Raspberry Pi Foundation

Scaffolding children's development of technical and ethical computing competencies

Jean Salac (University of Washington)

Salac, J. (2023). Scaffolding children's development of technical and ethical computing competencies. In: Primary (K–5) computing education research – teaching and teachers. Understanding computing education (Vol. 4). Proceedings of the Raspberry Pi Foundation Research Seminars.

Available at: rpf.io/seminar-proceedings-vol-4-salac

Scaffolding children's development of technical and ethical computing competencies

Jean Salac (University of Washington)

Abstract

With many countries worldwide integrating computing instruction at the primary school level, it is crucial that we ensure that computing instruction is effective for all students. In this chapter, I discuss my work on identifying inequities in elementary computing instruction and in developing a learning strategy, TIPP&SEE, to address these inequities. Students using TIPP&SEE demonstrated improved understanding of computing concepts and better code quality in assignments. Further, the gaps between students with and without academic challenges narrowed when using the TIPP&SEE strategy.

Additionally, I outline the next steps for my work, expanding my research on computing education for youth to not only include programming, but also the societal and ethical impacts on computing. Currently, I am researching how youth may learn to examine technology's role in their lives and society, and how educators can foster in youth a critical understanding of computing. In a recent study, we observed that children were capable of reasoning around both explicit and implicit effects of algorithmic bias, grounded in their lived experiences and situated knowledge. Through my work, I aim to support youth in developing both technical and ethical competencies in computing, so that they can be active and critical participants for a more just computing future.

Access to computing instruction in schools isn't enough

Instructional context

The study that spurred my early research agenda took place in the San Francisco Unified School District (SFUSD) in San Francisco, CA, USA. At the time, the SFUSD was in their second year of their large-scale implementation of computing instruction in all their primary and secondary schools. Our study involved nine fourth-grade (ages 9–10) classrooms involved in the SFUSD's implementation: three classrooms from three schools identified as high-, mid-, and low-performing by the SFUSD for a total of 204 students.

All teachers taught the same introductory computing curriculum in Scratch, which was a modification of the Creative Computing Curriculum (Balch et al., n.d.). Students completed three modules in that curriculum — Module 1 covered an introduction to the Scratch platform, Module 2 covered sequence and events, and Module 3 covered loops. Upon completion of Modules 2 and 3, students took a pen-and-paper assessment. It consisted of multiple-choice, fill-in-the-blank, and open response questions, asking students to perform a variety of tasks related to events and loops summarized below:

- Identify the scripts that would run when they clicked a sprite
- Identify the best descriptions of sprite actions based on code
- Identify how many times the loop in question would run
- Identify the unrolled code that accomplishes the same actions as the loop provided in question

- Identify the scripts that do the same actions, among several options of sequential and repeated code
- Identify the code that runs before, in, and after a loop
- Explain in your own words what a loop in question would do
- [Extra Credit] Identify how many times the code in the inner loop of a nested loop would run

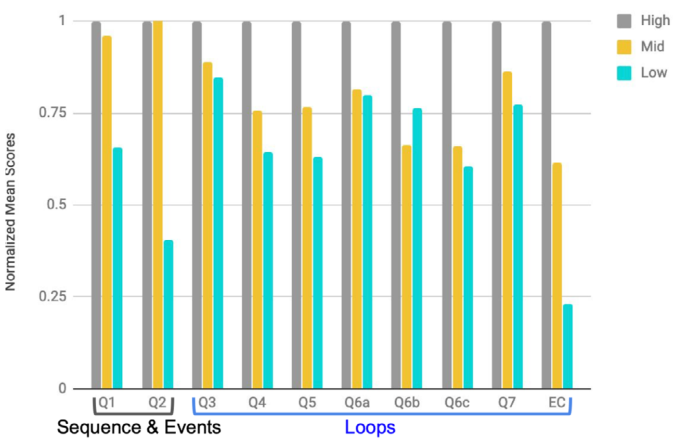

Figure 1. Normalised mean scores of high-, mid-, and low-performing schools on computing assessments covering, sequence, events, and loops.

Findings

Figure 1 depicts the mean scores for each question, normalized with respect to the high-performing school. Students in high-performing schools, in general, showed a good understanding of sequence, events, and loops. However, our results showed that students at mid- and low-performing schools exhibited a much shallower understanding of loops. While they could specify how many times a repeat loop will iterate (Q3), fewer than half could identify the unrolled equivalent of a repeat loop (Q4) and identify both constructs that repeat actions (Q5; repeat loop and sequential code). Overall, there were statistically significant differences between students in the high- and low-performing schools on all questions, and the mid- and low-performing schools on Q5, Q6, and Q7. While all students benefited from access to computing instruction in their schools, these performance differences indicated a need for pedagogical and curricular improvements to support struggling students (Salac et al., 2019).

TIPP&SEE: Scaffolding programming learning for equitable outcomes

Designing the TIPP&SEE learning strategy

Motivated by the inequitable performance disparities we observed, we designed TIPP&SEE, a learning strategy that scaffolds student exploration of provided programs for activities on the Use → Modify step of the Use → Modify → Create progression (Lee et al., 2011). In Use → Modify → Create, students are first introduced to new computing concepts by using example code. Next, they modify the example code and observe the results of their changes. Lastly, they create their own code from a blank slate based on the new concepts they learned.

We drew from previewing and text structure strategies for reading comprehension in our design of TIPP&SEE. Previewing helps students set goals for reading and activates prior knowledge (Klingner & Vaughn, 1998; Manz, 2002). When reading example code containing a new concept, students might scan the code to quickly identify familiar and unfamiliar concepts. They could think about their prior knowledge of the concepts, predict how the new concept might work, and inspect the syntax of the new concept. On the other hand, text structure prepares students to recognize disciplinary-specific text structures and use this knowledge to plan for reading and guide comprehension (Gersten, 2001; Williams, 2005). In computer science, programming languages and environments have specific structures that students must be able to discover to comprehend code and must be able to differentiate as they learn new languages and environments.

Inspired by previewing strategies, the first half, TIPP, guides students in previewing different aspects of a new Scratch project before looking at any code. As a last step, they run the code with very deliberate observations of the events and actions that occur. The second half, SEE, draws from text structure strategies. SEE provides a roadmap for finding code in the Scratch interface (clicking on the sprite and finding the event) and proceduralizes the process by which they can learn how code works by methodical exploration or deliberate tinkering.

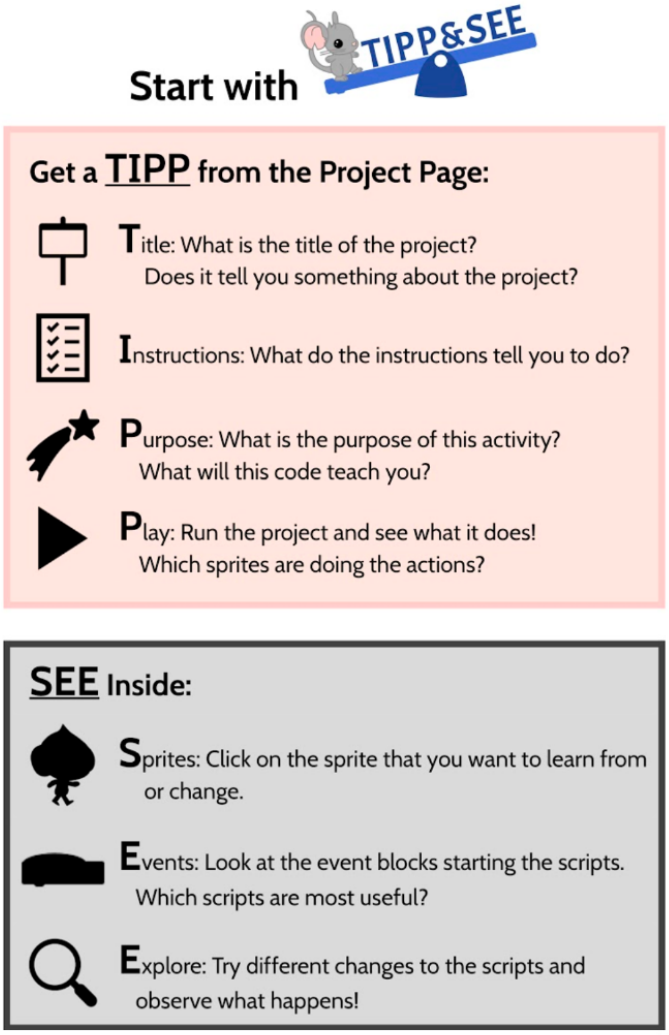

Figure 2. TIPP&SEE learning strategy.

