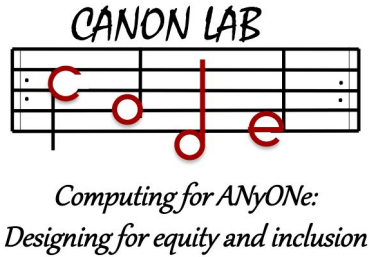
An Analysis through an Equity Lens of Computer Science in K-8 Classrooms

Jean Salac, Max White, Ashley Wang, Diana Franklin



What is the goal of CS for All?

*What does it mean to reach **all** students?*

csforall.org

What does equity look like?

Parental lack of CS vocabulary: Barrier to informal CS learning.

(DiSalvo et al., 2014)

What does equity look like?

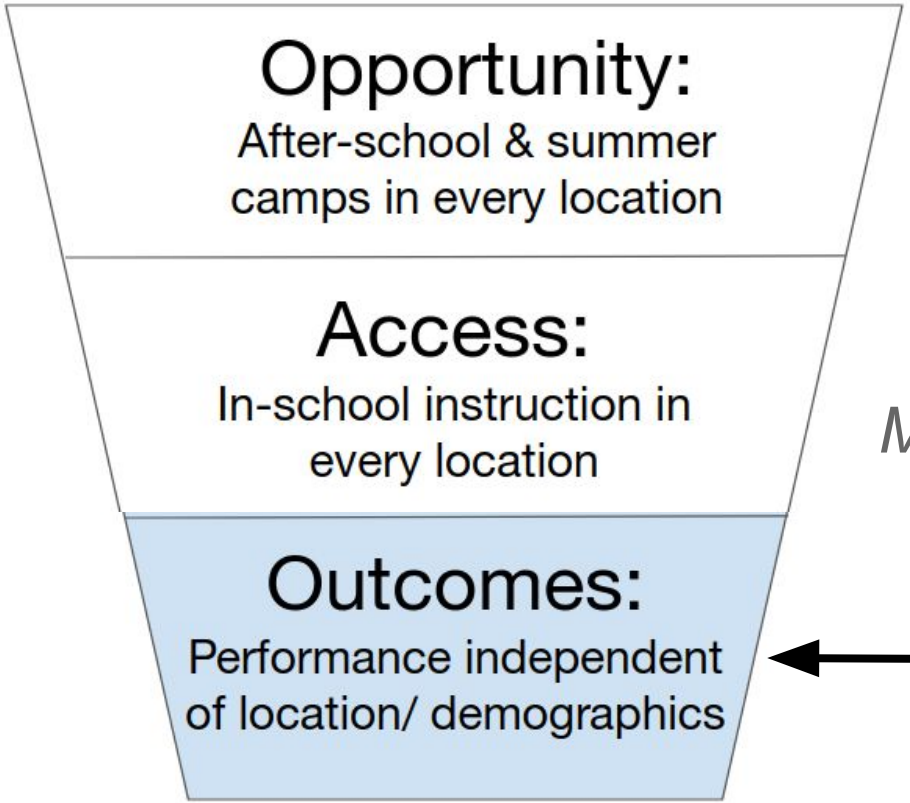
Parental lack of CS vocabulary: Barrier to informal CS learning.
(DiSalvo et al., 2014)

Structural Barriers
(Margolis et al.,2010)

Math & Literacy
(Century et al., 2018)

Demographics
(Fancsali et al., 2018)

What does equity look like?



Parental lack of CS vocabulary: Barrier to informal CS learning.
(DiSalvo et al., 2014)

Structural Barriers
(Margolis et al.,2010)

Demographics
(Fancsali et al., 2018)

Math & Literacy
(Century et al., 2018)

This study!

Why measure outcomes?

- Interventions for not-yet-successful students
- Teachers:
  - Professional Development
  - Teaching Strategies
- Students:
  - Better Curriculum
  - Learning Strategies

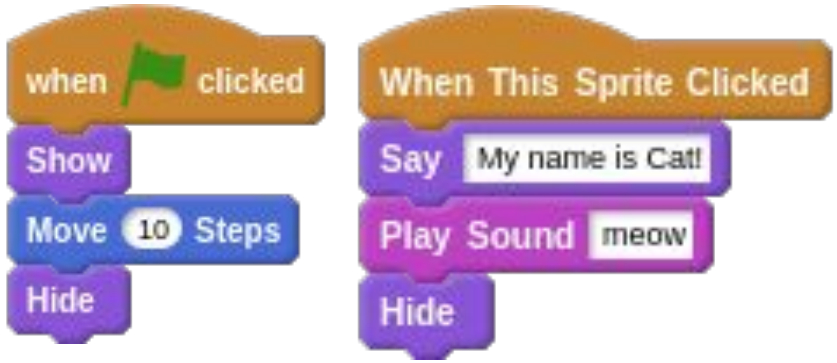
Milwaukee Public Schools

Our Approach

- Nationally, school performance correlated with:
  - Socioeconomic background
  - Underrepresented ethnic minorities
- Do similar systemic disadvantages leak into CS learning?

Research Goals

1. Where students succeed & struggle in:
   - ***sequence & events***
   - ***loops***



2. How school performance affects students' CS learning outcomes

Prior Work - Learning

- Success & challenges w/ block-based languages (Hill et al., 2015)
  - Initialization (Franklin et al., 2016)
  - Variables & Loops (Grover et al., 2017)
- Age-appropriate CS concepts(Flannery et al.,2013, Franklin et al.,2017, Seiter et al.,2013)
- Transition from Scratch to text-based programming languages (Weintrop et al.,2018)

Research Context: Schools

- 3 schools: 1 high-, 1 mid-, & 1 low-performing
- 3 fourth-grade classrooms per school
  - Student age: 9-10 years old

School Demographics

School	Non-URM (%)	URM (%)	Declined to State
High	71	15	14
Mid	20	73	7
Low	8	65	27

Research Context: Teachers

- Received same training
- Taught same curriculum
- Intro Computational Thinking course:
  - Sequence & Events (Assessment 1)
  - Loops (Assessment 2)

Assessment Design

- Multiple questions for each concept
- Design team consisted of K-8 CS Ed:
  - 2 practitioners
  - 1 professor
  - 1 graduate student
- Outside consultants:
  - Professor in reading comprehension strategies
  - 4th grade teachers

