CREATIVE TECHNOLOGY RESEARCH LAB

COLLEGE OF EDUCATION
UNIVERSITY OF FLORIDA

Unpacking Research on Supporting Students with Disabilities in Elementary CS

Maya Israel
University of Florida
misrael@coe.ufl.edu

Main Problem:

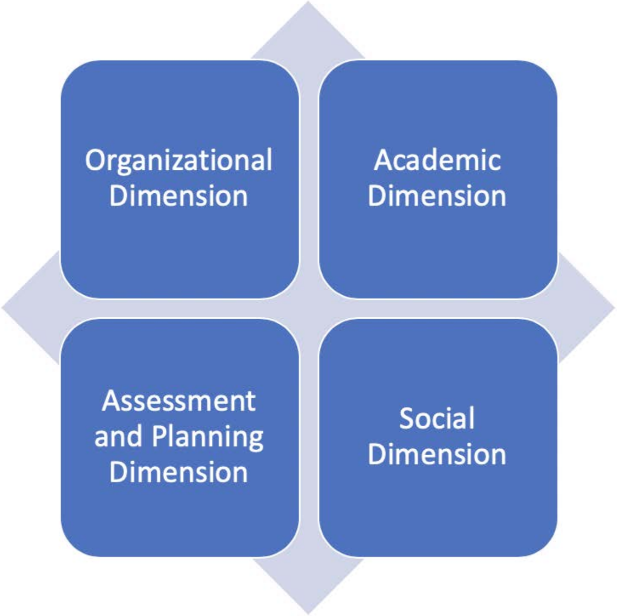
There is very little research about the inclusion of students with disabilities in elementary CS education.

CREATIVE TECHNOLOGY RESEARCH LAB

Multidimensional Approach to Inclusive Best Practices

(Crouch, Keys, & McMahon, 2014)

- **Organizational Dimension**
  - CS co-teaching, co-planning
- **Academic Dimension**
  - CS curriculum and instruction is accessible
  - Instructional supports in place
- **Assessment & Planning Dimension**
  - CS assessments address strengths & deficits
- **Social Dimension**
  - Students w/disabilities proportionally represented in CS



What we do know

(informed by Crouch et al., 2014)

Belief in CS for All including Students with Disabilities

Policy Examples:

- K-12 CS Framework- Equity language
- CSTA Equity Standards for CS Teachers

Accessible Computational Tools & Curricula

Examples

- Schanzer et al., (2019)
- Stefik & Siebert (2013)

3 Pillars of Inclusive CS Education

Effective Pedagogical Practices/supports

Examples:

- Israel et al., 2018
- Ray et al., 2018

What we don’t know:

To what extent are students w/disab included in CS education?

How does the intersectionality of disab, gender, SES, culture, etc play into CS learning?

How can we best support teachers in meeting the needs of ALL their learners?

What instructional supports help students who struggle in CS education?

CREATIVE TECHNOLOGY RESEARCH LAB

Tool and Curriculum Accessibility Barriers

- There is no such thing as "fully accessible" programming platforms for young learners:
  - Limited access with screen readers
  - Rely on visual representations
- Cognitive supports our typically outside the system (i.e., teachers)
- Many activities are open-ended

Individual Learning Barriers

- Struggle with programming **languages**
  - Challenges with decoding and comprehending the code
- Struggle with multi-step complex **problem solving**
  - Debugging a program that does not work
  - Strategically planning programs from the beginning to end
- Results=**Frustration** and task abandonment (Israel et al., under review)