Instructional context

We experimented with TIPP&SEE in the Austin Independent School District (AISD) in Austin, TX, USA. Fifteen teachers underwent the same professional development to teach the Scratch Act 1 curriculum to fourth grade students (ages 9–10). Within a semester (approximately 5 months), students completed Scratch Act 1 (n.d.), an introductory computing curriculum modified from the Creative Computing curriculum consisting of three modules. Each module began with Use/Modify project(s) and culminated in a Create project. Students took two pen-and-paper assessments, one each after Module 2 (events & sequence) and Module 3 (loops). Assessment question tasks are summarized below.

Events & sequence

- Identify the event that triggered one action block
- Given an image of a Scratch stage of two sprites saying different things, identify the script that ran for each sprite
- Identify a multi-block script triggered by the *when the sprite clicked* event
- Identify the last block in a sequence
- Identify the different orders of blocks in two scripts
- Explain in your own words a sequential script of three blocks
- Explain in your own words a sequential script of three blocks triggered by an event

Loops

- Identify how many times a loop in question would repeat
- Identify which code snippet out of four options would cause a sprite to change costumes three times
- Identify the correct unrolled version of a repeat 3 loop with two blocks among options of the blocks in the loop repeated one to four times

- Identify the correct unrolled version of the loop in Q3. Options included a “split loop” option — where the first block was repeated three times followed by the second block repeated three times
- Identify the code that ran before, in, and after a loop
- Identify correct descriptions for one sprite with sequential loops and another sprite with parallel loops
- Explain in your own words a loop in question
- [Extra Credit] Identify how many times the code in the inner loop of a nested loop would run

A total of 16 classrooms participated in the study, including six bilingual English/Spanish classrooms. Teachers were randomly assigned to either the TIPP&SEE or the control condition, resulting in five English-only and three bilingual classrooms in each condition. Treatment classrooms used TIPP&SEE worksheets, whereas control classrooms used worksheets that only had the Use → Modify → Create progression, without stepping students through the TIPP&SEE protocol. After excluding students who switched schools or were chronically absent, there were a total of 96 and 88 students in the control and TIPP&SEE condition, respectively, for a total of 184 students.

Improving primary computing instruction with TIPP&SEE

Our findings showed that students using TIPP&SEE outperformed students who used an unmodified Use → Modify → Create approach on nearly all questions of moderate and hard difficulty (Salac et al., 2020a). TIPP&SEE students outperformed the control students in all but the most basic questions on the events and sequence assessment (Figure 3; asterisks denote significance). Most students were able to demonstrate a simple understanding of events and sequence with just the scaffolding provided by Use → Modify → Create, but with TIPP&SEE, they could demonstrate a more sophisticated understanding.

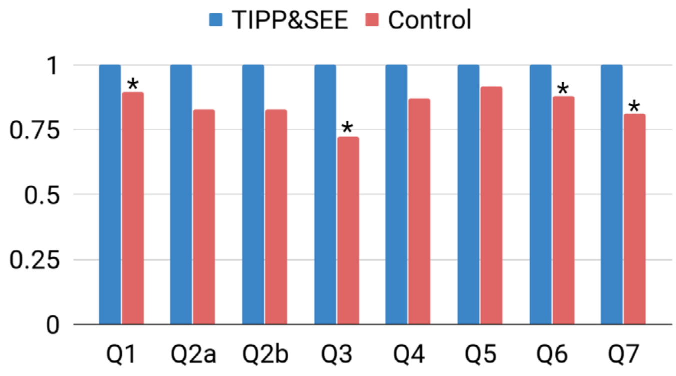

Figure 3. Events and sequence assessment results in normalised mean scores. Asterisks denote statistically significant differences.

As for questions on loops, the TIPP&SEE students performed better than the control students in almost all questions; only parallelism and nested loops (which was not explicitly covered in the curriculum) were beyond their grasp (Figure 4). This suggests that while students are able to make significant learning gains with TIPP&SEE, there is still room for improvement in the instruction of parallelism.

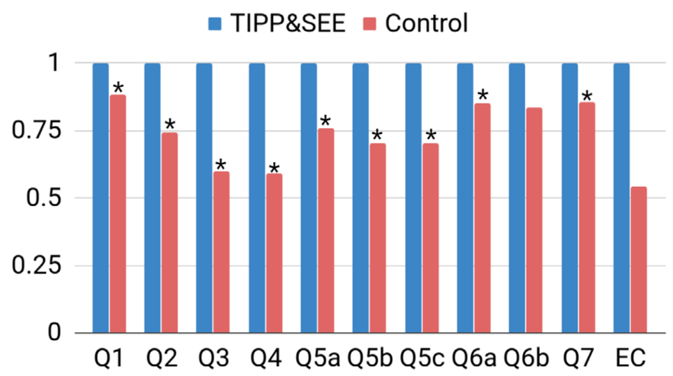

Figure 4. Loops assessment results in normalised mean scores. Asterisks denote statistically significant differences.

Supporting learners of marginalized backgrounds with TIPP&SEE

A learning strategy like TIPP&SEE provides some much-needed scaffolding for students, improving their learning of introductory computing concepts. However, we wanted to investigate if TIPP&SEE worked for all students, especially those who experience academic challenges, such as students in poverty, multilingual students, students with disabilities, and students who had below-grade-level proficiency in reading and math (Salac et al., 2021).

After finding that students using TIPP&SEE exhibited better learning outcomes than students learning only with Use ▸ Modify ▸ Create in our experiment, we decided to focus on students who faced such academic challenges. Students were identified

	TIPP&SEE	**Use ▸ Modify ▸ Create Only**
Economically disadvantaged	70	91
Special education/ Disability	16	15
Limited English proficiency	25	52
Below grade level in reading	54	46
Below grade level in math	55	59

Table 1. Students facing academic challenges in each condition.

as economically disadvantaged if they received a free or reduced-price lunch at school. (The US government has lunch programs in schools for students in poverty.) Students who have limited English proficiency, a disability, or were below grade level in reading and math proficiency were identified through state testing and district-provided demographic data. Some students fulfilled more than one of these characteristics. The distribution of students in each condition is shown in Table 1.

Across all five categories, students using TIPP&SEE performed better than students in the Use ▸ Modify ▸ Create only group for both the Events & Sequence and Loops assessments. Students facing any academic challenge, except for limited English proficiency, still statistically significantly underperformed compared to students without any challenges in both assessments. However, the gap between students with and without any academic challenge was smaller in the TIPP&SEE condition compared with the Use ▸ Modify ▸ Create condition.

Most notably, there were no statistically significant performance differences between Use ▸ Modify ▸ Create students without any academic challenges and TIPP&SEE students with economic disadvantages, disabilities, and proficiencies below grade level in math and reading (Figures 5–8). This suggests that TIPP&SEE scaffolds computing learning for marginalized learners such that they achieve similarly to their peers who do not face academic challenges.

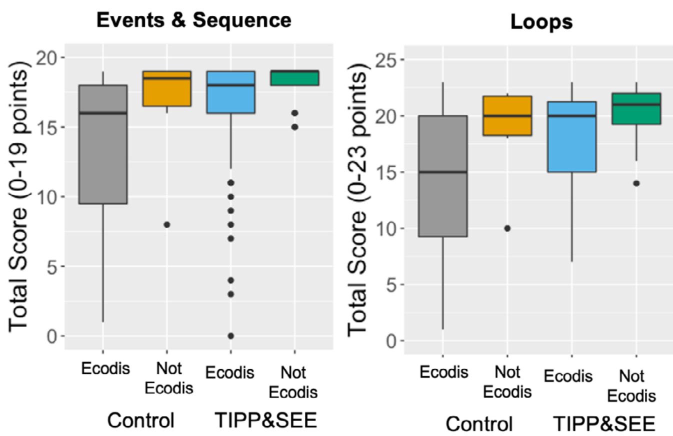

Figure 5. Scores of economically disadvantaged students (abbreviated as "Ecodis") in contrast to the control group ("Not Ecodis") on both assessments.

