the knowledge to quickly make more.

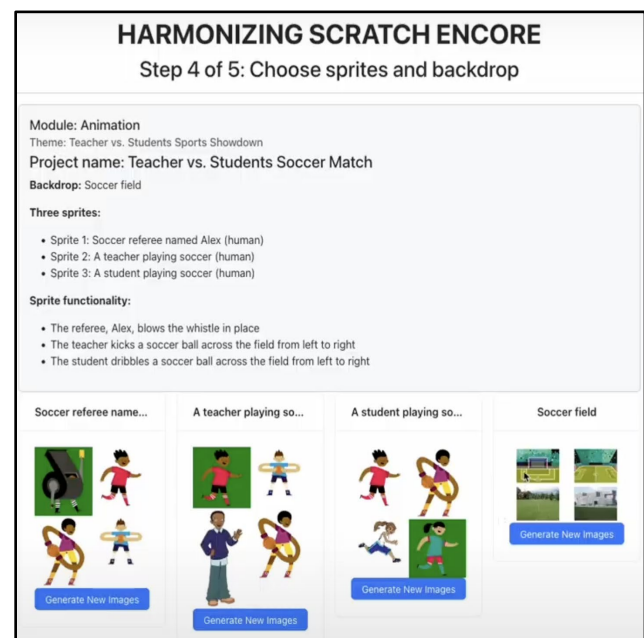


Figure 9. AI generator example.

REFERENCES

- Canon Lab. (n.d.). *Scratch Encore*. Retrieved May 17, 2025, from <https://www.canonlab.org/scratchencoremodules>
- Computer Science Teachers Association. (2017). K-12 Standards. <http://www.csteachers.org/standards>
- Coenraad, M., Palmer, J., Franklin, D., & Weintrop, D. (2019). Enacting identities: Participatory design as a context for youth to reflect, project, and

- apply their emerging identities. *IDC '19: Proceedings of the 18th ACM International Conference on Interaction Design and Children*, 185-196.
<https://doi.org/10.1145/3311927.3323148>
- CSforAll. (n.d.). *Home*. <https://csforall.org/>
- edX. (n.d.). *UChicagoX: Teaching coding in grades 5-8 with Scratch Encore*. Retrieved May 17, 2025, from <https://www.edx.org/learn/teacher-training/university-of-chicago-teaching-coding-in-grades-5-8-with-scratch-encore>
- Franklin, D., Coenraad, M., Palmer, J., EATINGER, D., Zipp, A., Anaya, M., White, M., Pham, H., Gökdemir, O., & Weintrop, D. (2020). An analysis of use-modify-create pedagogical approach's success in balancing structure and student agency. *Proceedings of the 2020 ACM Conference on International Computing Education Research*, 14-24.
<https://doi.org/10.1145/3372782.3406256>
- International Society for Technology in Education. (2016). *ISTE standards: For students*.
<https://iste.org/standards/students>
- Ladson-Billings, G. (2009). *The dreamkeepers: Successful teachers of African American children* (2nd ed.). Jossey-Bass.
- Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A., Isvik, A., Bounajim, D., Wiebe, E., & Barnes, T. (2019). Use, modify, create: Comparing computational thinking lesson progressions for STEM classes. *ITiCSE '19: Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 395-401. <https://doi.org/10.1145/3304221.3319786>
- Papert, S. (1980). *Mindstorms. Children, computers and powerful ideas*. Basic books.
- Ryoo, J., Scott, S., & D'Angelo, C. (2013). Culturally responsive computing education: Bridging the gap between theory and practice. *In Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 235-240.
- Salac, J., Thomas, C., Butler, C., & Franklin, D. (2021). Supporting diverse learners in K-8 computational thinking with TIPP&SEE. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 246-252.
<https://doi.org/10.1145/3408877.3432366>
- Scott, S. D., Sheridan, J., & D'Angelo, C. M. (2015). Culturally responsive computing instruction. *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 432-437.
- Scratch Encore. (2020, August 6). *Scratch Encore module 2 engage: Events* [Video]. YouTube.
<https://youtu.be/zdiltngtpeuY?si=8w-VuCs5s7ePF654V>
- Scratch Encore Student Project Checker. (n.d.). Retrieved May 17, 2025, from <https://people.cs.uchicago.edu/~dmfranklin/assessment/automated-assessment-master/projects.html>
- Scratch Encore Studio Grader. (n.d.). Retrieved May 17, 2025, from <https://people.cs.uchicago.edu/~dmfranklin/assessment/automated-assessment-master/index.html>
- ScratchEncore. (2019a, August 9). *Events: Ofrenda - student modify*. Scratch.
<https://scratch.mit.edu/projects/323737554/>
- ScratchEncore. (2019b, August 9). *Events: Ofrenda teacher sample*. Scratch.
<https://scratch.mit.edu/projects/323737287/>
- Tsan, J., EATINGER, D., Pugnali, A., Gonzalez-Maldonado, D., Franklin, D., & Weintrop, D. (2022). Scaffolding young learners' open-ended programming projects with planning sheets. *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education*, 372-378.
<https://doi.org/10.1145/3502718.3524769>
- Tran, M., Gonzalez-Maldonado, D., Zhou, E., & Franklin, D. (2025). Can GPT help? Supporting teachers to brainstorm customized instructional scratch projects. *SIGCSE 2025: Proceedings of the 56th ACM Technical Symposium on Computer Science Education*, 1, 1134-1140.
<https://doi.org/10.1145/3641554.3701858>
- Tran, M., Killen, H., Palmer, J., Weintrop, D., & Franklin, D. (2024). Harmonizing Scratch Encore: Scaffolding K-8 teachers in customizing culturally responsive computing materials. *SIGCSE 2024: Proceedings of the 55th ACM Technical Symposium on Computer Science Education*, 1, 1134-1140.

