Including All Learners in K-12 CS Education through Universal Design for Learning

Maya Israel
@misrael09

CREATIVE TECHNOLOGY RESEARCH LAB
COLLEGE OF EDUCATION
UNIVERSITY OF FLORIDA

Primary Goal:

Increase access to CS education for students with disabilities and others at risk for academic failure.

CREATIVE TECHNOLOGY RESEARCH LAB

Research Projects focused on UDL & Inclusion

TIME FOR CS

MICHIGAN STATE UNIVERSITY

INFACT

UDL4CS

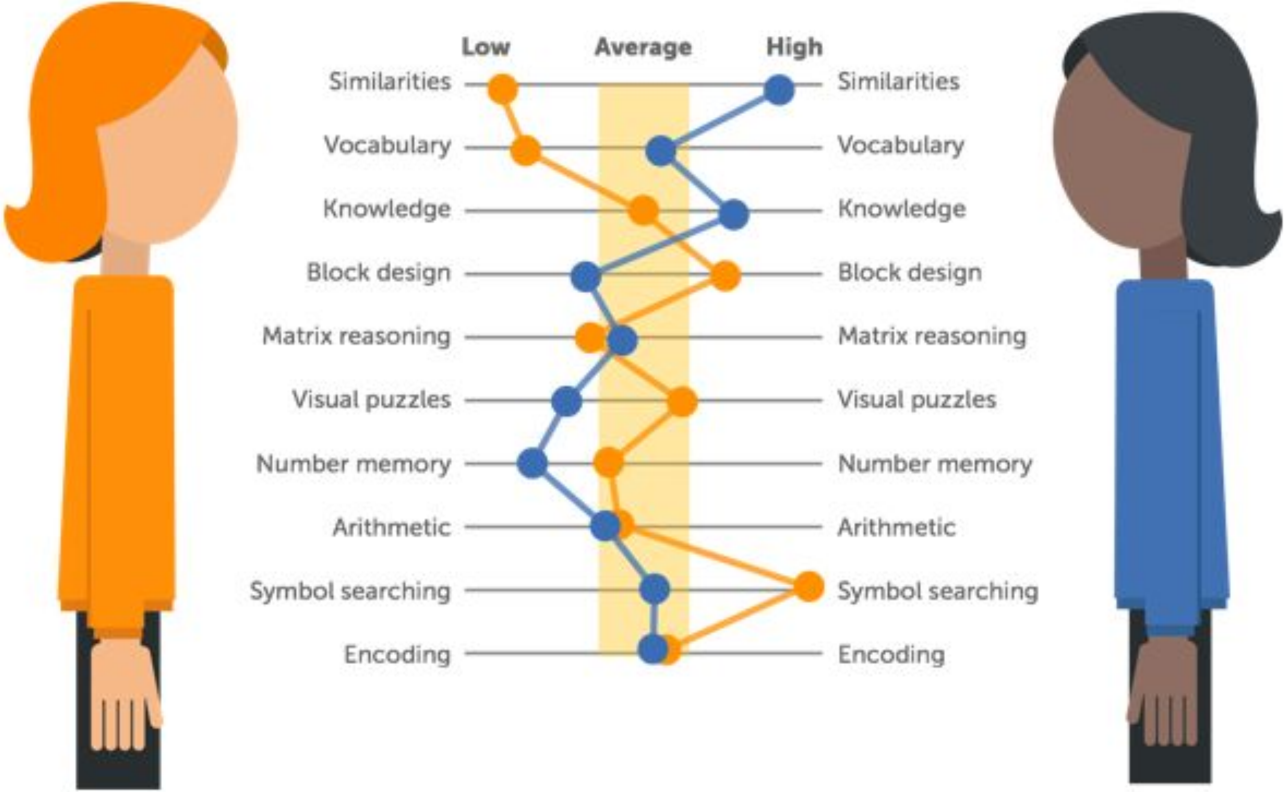
UNIVERSAL DESIGN FOR LEARNING FOR COMPUTER SCIENCE

LTEC

Learning Trajectories for Everyday Computing

We all have a jagged learning profile

https://digitalpromise.org/2019/01/14/powerful-learning-personal-accessible/



A Few Statistics about Students with Disabilities in the United States

- ~ 7.5 million students w/disabilities are in U.S. public schools (~13%).