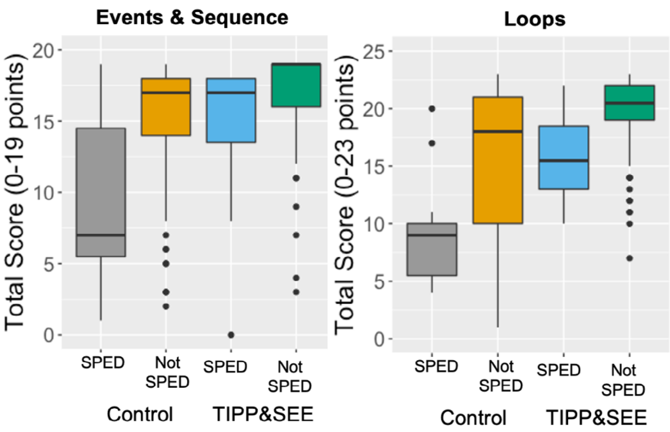

Figure 6. Scores of students with special education needs or disabilities (abbreviated as "SPED") in contrast to the control group ("Not SPED") on both assessments.

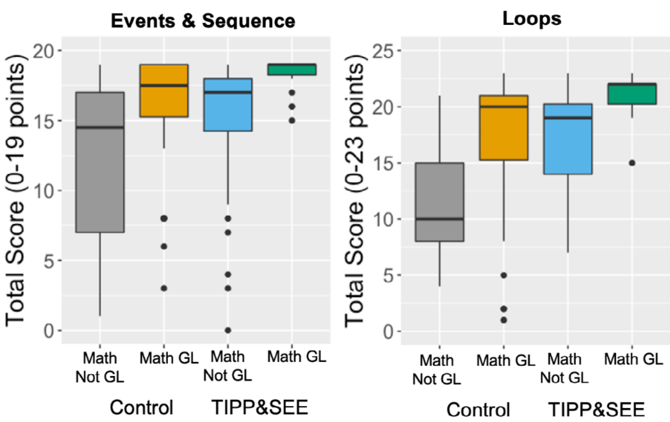

Figure 7. Scores of students with below grade level math proficiency (abbreviated as "Math Not GL") in contrast to the control group ("Math GL") on both assessments.

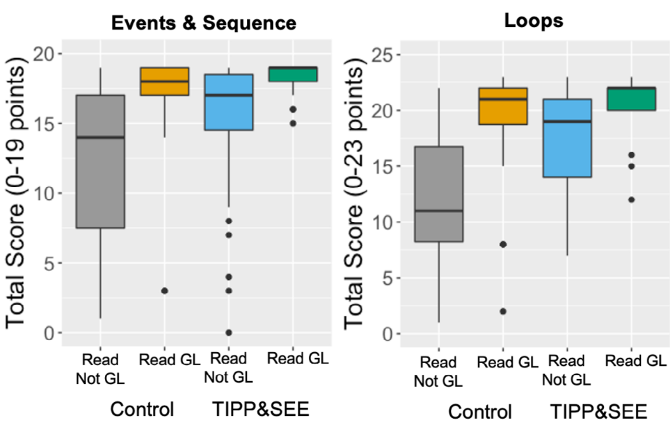

Figure 8. Scores of students with below grade level reading proficiency (abbreviated as "Read Not GL") in contrast to the control group ("Read GL") on both assessments.

The only exception to these trends was limited English proficiency, which did not have a statistically significant association in either assessment (Figure 9). This may be due to bilingual instruction in both conditions. Not only were multilingual students taught in Spanish and English, they also had access to Spanish computer science materials and could even translate Scratch into Spanish.

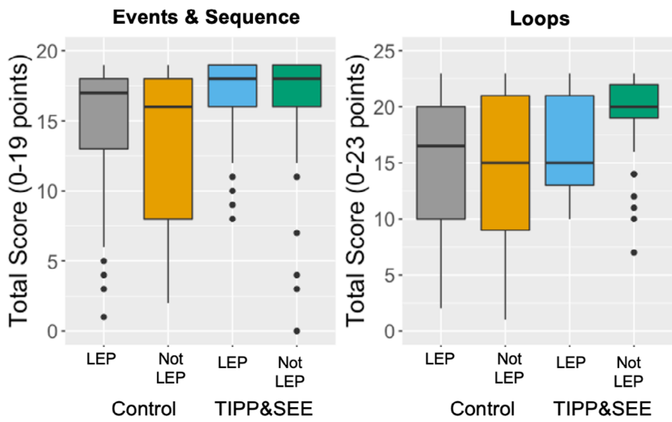

Figure 9. Scores of students with limited English proficiency (abbreviated as "LEP") in contrast to the control group ("Not LEP") on both assessments.

Overall, our findings provide evidence that supports the use of meta-cognitive strategy instruction in computing for marginalized learners who face academic challenges. In this analysis, computing instruction using the TIPP&SEE strategy to scaffold the Use → Modify → Create framework within a Scratch curriculum for fourth grade students (ages 9–10) effectively leveled the playing field. This is supported by findings from math and science education, where open inquiry was less effective than scaffolded inquiry for students with disabilities (Krawec et al., 2019; McGrath et al., 2018; Rizzo et al., 2016). TIPP&SEE enabled students in poverty, students with disabilities, and students who were performing below proficiency on testing in reading and math to perform similarly to their typically achieving peers on computing tasks.

Exploring student behavior using TIPP&SEE

To understand why such dramatic learning differences may have occurred with TIPP&SEE, we first analyzed students' Scratch projects and TIPP&SEE worksheets independently, followed by an investigation of the relationships between the projects, worksheets, and assessments (Franklin et al., 2020).

Scratch projects

Student Scratch projects were statistically analyzed to extract the completion of requirements in all the projects. These requirements were listed on their project planning worksheets. Some requirements were designed to help students demonstrate the computing concept, while others were designed to encourage creativity (Table 2).

Project	Requirements
Name Poem	Modify at least half the sprites Modify backdrop Average script length of at least two
Ladybug Scramble	Ladybug eats at least one aphid Use *Eat Aphid* block Use *Move Steps* block Use *Turn* block
5 Block Challenge	Only use the five required blocks Add new backdrop Add at least two sprites
Ofrenda	Modify *Say* block for at least one sprite Modify at least one sprite's costume Add interactivity for at least one sprite
Parallel Path	At least one sprite has parallel actions on click Two sprites have actions on "9" key press
About Me	At least one sprite At least one interactive sprite
Build-a-Band	Add a script for guitar At least one new sprite At least one new sprite with a script Cat sprite is animated
Interactive Story	Interactive backdrop At least one sprite with a script At least one event block At least one loop block

Table 2. Scratch Act 1 project requirements.

We found that TIPP&SEE students were more likely to complete all the project requirements for 5 Block Challenge, Ofrenda, Parallel Path, and About Me. Figure 10 depicts overall requirement completion rate across the entire curriculum. For each project, the left (blue) bar shows control, and the right (red) bar shows the TIPP&SEE results. This finding suggests that TIPP&SEE students were more capable of applying their new knowledge, and that they benefited from the scaffolding encouraged by the curriculum design.