Quantitative Analysis

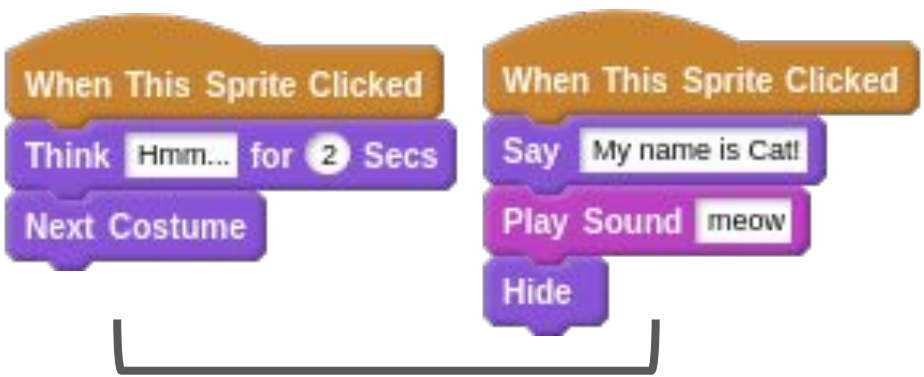
- Completely Randomized Hierarchical Design
  - Classrooms nested within schools
- Linear Model: $Y_{ijk} = \mu + \alpha_j + \beta_{k(j)} + \varepsilon_{i(jk)}$
- Analysis Steps:
  1. ANOVA F-test: Overall school effect
  2. Fisher-Hayter: Pairwise school difference
  3. Effect size

Qualitative Analysis

- Free-response questions were open-coded
- 2 researchers with inter-rater reliability > 80%

Events with 1 Script

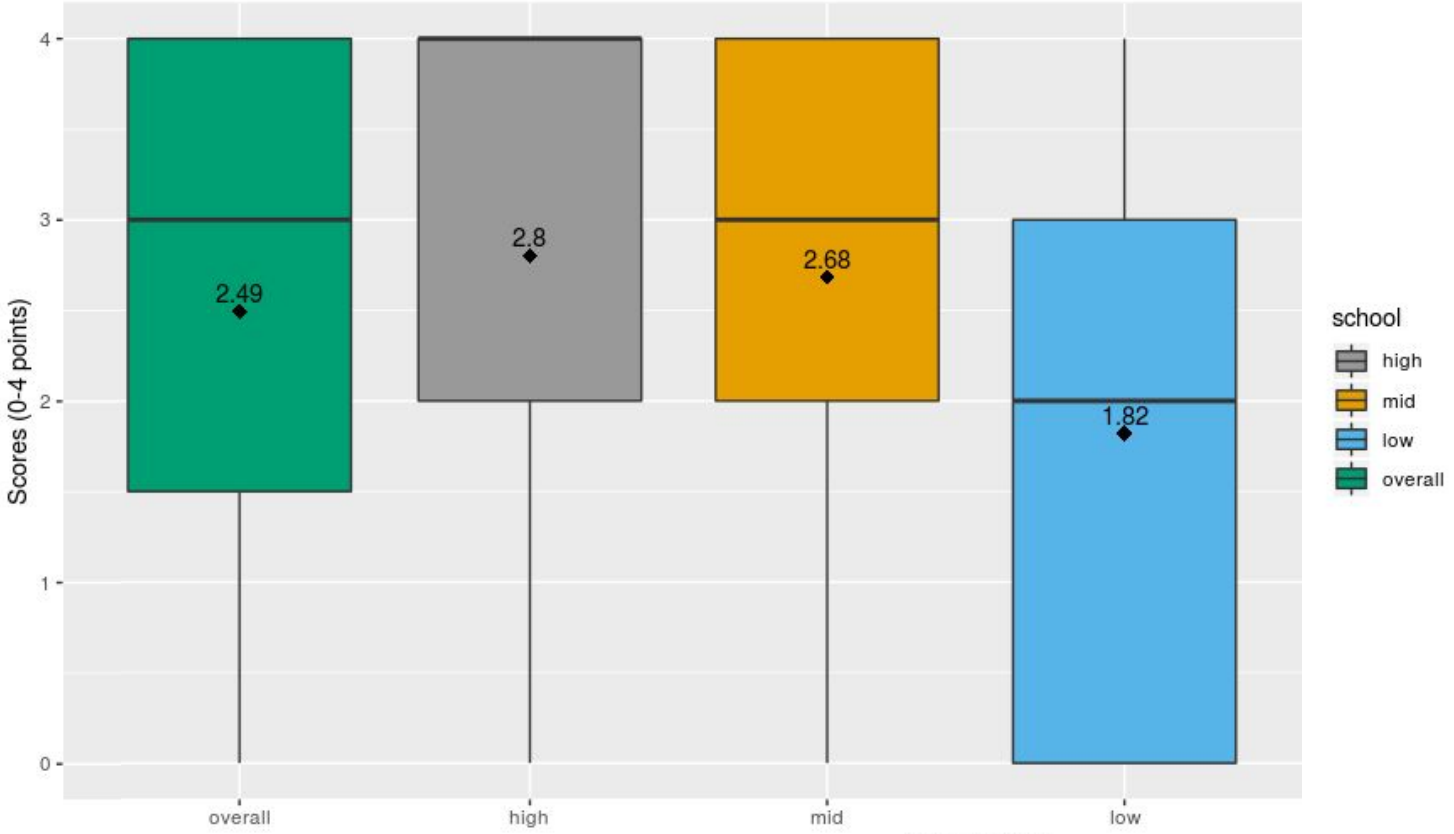
- ***Concept:*** Sequential execution with 1 event
- ***Question:*** Circle all the scripts that run when you click the sprite:



+2

-1

Events with 1 Script



Events with 1 Script

Breaking it down further...

- **61%** circled at least 1 correct scripts
- Only **43%** circled ALL correct scripts.