Symposium on Computer Science Education, 1,
1335-1341.
<https://doi.org/10.1145/3626252.3630756>

SUPPORT MATERIALS

Scratch Encore Public Assessment Links. (n.d.). Retrieved May 17, 2025, from <https://docs.google.com/document/d/e/2PACX-1vQdDrteCGMOUFIE7zn0hbTOTnyARJSBGmbrCBLb00YQpunn2XRimR50OHobQY0uLFctnCmBMkBGy5b/pub>

Scratch Encore Student Project Checker. (n.d.). Retrieved May 17, 2025, from <https://people.cs.uchicago.edu/~dmfranklin/assessment/automated-assessment-master/projects.html>

Scratch Encore Studio Grader. (n.d.). Retrieved May 17, 2025, from <https://people.cs.uchicago.edu/~dmfranklin/assessment/automated-assessment-master/index.html>

ABOUT THE AUTHORS

Rasha Alkhateeb is a PhD Candidate in Literacy Education and a Writing Fellow at The Graduate School's Center for Writing and Oral Communication at the University of Maryland, College Park.

J. Elisabeth Kasner is an Assistant Professor of Elementary STEM Education in Anne's College at Florida State University. Her research focuses on the processes of science teacher identity and learning, elementary and middle level education, and integrated STEM+C.

David Weintrop is an Associate Professor and the Dean's Impact Professor in the Department of Teaching & Learning, Policy & Leadership in the College of Education with a joint appointment in the College of Information at the University of Maryland. His research focuses on the design, implementation, and evaluation of effective, engaging, and equitable computational learning experiences.

Jen Palmer is a Curriculum Specialist at the University of Chicago. For over 15 years, she has supported high-quality science, engineering, and computer science instruction in K-8 classrooms. Her work has ranged from curriculum and tool

development to the design and facilitation of professional development opportunities and classroom coaching for teachers.

Merijke Coenraad is a Program Director focused on Inclusive Computing Research at Digital Promise. Her area of expertise includes co-design, research-practice partnerships, and equity in emerging technologies.

Minh Tran is a PhD student in Computer Science at the University of Chicago.

Diana Franklin is an Associate Professor in Computer Science at the University of Chicago where she leads the CANON (Computing for ANYONE) Lab. CANON Lab researches both 3rd-8th grade computer science interventions and quantum computing education for novices of any age with a particular focus on fostering more equitable learning experiences.

ACKNOWLEDGEMENTS

National Science Foundation grant assistance (Award #2201312) helped authors to produce article activities.

SHARING & MODIFICATION PERMISSIONS

Unless otherwise noted, this article and its resources are published under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license](https://creativecommons.org/licenses/by-nc-sa/4.0/):



You can freely share the article and its resources if you indicate the original authors, identify the Creative Commons license, and use them non-commercially.

You may also make and share modifications by:

- [Identifying the original authors](#).
- Using the resources non-commercially.
- Licensing modifications under the CC BY-NC-SA 4.0 license (and including a link to it).
- Indicating what modifications were made.