TIPP&SEE worksheets

Students in the TIPP&SEE condition worked on worksheets prior to starting the Use Modify projects. Questions were divided between the three types of questions: Observe (Figure 11), Predict (Figure 12), and Explore (Figure 13). The Observe questions were first, asking students to record their observations from running the provided project. All worksheets had Observe questions. The other two question categories were only on a subset of worksheets. Predict questions asked students to look at the code and predict what blocks caused which actions they observed. Explore questions had two parts. First, we asked students to make a change to the code and run it, and next, record what happened in response. Answers were transcribed electronically and analyzed for completion and accuracy. We categorized student answers into four categories: Correct, Incorrect, Blank, and No Sheet. The distinction between Blank and No Sheet is that a Blank answer was collected but was not answered by the student, whereas No Sheet indicates that we are missing the entire worksheet for that student.

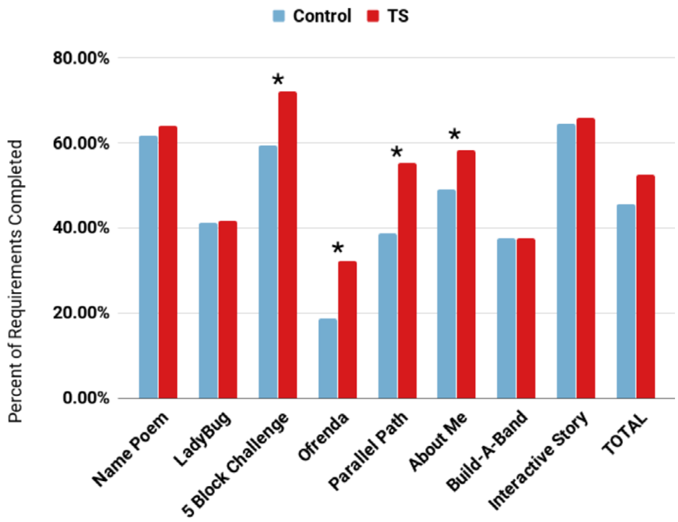

Figure 10. Requirement completion rates across conditions.

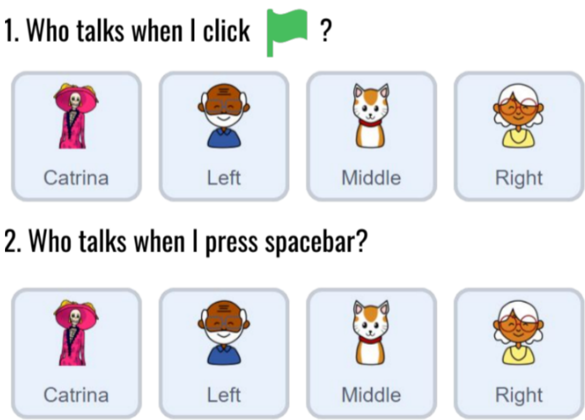

Figure 11. Example Observe questions.

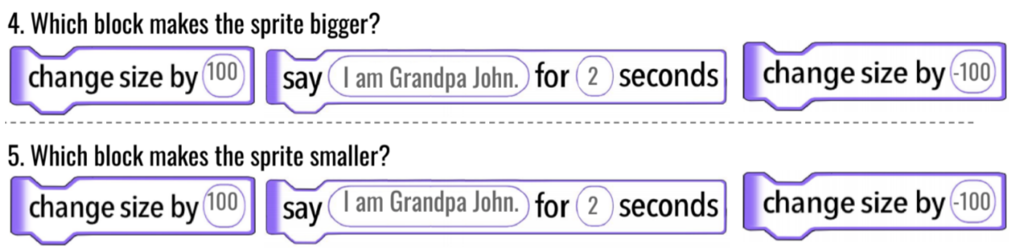

Figure 12. Example Predict questions.

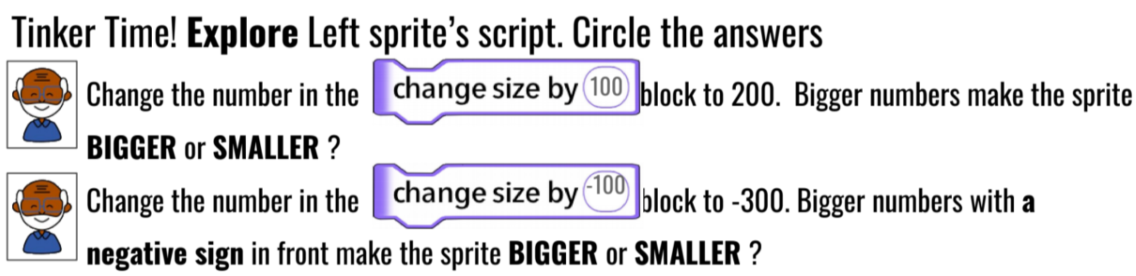

Figure 13. Example Explore questions.

We found that a majority of students completed and correctly answered Observe and Predict questions, while Explore questions were largely left blank. Figure 14 shows the percentage of students that completed and correctly answered questions across all TIPP&SEE worksheets, sorted by the type of question. It was unclear if the reason for skipping Explore questions was because students did not follow the Explore prompt or because they did not record their observations. There were several reasons, however, that students could have skipped them. First, because explore questions were only included in a few projects, following and recording explore prompts may not have become routine. On a related note, students may have needed more scaffolding with this type of questions, requiring the teacher to model and practice them. In addition, making code changes is a more difficult task than merely answering a question about what one observes or is thinking, so this may have been cognitively difficult for some students.

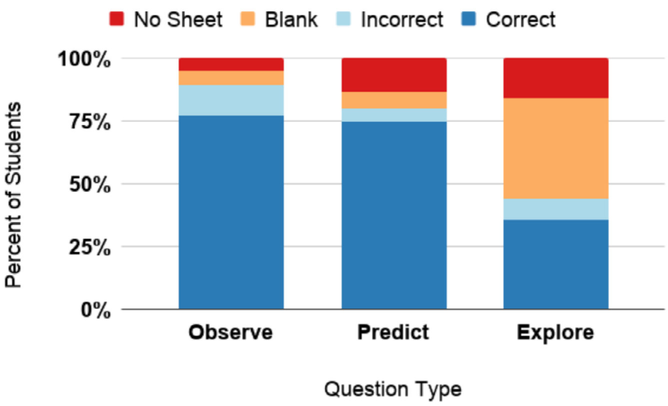

Figure 14. TIPP&SEE worksheet responses across question types.

Relationships between Scratch projects, worksheets, and assessments

We found very few statistical correlations between any of the behavioral measures: individual requirement completed, percentage of requirements completed, worksheet questions completed, and individual written assessment question performance.

The lack of correlations between project attributes and assessments is not entirely surprising. On the Matrix Taxonomy for Computer Science Learning (Fuller et al., 2007), project attributes reflect the 'Producing' dimension (designing and building new code), while assessments reflect the 'Interpreting' dimension (understanding existing code). It is possible for both dimensions to develop independently. Further, Brennan et al. (2012) have shown that students frequently use code that they do not fully understand. Another prior study also revealed that student artefacts can have false positives, where students use code that they do not understand, and false negatives, where students understand a concept but do not use related code constructs (Salac et al., 2020b). Students may have run out of time to include these code constructs or simply did not see the need for those constructs in their projects.

In contrast, the fact that the worksheet behaviors (both completeness and correctness) were hardly correlated with the assessments was more unexpected, as both reflect the same 'Interpreting' dimension of the Matrix Taxonomy. Previous studies have found relationships between formative activities or assignments and learning in Scratch (Grover et al., 2015; Su et al., 2015). These activities and assignments varied widely in structure. Even within our curriculum, the TIPP&SEE worksheets differed in structure as well.

Taken together, while students in the TIPP&SEE group performed actions we believe lead to more success, no individual actions directly explain the results. Like other meta-cognitive strategies, the value of TIPP&SEE may lie in cognitive processes not directly observable, and may vary based upon individual student differences.

Slow-reveal algorithms: Scaffolding critical reflection of technology's ethical impacts

Driven by the groundbreaking work by Benjamin (2020), Noble (2018), O'Neil (2017), and many others on the role technology plays in amplifying societal biases, I decided to grow my research agenda to not only fostering children's technical computing competencies (e.g. programming), but also their ethical computing competencies. In this section, I describe a scaffold we developed to support children's sensemaking around algorithmic bias, and our results from an investigation of this technique (Salac et al., 2023).