- Majority receiving special education services includes Learning Disabilities (LD) category (approx. 35%)

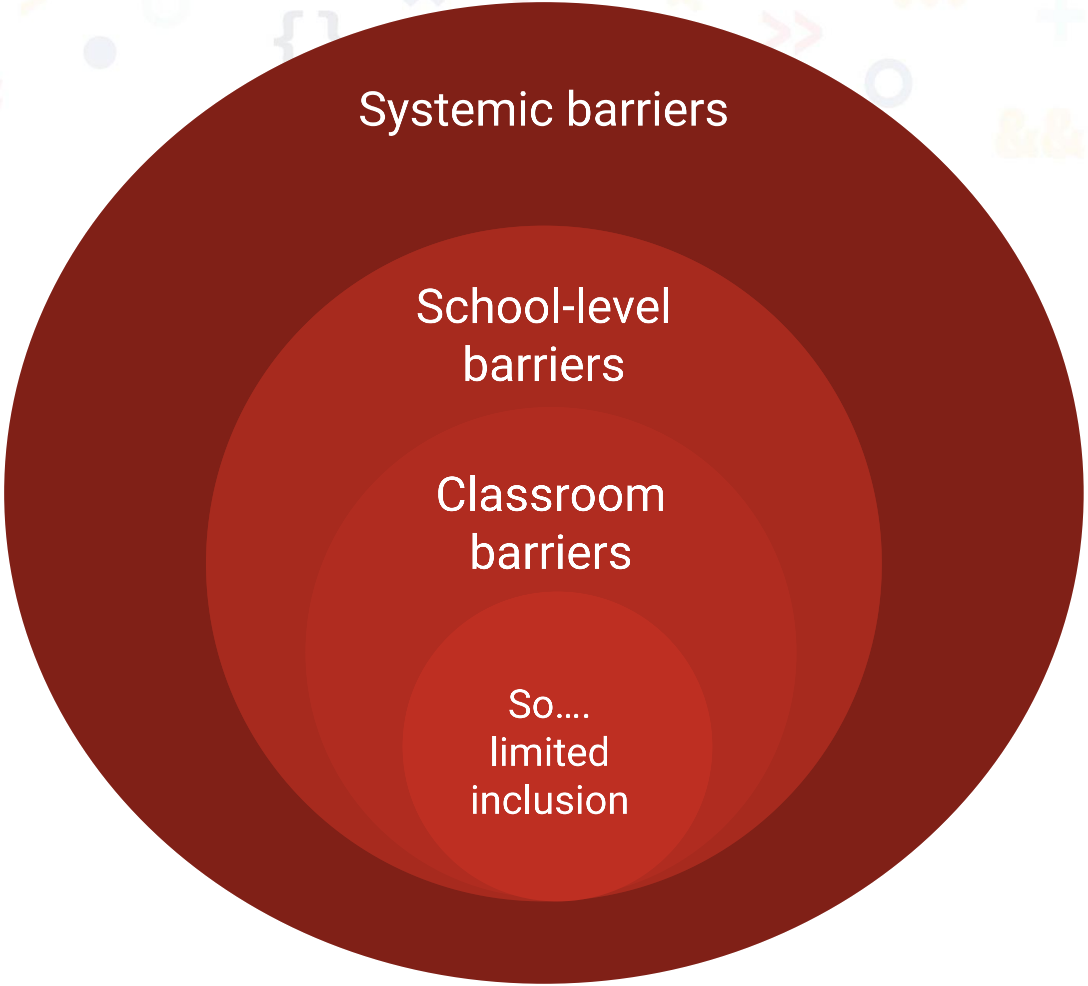
- The next largest categories are Speech/language, Other health impairments, and Autism.

- Most students with disabilities are taught alongside their peers.