Key Takeaways

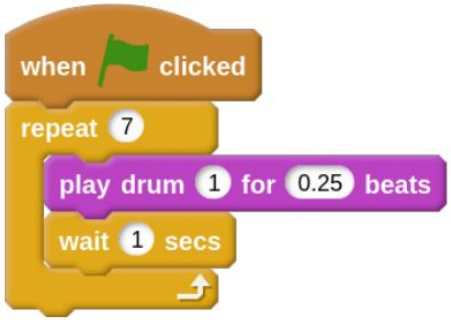
- High-performing school: More frequently had correct and complete answers
- Low-performing school: More frequently had incorrect and incomplete answers

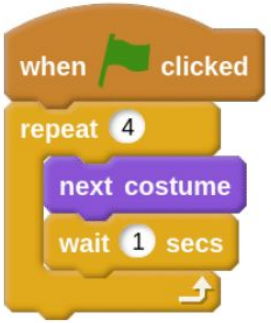
Events with Parallel Scripts

- ***Concept:*** Sequential vs Parallel Execution
- ***Question:*** Circle all the statements that are true about the scripts:

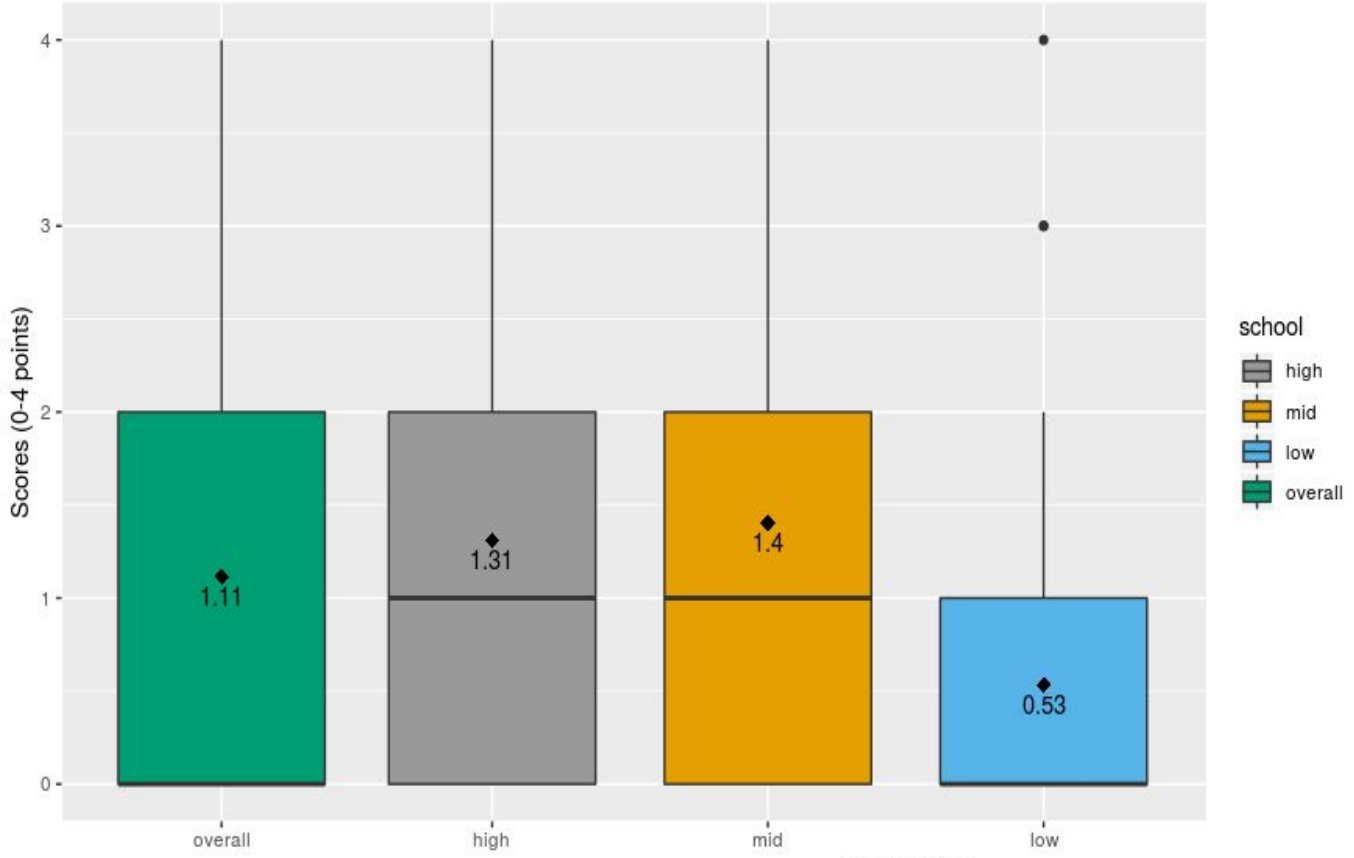
Pico's Code

Giga's Code





Events with Parallel Scripts



Events with Parallel Scripts

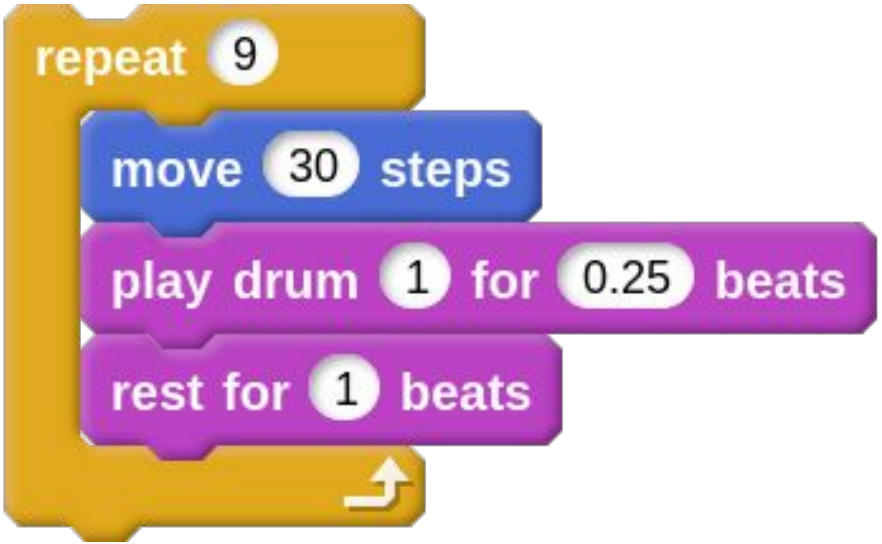
School	Pico Sequential	Giga Parallel
High	64%	41%
Mid	70%	37%
Low	47%	36%