Designing slow-reveal algorithms

Prior research has investigated children's perceptions of algorithmic bias, but provides little guidance on engaging children in conversations on algorithmic bias that center their agency and well-being. To address this, we developed discussions and design activities based on three scenarios of algorithmic (un)fairness.

In our design, we drew from sensemaking practices in math and data science education, in particular *slow reveal graphs* (Laib, 2022). Slow reveal graphs are instructional routines that use scaffolded visuals and discourse to make sense of data. These routines start with a graph with minimal information, prompting students to develop hypotheses about the graph. At each incremental reveal of more information on the graph, students are scaffolded in making sense of the information and refining their interpretations. In our discussions, we adapt this routine for different layers of algorithmic bias, encouraging students to form their own interpretations at each revealed layer.

We created three scenarios of algorithmic decision-making that surfaced potential fairness issues to seed sensemaking discussions. These scenarios were selected because they do not have straightforward conceptions of fairness and, thus, may elicit interesting insights from participants.

1. The Search Engine scenario was based on biases in representation from search results (Noble, 2018; Strickland, 2019).
2. The Smart Speaker scenario was based on the failure of many voice recognition systems to recognize other languages or accents (Lawrence, 2013).
3. The School scenario was adapted from the scenario used in Elenbaas & Killen (2017) to understand youth's perceptions of social resource inequality to reflect algorithmic redlining (Safransky, 2020).

Table 3 shows the three sensemaking discussions that students engaged in. Each discussion centered on a specific scenario designed to highlight different aspects of algorithmic unfairness. Each scenario started with seed text describing the situation. This was followed by the incremental reveal of different layers of algorithmic decision-making, similar to slow reveal graphs — whether a computer was used in decision-making, what algorithm was used, what data was used, and what the composition of the team behind the algorithm was.

Scenario	Seed text	Sit.	Comp.	Algo.	Data	Team
Search Engine	Ahmad is making a presentation for what he wants to major in college: nursing. When he searches online for images of nurses, he can barely find images of male nurses. Almost all the images are of women.	✓		✓		✓
Smart Speaker	Alex and her friends are playing with her family's new smart speaker, Blurty. She notices Blurty responds to all her friends except Maximo, who just moved to the US from Mexico.	✓		✓	✓	✓
School	There are two schools, School A and School B, in the same city. There are the same number of kids who go to both schools. Here are some of the kids who go to School A (show a group of White children) and here are some of the kids who go to School B (show a group of Black children). In School A, every classroom has six boxes of school supplies, such as books, calculators, art supplies, and notebooks, to use when kids are learning. In School B, every classroom has one box of school supplies.	✓	✓	✓	✓	✓

Table 3. Seed text and layers discussed in each scenario.

We presented scenarios in this order to highlight an increasing scope of harm. In the Search Scenario, only a single individual is harmed. In the Speaker scenario, while only a single individual is harmed, the harm results in group exclusion. In the School scenario, a community is harmed. We also designed the scenarios to have varying technical focuses, with the Search scenario involving only software components, the Speaker scenario including hardware and software components, and the School scenario involving a covert, non-obvious technical component.

Table 4 gives a detailed overview of the School scenario. The Search and Speaker scenarios followed a similar structure, with some key differences. First, both scenarios had an apparent technical component that did not require uncovering. Second, they had different high-level abstractions of the algorithm. The Search Engine followed a naive search algorithm accounting for keyword presence in images' metadata and the Smart Speaker being activated by a specific phrase. Third, the Search scenario had no training data as it was not a machine learning-based algorithm, while the Speaker scenario had training data of voices from English-speaking countries. Lastly, the various teams in the Search scenario differed based on gender, while the Speaker scenario differed based on country of origin.

Stage	School: Question and revealed layers
Warm-up	*Ask*: Where do you go to school? When you walk into your school, what do you see? When you walk into your classroom, what do you see?
Reflection	
Situation	*Reveal seed text (Table 3)* *Ask:* Why do you think School A has more supplies than School B?
Computer	*Reveal:* A computer decided how many supplies each school should get. *Ask:* **a.** What do you think of a computer making that decision? **b.** Why do you think a computer decided to give School A more supplies than School B?
Algorithm	*Reveal:* School A is in neighborhood A and School B is in neighborhood B. The computer made its decision using this rule: "For every $100 the neighborhood gives to the school, every classroom gets an extra box of school supplies." *Ask:* **a.** What do you think of the rules the computer used? [If participants don't mention fairness] How fair do you think the rules are? Why? **b.** How do the rules impact different people? **c.** What are the pros and cons of using a computer to make that decision?
Data	*Reveal:* The computer used data about how much neighborhoods gave in the past to decide that each neighborhood should give $100 for each box of school supplies. *Ask:* **a.** What do you think of the data that the computer used? **b.** How fair is it that the computer used past data? Why?

▼ Table continued below

Stage	School: Question and revealed layers
Team	*Reveal:* The team who designed the rules and data the computer used was made up of all White people. *Ask:* [If participants do not mention fairness for questions a, b, and c.] How fair do you think this team is? Why? **a.** What do you think of this team? **b.** What if the team was made up of all Black people? What do you think of this team? **c.** What if the team was made of people from different races? What do you think of this team? **d.** Which team is the most fair? Why? [If participants bring up factors] If you don't think any of the teams are the most fair, what would be the most fair team? Why?
Design activity	*Ask:* **a.** Imagine you're the boss and you're in charge of the rules. What rules would you use to decide how much supplies each school should get? **b.** Who will be applying the rules? Will it be a computer? A person? A team? Both? **c.** How do you make sure the rules are fair? [Follow-up questions if needed:] **a.** What kind of team would be the most fair in designing these rules? **b.** How would you and your team design the rules fairly? **c.** How would you and your team test the rules fairly?

Table 4. School sensemaking discussion questions in full. Italics denote actions performed by facilitators.

Scaffolding children's sensemaking around algorithmic bias

We conducted these discussions and activities with 16 children (ages 8–12) in the US, and examined our data using qualitative thematic analysis. Through our analysis, we identified many different salient factors in our participants' sensemaking of algorithmic fairness. To relay our results, we organize the following section using the metaphor of a camera, specifically lenses and filters (Figure 15).

In the metaphor that emerged from the data, we focus on the lenses and filters of a camera. Lenses allow photographers to change the scale and resolution of a shot, while filters allow photographers to change the kinds of light in a shot. Photographers can attach different filters to a lens to capture the same view but with different lights, making the final image appear different. In this metaphor, our participants are photographers, capturing algorithmic (un)fairness in different scales and lights to make sense of them. Each participant has their own camera and their own set of lenses and filters, which can grow over time. We identified two different lenses participants used to make sense of (un)fairness at different scales and resolutions: (1) a human lens, which ranged from individual to societal factors, and (2) a technical lens, which ranged from individual technology creators to broader technical factors. In addition to adjusting the scale and resolution with their chosen lens, we found that participants used different characteristics, such as gender and class, as filters to change what was most salient to their sensemaking for each scenario. The lenses and filters used by children in our study are summarized in Table 5.