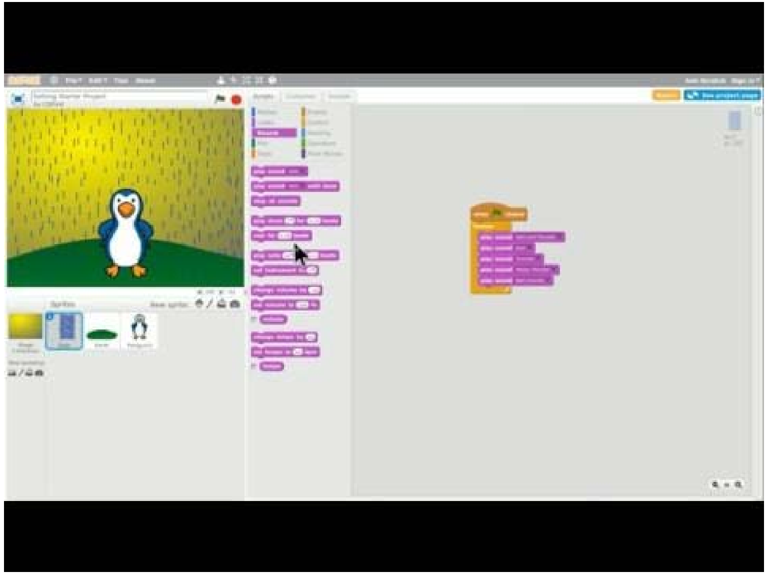
CREATIVE TECHNOLOGY RESEARCH LAB

Addressing Inclusion Barriers

- Use accessible tools (e.g., Bootstrap, Quorum)
- Provide individualized supports, accommodations, and strategies effective in other content areas (e.g., Snodgrass et al., 2016)
- Provide teachers with professional development on strategies and applications of Universal Design for Learning (UDL) in CS education (e.g., Israel et al., 2020)
- Use explicit instruction (Wright et al., 2019) in a balanced approach within open ended computational activities.

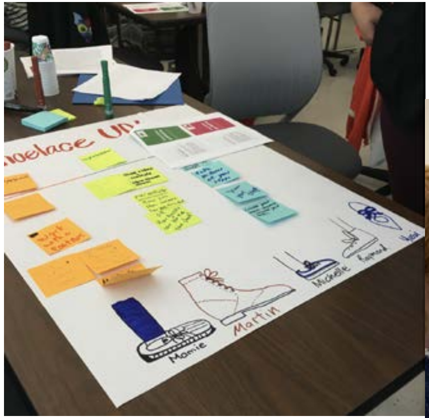
Our Approach to Studying Inclusive CS Education

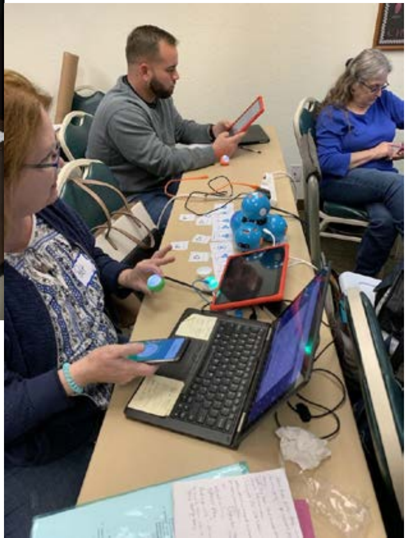
Student-level deep analysis of computational processes



(e.g., Israel et al., under review)

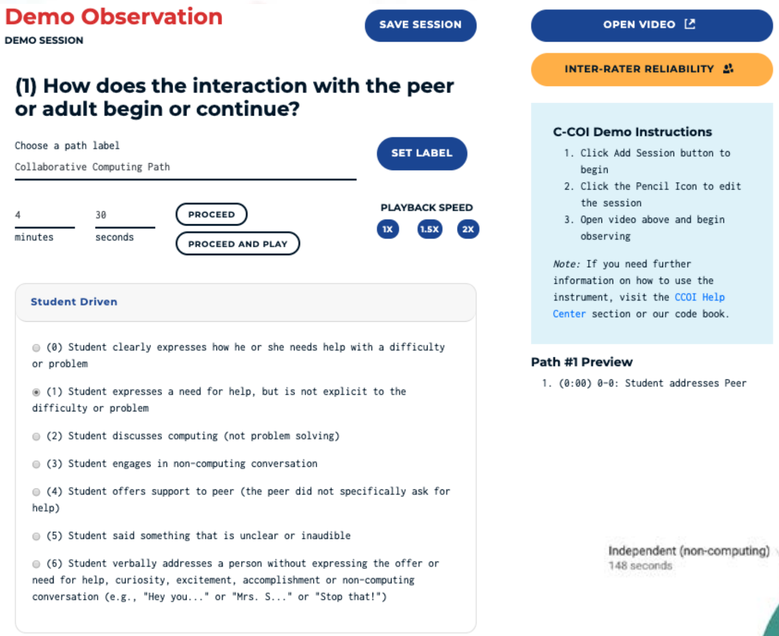
Teacher-level analysis of PD and instructional practices