Key Takeaways

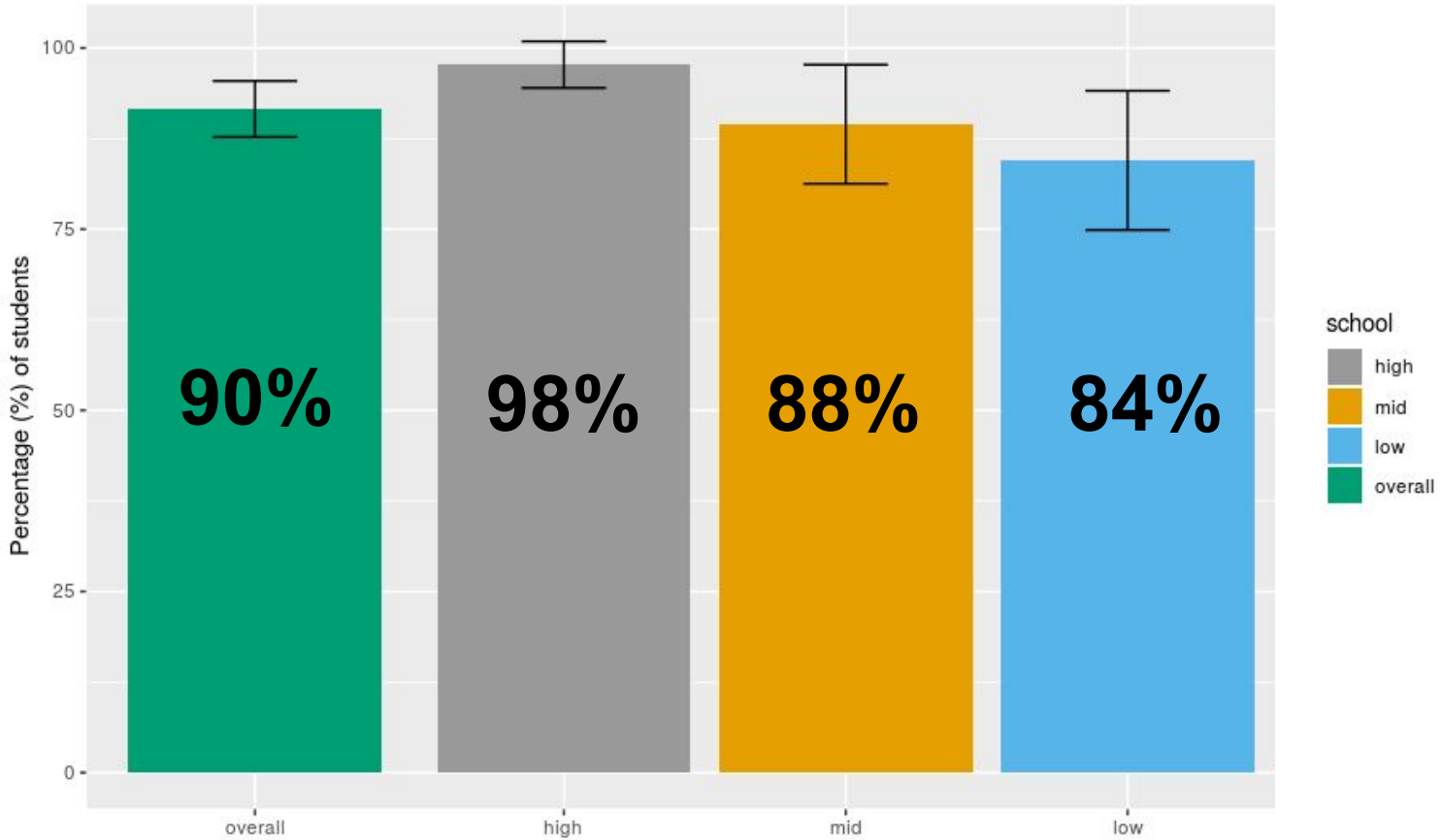
- Students understand sequence better than parallelism

Repetition Count

- ***Concept:*** Loop functionality
- ***Question:*** How many times will the loop below repeat?