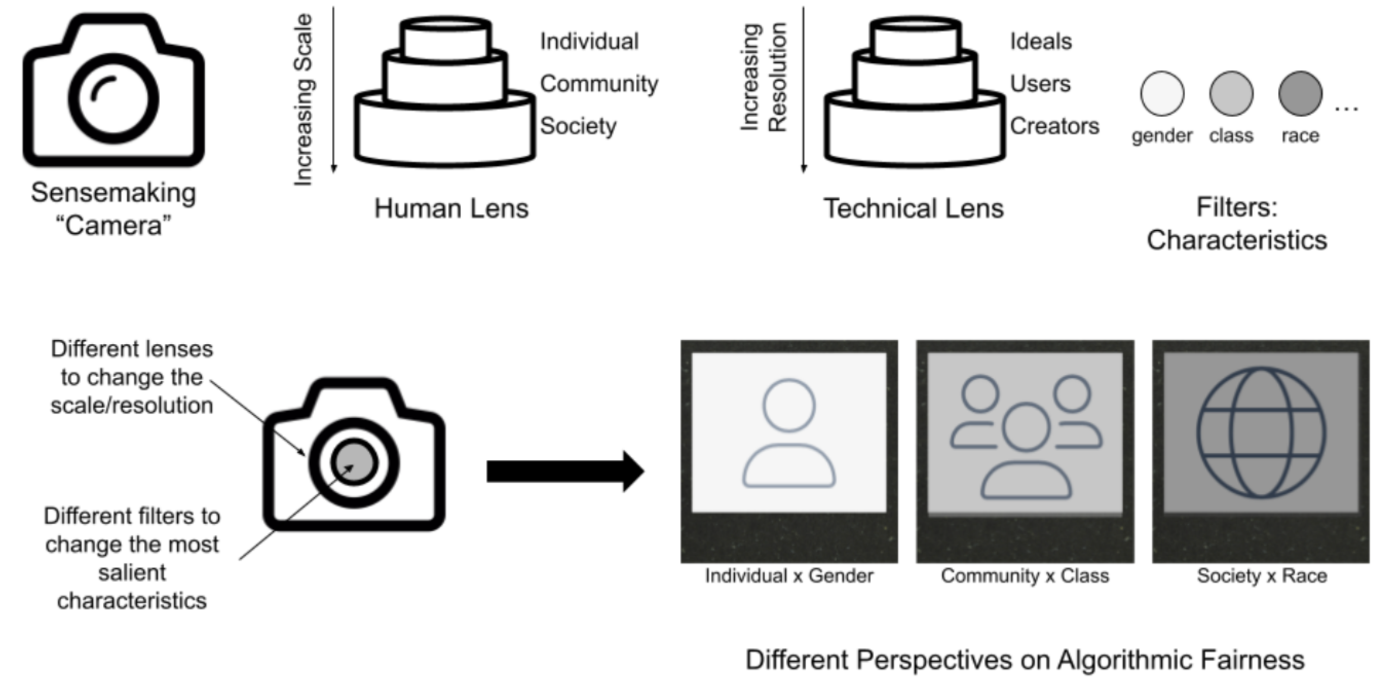

Figure 15. Camera metaphor as an organizational aid for children's sensemaking around algorithmic fairness.

Camera metaphor	Description
Human Lens	***Different scales of human factors***
Individual	Factors connected to an individual, e.g. children's own lived experiences
Community	Factors linked to a collective group of people, e.g. interpersonal relationships
Society	Factors attributed to larger structural issues, e.g. systemic marginalization
Technical Lens	***Different resolutions of technology factors***
Technology Creators	People involved in developing technology, e.g. engineers, designers
Users	People using technology, both real and hypothetical
Ideals	Attitudes towards technology
Characteristics as Filters	***Characteristics participants used to describe the factors above, e.g. gender, class, race***

Table 5. Description of camera metaphor elements.

In the human lens, participants used factors at increasing group sizes ranging from individual to society, which reflects ecological system theory that views an individual relative to their communities and larger society (Bronfenbrenner, 1992). At the individual level, participants often grounded their sensemaking in their lived experiences and principles. Similarly, at the community level, participants tended to base their reasoning in both real and hypothetical interpersonal relationships, which often arose when hypothesizing the impacts of algorithmic bias. However, while participants also reasoned at the societal level, this sensemaking was more vague — participants tended to attribute structural issues to individual bad actors, not fully comprehending the large, systemic scale. Participants also expressed societal ideals when making sense of the unfairness, but it was not always clear if they believed in the ideals themselves. This vagueness may be because these issues are more abstract, coming from surrounding adult society, and less grounded in their own lived experiences (Kohlberg, 1975).

As for the technical lens, participants often developed specific conceptualizations of users, drawing from both real people in their lives and hypothetical users. This mirrors the relevance of lived experiences and interpersonal relationships observed through the human lens. In contrast with prior work (Long & Magerko, 2020), participants also exhibited a distrust towards technology, doubting computers' abilities, and displayed an inclination towards a human approach to address unfairness. This may be due to various reasons, including but not limited to (1) the lack of personification of the technology in the scenarios (Festerling et al., 2022), (2) the child characters in the scenarios were easier to empathize with, (3) lived experiences with or exposure to adult tech use (Plowman et al., 2010), and (4) a broader attitude change towards technology in society. Given the salience of our participants' lived experiences regardless of the lens, we encourage designers, educators, and other stakeholders to consider centering children's lived experiences, and their resultant funds of knowledge, in discussions of algorithmic fairness.

With respect to the filters, participants seemed to be especially attuned to gender, race/ethnicity, country of origin, and age, as they used them to make sense of the unfairness in all scenarios regardless of whether they were prompted. Children develop identities around gender and race from a young age as part of learning social competence (Katz et al., 1997), which may explain the salience of gender and race. While we did not specifically ask in our demographics form, some participants brought up their immigrant backgrounds, which may account for the relevance of country of origin across the scenarios. Age may be particularly salient to participants because many developmental milestones in childhood are tied to age (Brain Architecture, 2019). Given the salience of our participants' lived experiences regardless of the lens, and with this understanding of how learners' funds of knowledge may change based on ages, stages of development, identities, and backgrounds, we encourage designers, educators, and other stakeholders to consider scaffolding discussions of algorithmic fairness accordingly, thus allowing learners to leverage their funds of knowledge for deeper reasoning.

Closing remarks and future directions

Most prior research in computing education has emphasized teaching the technical skills and competencies, and building students' senses of confidence and belonging in the field. As a result, much work has focused on fostering equitable access and learning outcomes for all students, including my own.

However, technology's role in amplifying societal oppression have only become clearer in recent years, with children increasingly at risk of being impacted by algorithmic bias. Just as countless efforts in computing education supported the development of technical computing competencies, I posit that the ability to critically examine technology's role in society — ethical competency

— should be considered a computing competency. Ethical competencies deserve the same, if not more, effort towards studying and developing new ways to foster it in children. Thus, future directions for my research include advancing capacity, access, participation, and engagement in fostering children's ethical computing competencies.

References

Balch, C., Chung, M., & Brennan, K. (n.d.). *Creative Computing Curriculum*. Retrieved April 25, 2023, from http://scratched.gse.harvard.edu/guide/

Benjamin, R. (2019). *Race after Technology: Abolitionist Tools for the New Jim Code*. Polity.

Center on the Developing Child at Harvard University. (2019, August 20). *Brain Architecture.* Center on the Developing Child at Harvard University. https://developingchild.harvard.edu/science/key-concepts/brain-architecture/

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada* (Vol. 1, p. 25). http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Bronfenbrenner, U. (1992). *Ecological Systems Theory*. Jessica Kingsley Publishers.

Elenbaas, L., & Killen, M. (2017). Children's perceptions of social resource inequality. *Journal of Applied Developmental Psychology*, *48*, 49–58. https://doi.org/10.1016/j.appdev.2016.11.006

Festerling, J., & Siraj, I. (2022). Anthropomorphizing technology: a conceptual review of anthropomorphism research and how it relates to children's engagements with digital voice assistants. *Integrative Psychological and Behavioral Science*, *56*(3), 709–738. https://doi.org/10.1007/s12124-021-09668-y

Franklin, D., Salac, J., Crenshaw, Z., Turimella, S., Klain, Z., Anaya, M., & Thomas, C. (2020, August). Exploring student behavior using the TIPP&SEE learning strategy. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (pp. 91–101). https://doi.org/10.1145/3372782.3406257

Fuller, U., Johnson, C. G., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J., Lahtinen, E., Lewis, T. L., McGee Thompson, D., Riedesel, C., & Thompson, E. (2007). Developing a computer science-specific learning taxonomy. *ACM SIGCSE Bulletin*, *39*(4), 152–170. https://doi.org/10.1145/1345375.1345438

Gersten, R., Fuchs, L. S., Williams, J. P., & Baker, S. (2001). Teaching reading comprehension strategies to students with learning disabilities: A review of research. *Review of Educational Research, 71*(2), 279–320. https://doi.org/10.3102/00346543071002279

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education, 25*(2), 199–237. https://doi.org/10.1080/08993408.2015.1033142

Katz, P. A., & Kofkin, J. A. (1997). Race, gender, and young children. *Developmental psychopathology: Perspectives on Adjustment, Risk, and Disorder, 21*, 51–74.