(e.g., Israel et al., 2020)

Studying Student Engagement and Learning



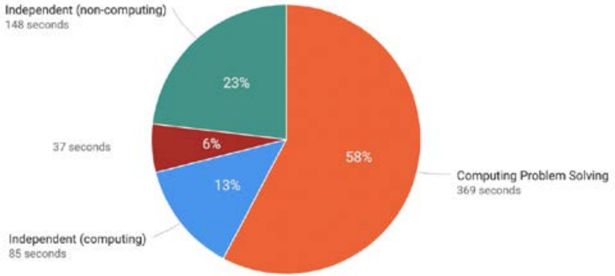
Data includes:

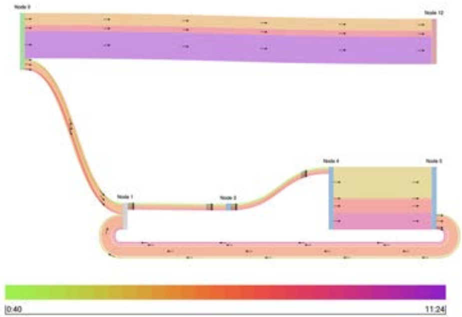
-Videos of students' computational behaviors

-Student observations

-New: Eye tracking/gaze fixation

-Artifact-based interviews





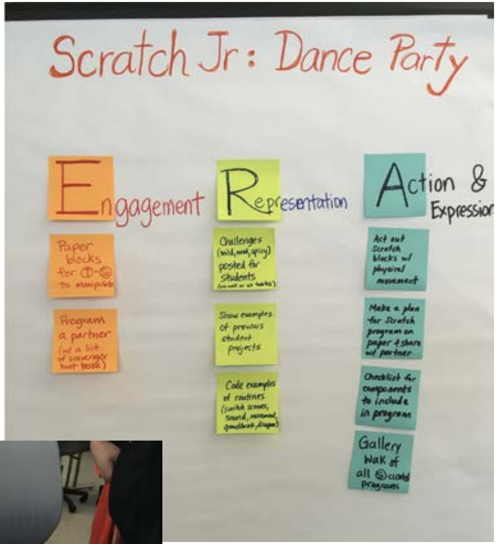
CREATIVE TECHNOLOGY
RESEARCH LAB

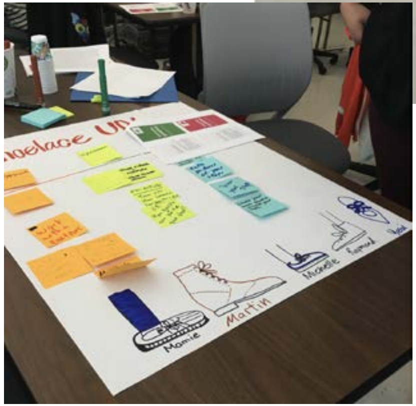
Teacher Learning, Agency, and Instructional Practices

Universal Design for Learning Guidelines + Computer Science / Computational Thinking

	Multiple Means of **Engagement**	Multiple Means of **Representation**	Multiple Means of **Action & Expression**
	Affective Networks The "WHY" of learning	Recognition Networks The "WHAT" of learning	Strategic Networks The "HOW" of learning
Access	Provide options for **Recruiting Interest** • Give students choices (choose project, software, topic) • Allow students to make projects relevant to culture and age • Minimize possible common "pitfalls" for both computing and content • Allow for differences in pacing and length of work sessions • Provide options to increase or decrease sensory stimulation (for example listening to music with headphones or using noise cancelling headphones) • Allow for differences in pacing and length of work sessions	Provide options for **Perception** • Model computing using physical representations as well as through an interactive whiteboard, videos • Give access to modeled code while students work independently • Provide access to video tutorials of computing tasks • Select coding apps and websites that allow the students to adjust visual settings (such as font size & contrast) and that are compatible with screen readers.	Provide options for **Physical Action** • Provide teacher's codes as templates • Include CS Unplugged activities that show physical relationship of abstract computing concepts • Use assistive technology including larger/smaller mice, touch-screen devices • Select coding apps and websites that allow coding with keyboard shortcuts in addition to dragging & dropping with a mouse

Israel, M., Lash, T., Ray, M. (2017). Universal Design for Learning within Computer Science Education. *Creative Technology Research Lab*. University of Florida.