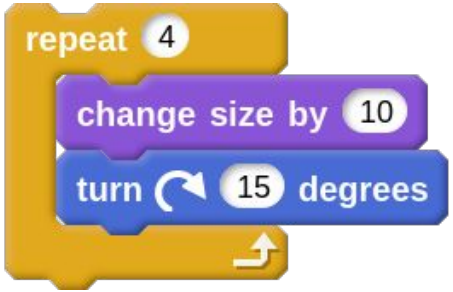


Repetition Count

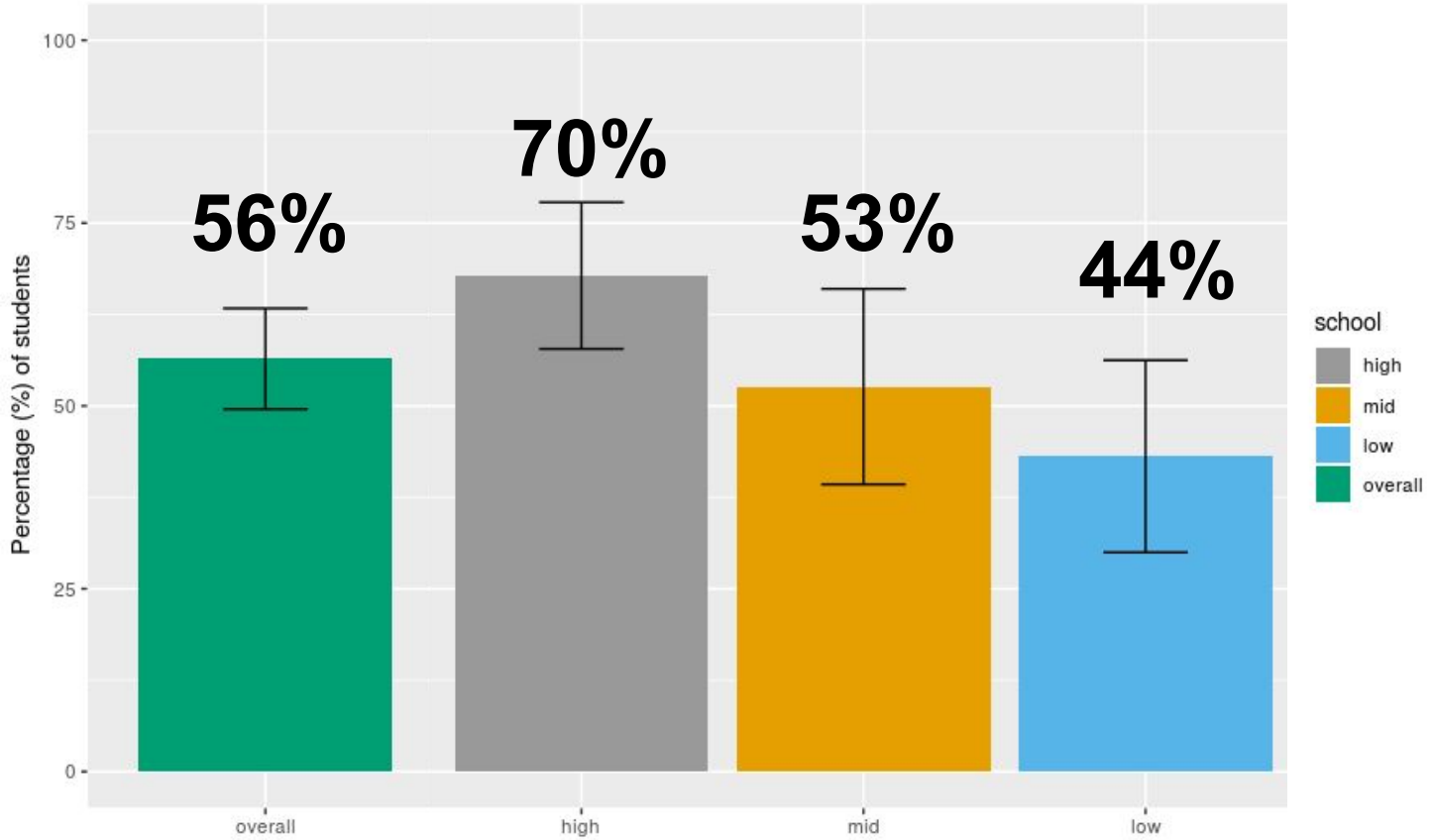


Unrolling a Loop

- ***Concept:*** Loop functionality
- ***Question:*** Circle the script that makes the sprite do the same thing as the loop



Unrolling a Loop



Unrolling a Loop

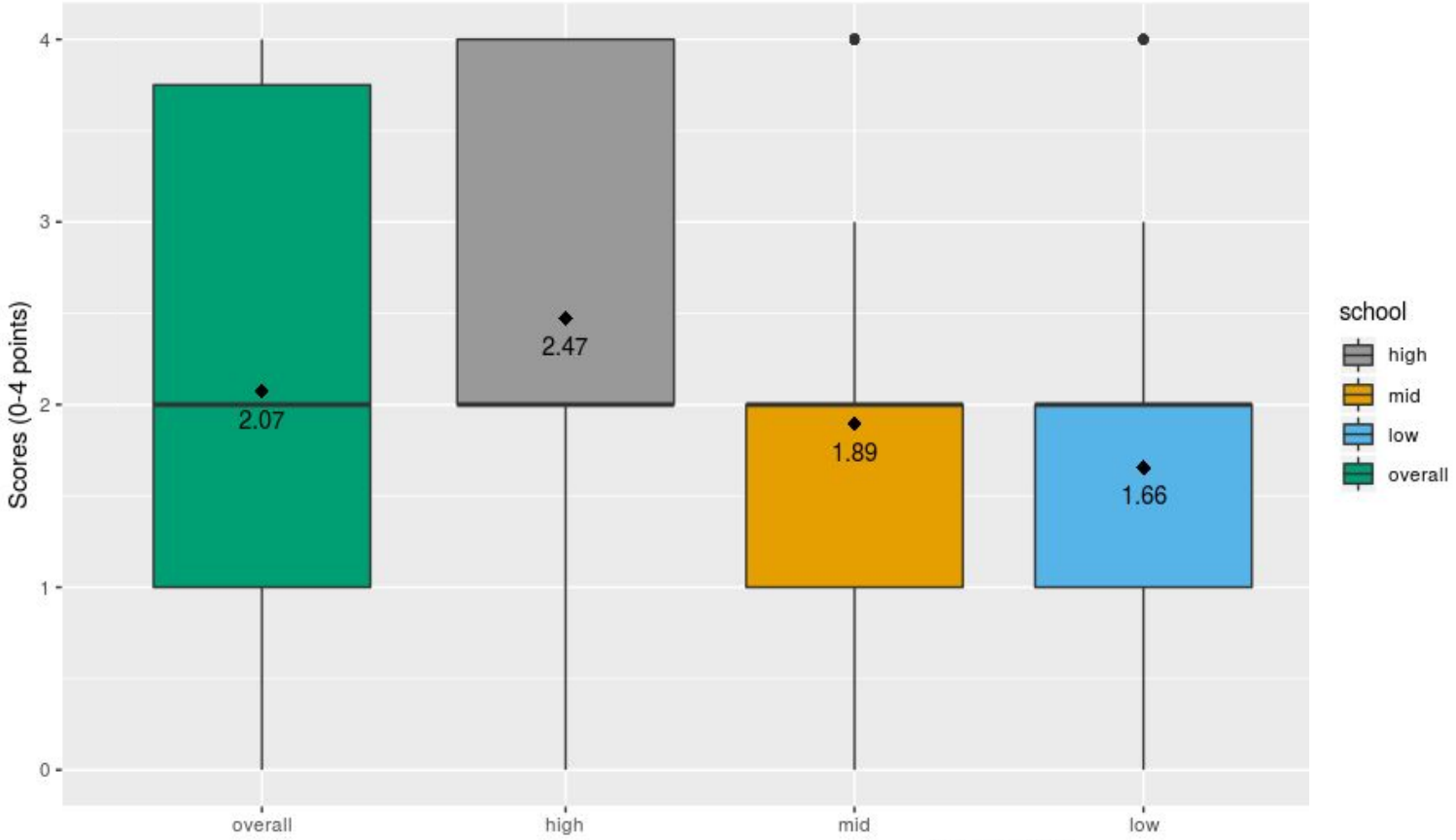
Key Takeaways

- Limited understanding of loop functionality

Repeated Blocks vs Loops

- ***Concept:*** Repeated blocks in loop vs loop iterations
- ***Question:*** Circle all the scripts that make the sprite play drum & change costume,both exactly 3 times.