Klingner, J. K., & Vaughn, S. (1998). Using collaborative strategic reading. *Teaching Exceptional Children, 30*(6), 32–37. https://doi.org/10.1177/004005999803000607

Kohlberg, L. (1975). The cognitive-developmental approach to moral education. *The Phi Delta Kappan, 56*(10), 670–677. https://www.jstor.org/stable/20298084

Krawec, J., & Steinberg, M. (2019). Inquiry-Based Instruction in Mathematics for Students with Learning Disabilities: A Review of the Literature. *Learning Disabilities: A Multidisciplinary Journal*, *24*(2), 27–35. https://doi.org/10.18666/LDMJ-2019-V24-I2-9866

Laib, J. (2022, November 27). *Slow Reveal Graphs*. Slow Reveal Graphs. https://slowrevealgraphs.com/

Lawrence, H. M. (2013). *Speech Intelligibility and Accents in Speech-mediated Interfaces: Results and Recommendations*. Illinois Institute of Technology.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

Long, D., & Magerko, B. (2020, April). What is AI literacy? Competencies and design considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1–16). https://doi.org/10.1145/3313831.3376727

Manz, S. L. (2002). A strategy for previewing textbooks: teaching readers to become THIEVES. (Teaching Ideas). *The Reading Teacher, 55*(5), 434–436. https://doi.org/10.36312/jupe.v4i4.684

McGrath, A. L., & Hughes, M. T. (2018). Students with learning disabilities in inquiry-based science classrooms: A cross-case analysis. *Learning Disability Quarterly, 41*(3), 131–143. https://doi.org/10.1177/0731948717736007

Noble, S. U. (2018). *Algorithms of Oppression: How Search Engines Reinforce Racism*. New York University Press.

O'Neil, C. (2017). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown.

Plowman, L., McPake, J., & Stephen, C. (2010). The technologisation of childhood? Young children and technology in the home. *Children & Society, 24*(1), 63–74. https://doi.org/10.1111/j.1099-0860.2008.00180.x

Rizzo, K. L., & Taylor, J. C. (2016). Effects of inquiry-based instruction on science achievement for students with disabilities: An analysis of the literature. *Journal of Science Education for Students with Disabilities, 19*(1), 2. https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=1053&context=jsesd

Safransky, S. (2020). Geographies of algorithmic violence: Redlining the smart city. *International Journal of Urban and Regional Research, 44*(2), 200–218. https://doi.org/10.1111/1468-2427.12833

Salac, J., White, M., Wang, A., & Franklin, D. (2019, February). An analysis through an equity lens of the implementation of computer science in K–8 classrooms in a large, urban school district. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1150–1156). https://doi.org/10.1145/3287324.3287353

Salac, J., Thomas, C., Butler, C., Sanchez, A., & Franklin, D. (2020, February). TIPP&SEE: a learning strategy to guide students through use-modify Scratch activities. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 79–85). https://doi.org/10.1145/3328778.3366821

Salac, J., & Franklin, D. (2020, June). If they build it, will they understand it? Exploring the relationship between student code and performance. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 473–479). https://doi.org/10.1145/3341525.3387379

Salac, J., Thomas, C., Butler, C., & Franklin, D. (2021, March). Supporting diverse learners in K–8 computational thinking with TIPP&SEE. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 246–252). https://doi.org/10.1145/3408877.3432366

Salac, J., Thomas, C., Butler, C., & Franklin, D. (2021, August). Investigating the role of cognitive abilities in computational thinking for young learners. In *Proceedings of the 17th ACM Conference on International Computing Education Research* (pp. 2–17). https://doi.org/10.1145/3446871.3469746

Salac, J., Landesman, R., Druga, S., & Ko, A. J. (2023, June). Scaffolding children's sensemaking around algorithmic fairness. In *Proceedings of the 2023 ACM Conference on Interaction Design for Children* (IDC '23). https://jeansalac.github.io/docs/IDC2023_Children_Sensemaking_Algo_Fairness.pdf

Scratch Act 1. Everyday Computing for All Curriculum. (n.d.). Retrieved April 25, 2023, from https://impactconectar.wixsite.com/website/act-1

Su, A. Y., Huang, C. S., Yang, S. J., Ding, T. J., & Hsieh, Y. Z. (2015). Effects of annotations and homework on learning achievement: An empirical study of Scratch programming pedagogy. *Journal of Educational Technology & Society, 18*(4), 331–343. https://www.jstor.org/stable/jeductechsoci.18.4.331

Williams, J. P. (2005). Instruction in reading comprehension for primary-grade students: A focus on text structure. *The Journal of Special Education, 39*(1), 6–18. https://doi.org/10.1177/00224669050390010201

Jean Salac
(University of Washington)

Dr Jean Salac is a postdoctoral researcher and Computing Innovations Fellow at the University of Washington's Code & Cognition Lab. Her research interests include computer science education and child–computer interaction, particularly in justice-focused computing for young learners. Her work has won Best Paper at the International Computing Education Research Conference (ICER) and an honourable mention for Best Paper at the conference on Human Factors in Computing Systems (CHI).

Raspberry Pi
Foundation

www.raspberrypi.org

Title here Lut la nosa velessum earchit atemod eatem sum et

Sub Title One

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo. Otaspic iiscid magnam quist, sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel moloreprepta dolorumqui que eturias imagnam hitia dolendam, quidigendis dis quo te pratur, ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem quis volupta vendae. Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Sub title Two

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo.

Otaspic iiscid magnam quist: sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel moloreprepta dolorumqui que eturias imagnam hitia dolendam.

Quidigendis dis quo te pratur: ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem quis volupta vendae. Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Sub title Two

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo. Otaspic iiscid magnam quist, sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel moloreprepta dolorumqui.

Que eturias imagnam hitia dolendam, quidigendis dis quo te pratur, ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem vendae.

Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Authors

Dr Jean Salac
(University of Washington)

Dr Jean Salac is a postdoctoral researcher and Computing Innovations Fellow at the University of Washington's Code & Cognition Lab. Her research interests include computer science education and child–computer interaction, particularly in justice-focused computing for young learners. Her work has won Best Paper at the International Computing Education Research Conference (ICER) and an honourable mention for Best Paper at the conference on Human Factors in Computing Systems (CHI).

Title here Lut la nosa velessum earchit atemod eatem sum et

Name here (University here)
Lorem Ipsum (Educational establishment)

Sub Title One

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo. Otaspic iiscid magnam quist, sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel moloreprepta dolorumqui que eturias imagnam hitia dolendam, quidigendis dis quo te pratur, ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem quis volupta vendae. Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Sub title Two

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo.

Otaspic iiscid magnam quist: sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel moloreprepta dolorumqui que eturias imagnam hitia dolendam.

Quidigendis dis quo te pratur: ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem quis volupta vendae. Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Sub title Two

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo. Otaspic iiscid magnam quist, sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel moloreprepta dolorumqui.

Que eturias imagnam hitia dolendam, quidigendis dis quo te pratur, ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem vendae.

Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Title here Lut la nosa velessum earchit atemod eatem sum et

Name here (University here)
Lorem Ipsum (Educational establishment)

Sub Title One

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo. Otaspic iiscid magnam quist, sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel molореprepta dolorumqui que eturias imagnam hitia dolendam, quidigendis dis quo te pratur, ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem quis volupta vendae. Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Sub title Two

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo.

Otaspic iiscid magnam quist: sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel molorepreрta dolorumqui que eturias imagnam hitia dolendam.

Quidigendis dis quo te pratur: ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem quis volupta vendae. Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Sub title Two

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo. Otaspic iiscid magnam quist, sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel molorepreрta dolorumqui.

Que eturias imagnam hitia dolendam, quidigendis dis quo te pratur, ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem vendae.

Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Sub Title One

Odi repedio volorerae eaquiandere ma plist ute officit inulpa ium as dem. Ut alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo. Otaspic iiscid magnam quist, sint et as que experorumqui doluptas nus minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel moloreprepta dolorumqui que eturias imagnam hitia dolendam, quidigendis dis quo te pratur, ium qui comnis eum amendam, volessumqui ipsamet qui ditium ello corro blam faciendebis dus dolorion pro to tem quis volupta vendae. Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Sub title three for things like bullet points:

- Eaquiandere ma plist ute officit inulpa ium as dem. Alicide vita vel iducil eum ratinctent quam veliqui corro doluptatis abo.
- Liscid magnam quist, sint et as que experorum qui doluptas nus.
- Minia net laborehent min pa si non nis num hil im ullorestis et magnimi ncipidel moloreprepta dolorumqui que.
    - **i.** Ditium ello corro blam faciendebis dus dolorion pro to tem quis volupta vendae.
    - **ii.** Ut atem quaerum et optatquiate net voloria dignihil erum ilitatias vel molupid quas aces doluptatius derumet faciatur, ipitent quae.

Graph/Data/Image

Caption lorem ipsum dolor sit amet

TIPP&SEE: Scaffolding programming learning for equitable outcomes

Designing the TIPP&SEE learning strategy

Motivated by the inequitable performance disparities we observed, we designed TIPP&SEE, a learning strategy that scaffolds student exploration of provided programs for activities on the Use → Modify step of the Use → Modify → Create progression (Lee et al., 2011). In Use → Modify → Create, students are first introduced to new computing concepts by using example code. Next, they modify the example code and observe the results of their changes. Lastly, they create their own code from a blank slate based on the new concepts they learned.

We drew from previewing and text structure strategies for reading comprehension in our design of TIPP&SEE. Previewing helps students set goals for reading and activates prior knowledge (Klingner & Vaughn, 1998; Manz, 2002). When reading example code containing a new concept, students might scan the code to quickly identify familiar and unfamiliar concepts. They could think about their prior knowledge of the concepts, predict how the new concept might work, and inspect the syntax of the new concept. On the other hand, text structure prepares students to recognize disciplinary-specific text structures and use this knowledge to plan for reading and guide comprehension (Gersten, 2001; Williams, 2005). In computer science, programming languages and environments have specific structures that students must be able to discover to comprehend code and must be able to differentiate as they learn new languages and environments.

Inspired by previewing strategies, the first half, TIPP, guides students in previewing different aspects of a new Scratch project before looking at any code. As a last step, they run the code with very deliberate observations of the events and actions that occur. The second half, SEE, draws from text structure strategies. SEE provides a roadmap for finding code in the Scratch interface (clicking on the sprite and finding the event) and proceduralizes the process by which they can learn how code works by methodical exploration or deliberate tinkering.

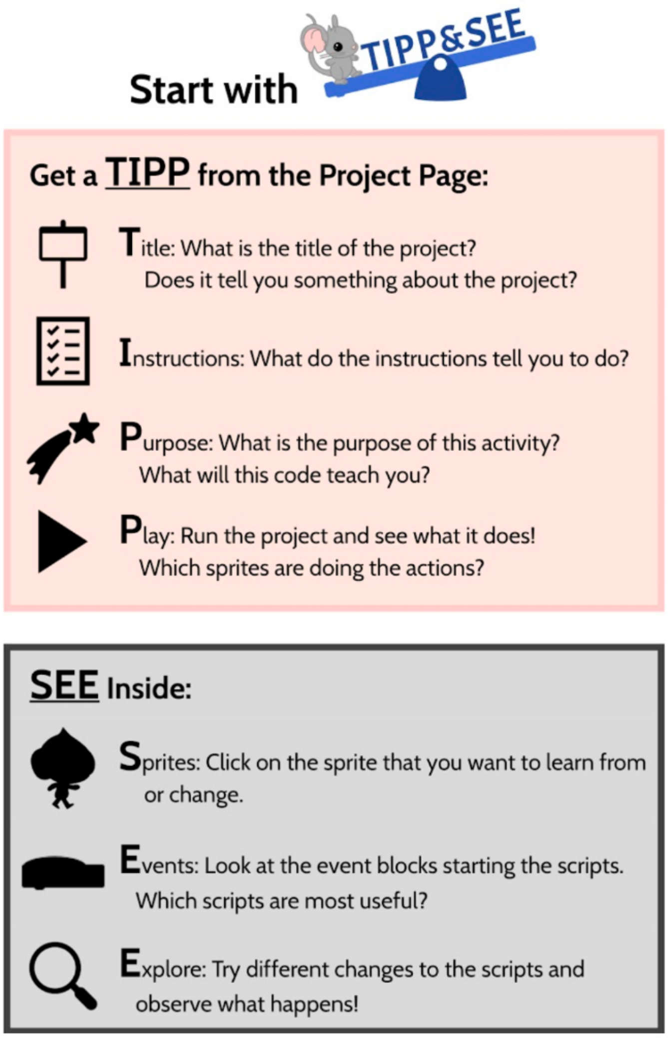

Figure 2: TIPP&SEE learning strategy.

Table style

Title	Title
aaa	aaa
aaa	aaa
aaa	aaa

References

Et eum rerferias vitatur andus, aut dolessit quia prae omnitiumquas soles dis dolorion et et faccusda id moditiis et modiatius elestiur. https://linkhere/loremipsum

Coremoluptur magnati amusandi delesed qui cor rernati con plaut que etus acest, il illam, vel ium endis magnate est eos ipsunt ero dolorpo. https://linkhere/loremipsum

Faccusant exped qui quae sequia porent quo voloribusdae dicabore volorios ex es non. https://linkhere/loremipsum

Ipit aut voluptur, sanda dolorro to to omnis explatiorit minctio voluptate delici derum repta. https://linkhere/loremipsum

Nonsequiae. Nam et explicte voloritaquia que nobis int omniatur aspel illate latiumque non https://linkhere/loremipsum

Voluptati nis eos res dolectassi ipsuntion poresecesti comnihi liatatet, simil ex et harci si ut vellor audae nonsedi pictur, idellorum quidi. https://linkhere/loremipsum

Aboressequi occus inctat delecatet qui blam illecti deles sundaeperes dolut aut quiasit rem eosam, volorrunte etur? https://linkhere/loremipsum

Dolorio bearumquam, consequ iatecae odita con re, te pelia velescid quunt.

Facia dem facimporem sento iliqui ulparib usanditaqui quiatur remped quaerunti utas et

Exerrum volore nim ium qui optamenet dolore nonsed ma sumquiatur, sincips anisitint a

Cullendi corestio. Ta sus enimus nulpa natet endam, cum que magnam iusa alia et incturi buscias perore cum que si apicat ape optatus

Molende dolorpor atis exera con cori dolupta tectem re precabore digniet uriatio. Nem

Seque nestioribus volorro etus et eos quia niaturehent eum aut eatur ratiatium quidesedi sae nim harum et voluptae necepra turit, volendebit lit quiame molupic tectum velictem vendi tecum haria dolorum fuga.

Tem fuga. Nequis nis sit exeriat uriata vellis nes experio.

Ut ullacestin porest recus ex et renimo to evelestiore expligenis niae natatin ihiciis tincidit magnihitatem es se et as aliqui odi odios imperchit harum nem erfere

Volorpos as dis restotas est officab orrovidemo coremposam que lam reration eatquiaspel maxim et utempos nat odit la volupis enis aut laccus iliqui ut mil mossed quiducia ventem quidus, ex eos eiunt.

Lique magnim nos mi, simil il issequi volor aditiunto quiatqu idundae culloritatur simpore eos doluptinis reius endaesciis audis eium aut ide corro dentur? Ut volut dolum quodis eost remporu pitaeculpa ni illissi mustiisti dolorrunt hitaquam apitaquis volorem possequ iatessim facea pe corempore aute expero disim que con elest, cone istibus doluptatio magnita ecaectur, consequ atatum arum re none vent doluptam faceres eatum hic tem qui