References

Israel, M., Ray, M. J., Maa, W. C., Jeong, G. K., eun Lee, C., Lash, T., & Do, V. (2018). School-embedded and district-wide instructional coaching in K-8 computer science: Implications for including students with disabilities. *Journal of Technology and Teacher Education*, *26*(3), 471-501.

Israel, M., Jeong, G., Ray, M., & Lash, T. (2020). Teaching elementary computer science through universal design for learning. *In Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 1220-1226).

K-12 Computer Science Framework Steering Committee. (2016). K-12 computer science framework.

Ray, M. J., Israel, M., Lee, C. E., & Do, V. (2018, February). A cross-case analysis of instructional strategies to support participation of K-8 Students with disabilities in CS for All. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 900-905).

Schanzer, E., Bahram, S., & Krishnamurthi, S. (2019, February). Accessible AST-Based Programming for Visually-Impaired Programmers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 773-779).

Snodgrass, M. R., Israel, M., & Reese, G. C. (2016). Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. *Computers & Education*, *100*, 1-17.

Stefik, A., & Siebert, S. (2013). An empirical investigation into programming language syntax. *ACM Transactions on Computing Education (TOCE)*, *13*(4), 1-40.

Wright, J. C., Knight, V. F., Barton, E. E., & Edwards-Bowyer, M. (2019). Video Prompting to Teach Robotics and Coding to Middle School Students With Autism Spectrum Disorder. *Journal of Special Education Technology*, 0162643419890249.

Extra slides

About recent UDL in CS Ed PD this Study

Primary RQ: What does UDL look like in elementary CS education?

Methodology: Interpretive qual. triangulating lesson plans/artifacts, coaching logs, and interviews.

Name	Grade	Class Setting	License	Years teaching
Lester	2	Inclusive co-taught	Special and General Ed	4
Rios	K-3	Self-contained special ed	Special ed	4
Aldridge	3	Inclusive co-taught	General Education	2
Robinson	4	Gen Ed and Eng. as a New Language (ENL)	General Ed	3

Data Analysis

- Coding: UDL checkpoints
- Categorized data by similarities/ differences across teachers
- Interrater reliability among the two coders: Cohen's Kappa (above .98)

Provide options for
Executive Functions (6)

- Guide appropriate goal-setting (6.1)
- Support planning and strategy development (6.2)
- Facilitate managing information and resources (6.3)
- Enhance capacity for monitoring progress (6.4)

Provide options for
Recruiting Interest (7)

- Optimize individual choice and autonomy (7.1)
- Optimize relevance, value, and authenticity (7.2)
- Minimize threats and distractions (7.3)

CT & CS Activities

- Combination of plugged & unplugged activities
- Begin with unplugged, transition to simple plugged, and finally to more sophisticated plugged activities

Computing activity	Lester	Rios	Aldridge	Robinson
Unplugged	x	x	x	x
Robotics	x	x	x	x
Code.org	x		x	x
Scratch	x	x	x	x
Codesters				x
Stations		x		

Results...MORE CHECKPOINTS ≠ BETTER

Engagement (n=63 instances)	**Representation** (n=51 instances)	**Action and Expression** (n=31 instances)
*Choice in computational artifacts (with accountability) *Personally relevant/real-world applications *Student collaboration	*Options for perception (e.g., customize display of info) *Explicit directions, visual cues, modeling *Less for on supports for language and symbols	*Focus on executive function -Goal setting -Planning support -Progress monitoring *Less focus on options for physical action or options for expression and communication

Significant differences in UDL implementation across the teachers.

UDL is contextual.

**UDL checkpoints DO NOT hold equal weight in individual lessons: They are context and student dependent!