Repeated Blocks vs Loops



Repeated Blocks vs Loops

Key Takeaways

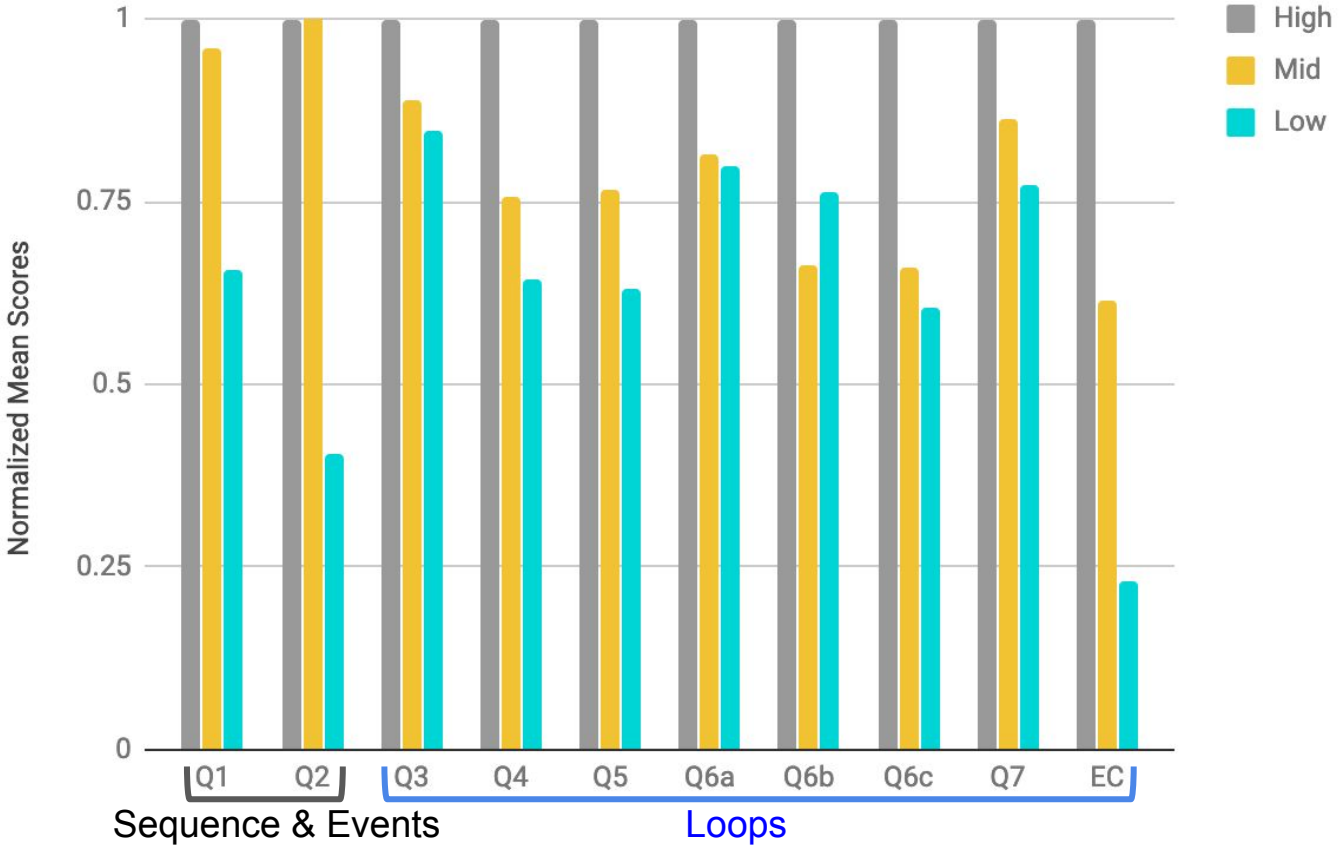
- 25% of high, 29% of mid, 45% of low chose:

- Students can’t distinguish between repeated blocks in a loop and a repeat loop

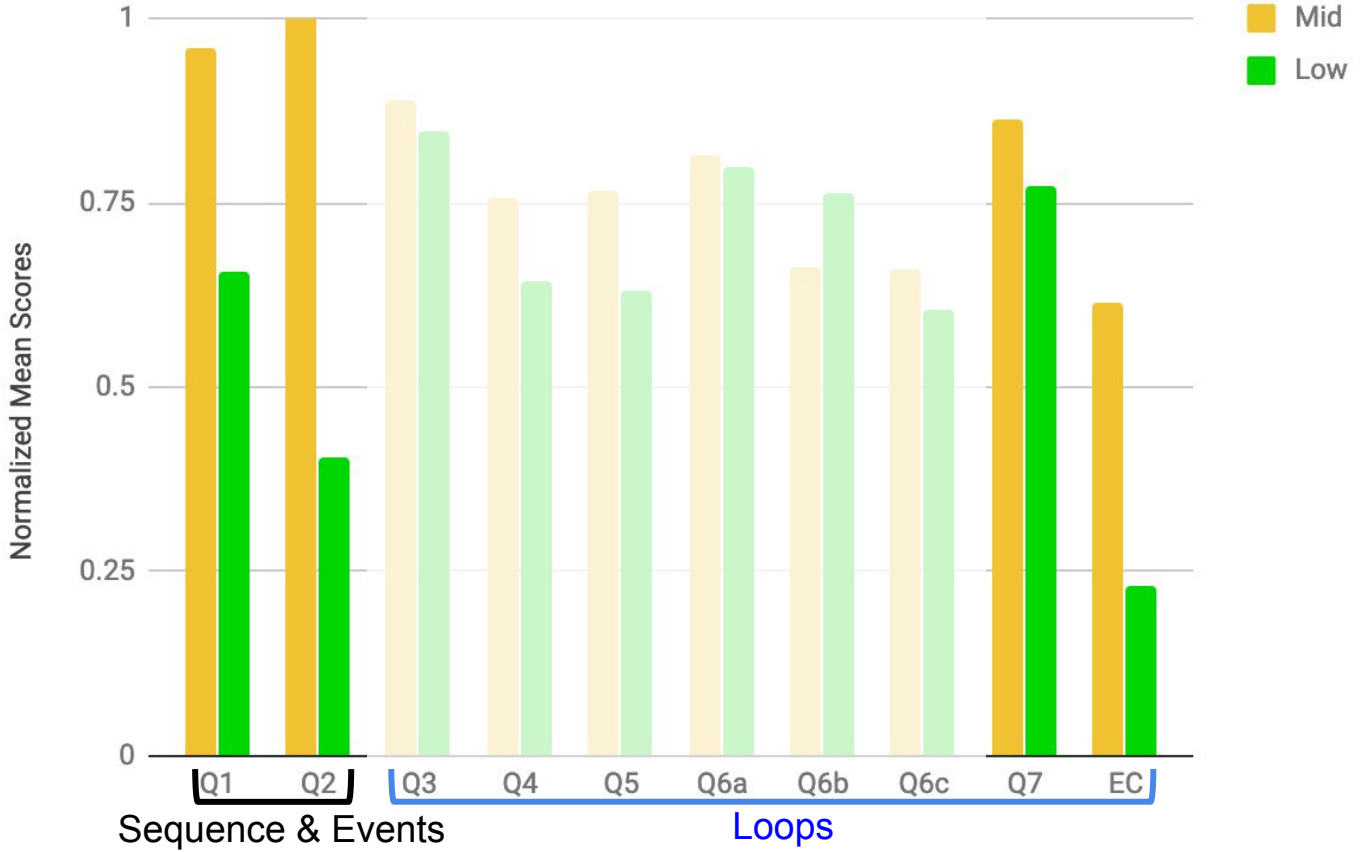
Overall Learning Outcomes

	Successes	Challenges
Events & Sequence	• Events starting one script	• Events starting parallel scripts
Loops	• Repetition Count • Code within a Loop	• Unrolling a Loop • Repeated blocks vs Repeat Loops • Code before & after a loop

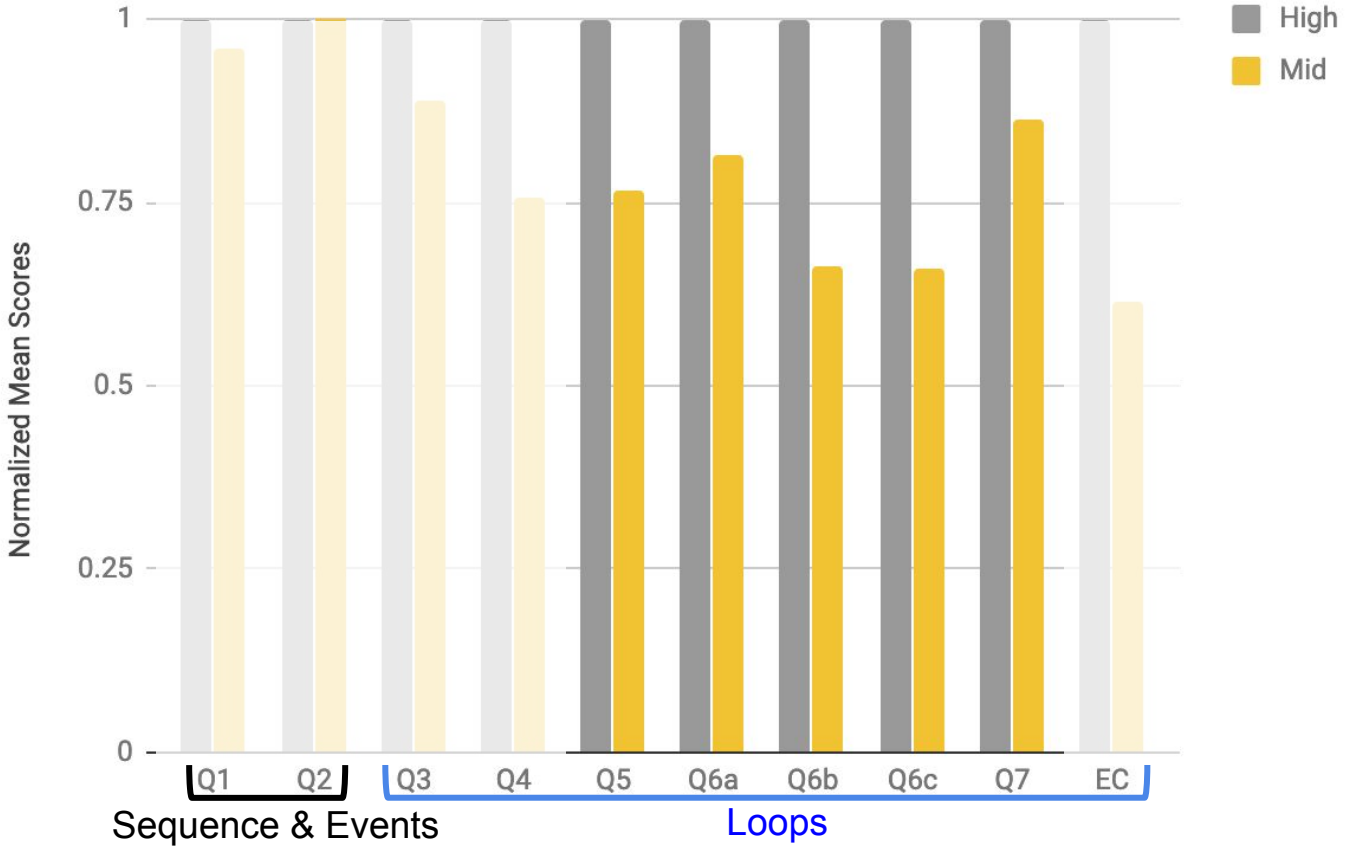
Overall Equity



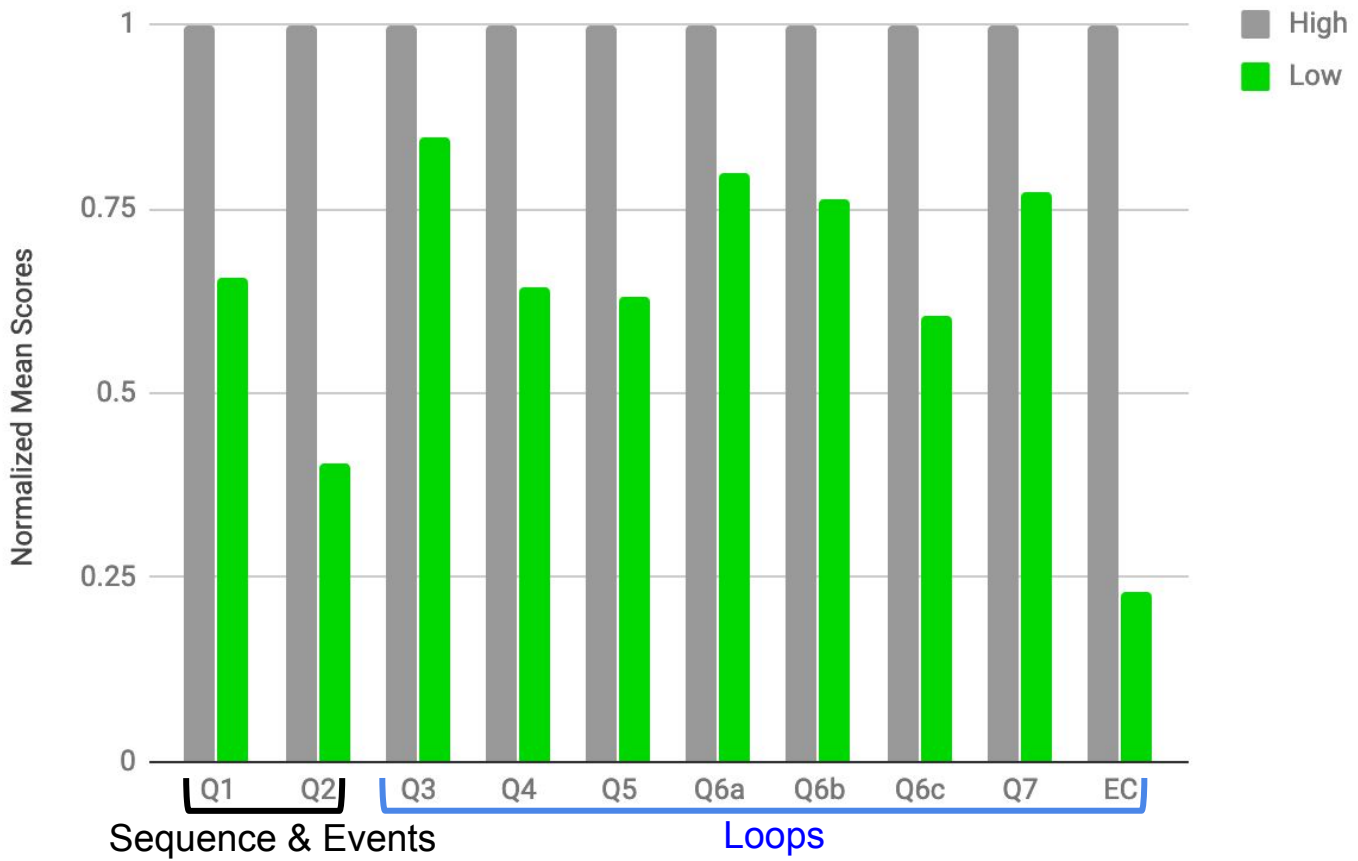
Significant Differences: M vs L



Significant Differences: H vs M



Significant Differences: H vs L



Conclusion

- Current instruction only supports *some, not all,* students.
- Underrepresented minority students clustered in schools under-served by current instruction.

What do we do about it?

1. Add more graphics & direct instruction
2. Use→ Modify→ Create pedagogy
3. Learning Strategy to guide students through Scratch exploration
4. Unplugged activity to elicit and correct student misconceptions

An Analysis through an Equity Lens of Computer Science in K-8 Classrooms

Jean Salac, Max White, Ashley Wang, Diana Franklin

Key Takeaways

- Current instruction only supports *some, not all,* students in learning sequence, events, & loops.
- Underrepresented minority students clustered in schools under-served by current instruction.

Many thanks to Bryan Twarek & Bill Marsland from SFUSD!!!

CANON LAB

Computing for ANyONe: Designing for equity and inclusion

This work was funded by NSF Award # 1660871

References

- Century,J., Ferris, K., and Zuo. Finding Time for Computer Science in the Elementary Day. url: http://outlier.uchicago.edu/TimeforCS/findings/.
- DiSalvo, B., Reid, C., & Roshan, P. K. (2014, March). They can't find us: the search for informal CS education. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 487-492). ACM.
- Fancsali, C., Tigani, L., Toro Isaza, P., & Cole, R. (2018, February). A Landscape Study of Computer Science Education in NYC: Early Findings and Implications for Policy and Practice. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 44-49). ACM.
- Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013, June). Designing ScratchJr: Support for early childhood learning through computer programming. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 1-10). ACM.
- Franklin, D., Skifstad, G., Rolock, R., Mehrotra, I., Ding, V., Hansen, A., ... & Harlow, D. (2017, March). Using upper-elementary student performance to understand conceptual sequencing in a blocks-based curriculum. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 231-236). ACM.
- Franklin, D., Hill, C., Dwyer, H. A., Hansen, A. K., Iveland, A., & Harlow, D. B. (2016, February). Initialization in Scratch: Seeking knowledge transfer. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 217-222). ACM.
- Grover, S., & Basu, S. (2017, March). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 267-272). ACM.
- Hill, C., Dwyer, H. A., Martinez, T., Harlow, D., & Franklin, D. (2015, February). Floors and Flexibility: Designing a programming environment for 4th-6th grade classrooms. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 546-551). ACM.
- Margolis, J. (2010). *Stuck in the shallow end: Education, race, and computing*. MIT Press.
- Seiter, L., & Foreman, B. (2013, August). Modeling the learning progressions of computational thinking of primary grade students. In *Proceedings of the ninth annual international ACM conference on International computing education research* (pp. 59-66). ACM.
- Weintrop, D., Hansen, A. K., Harlow, D. B., & Franklin, D. (2018, August). Starting from Scratch: Outcomes of early computer science learning experiences and implications for what comes next. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*(pp. 142-150). ACM.