Barriers to Inclusion in K-12 CS Education

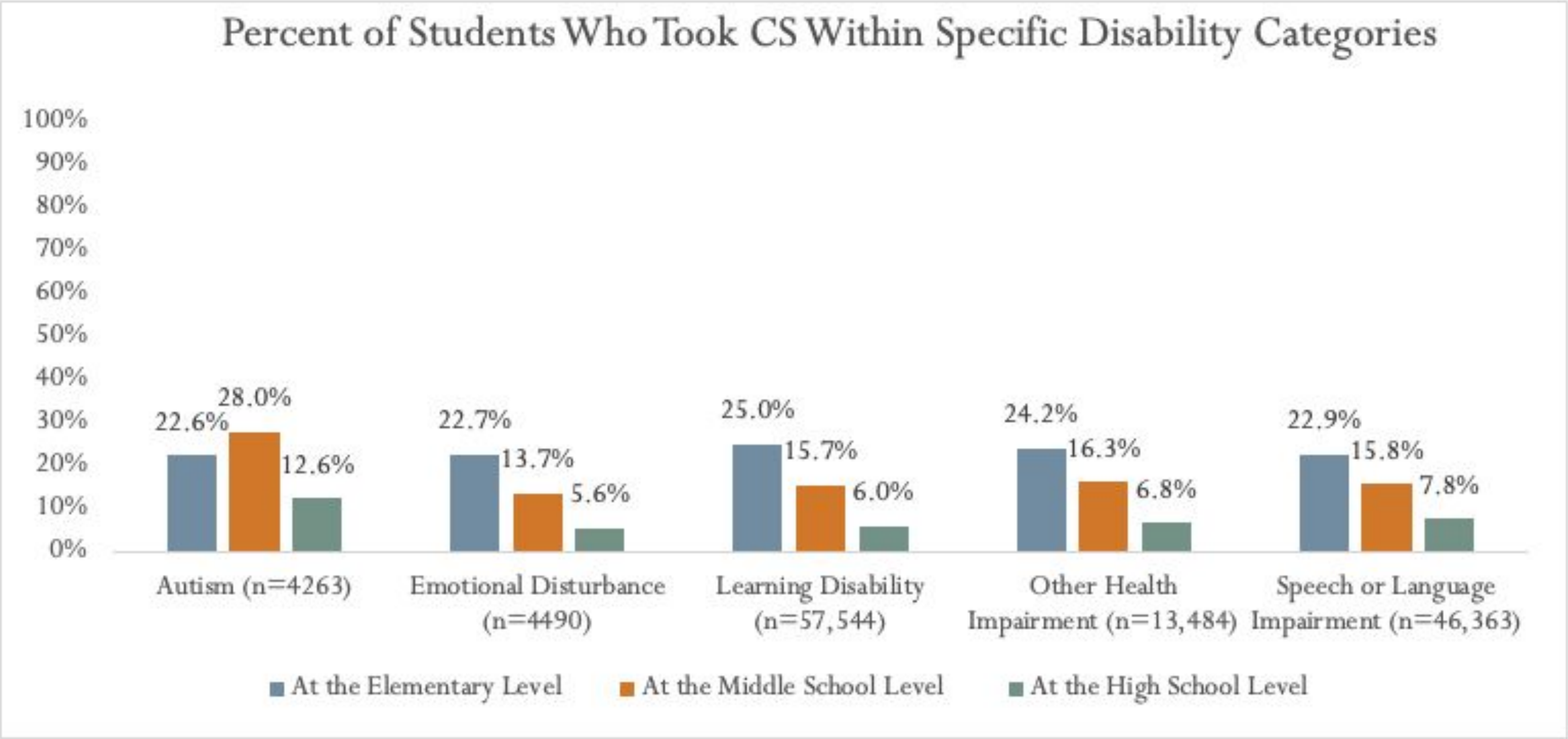
CREATIVE TECHNOLOGY RESEARCH LAB

This work cannot occur if we do not examine barriers to inclusion in a systematic way.



CREATIVE TECHNOLOGY RESEARCH LAB

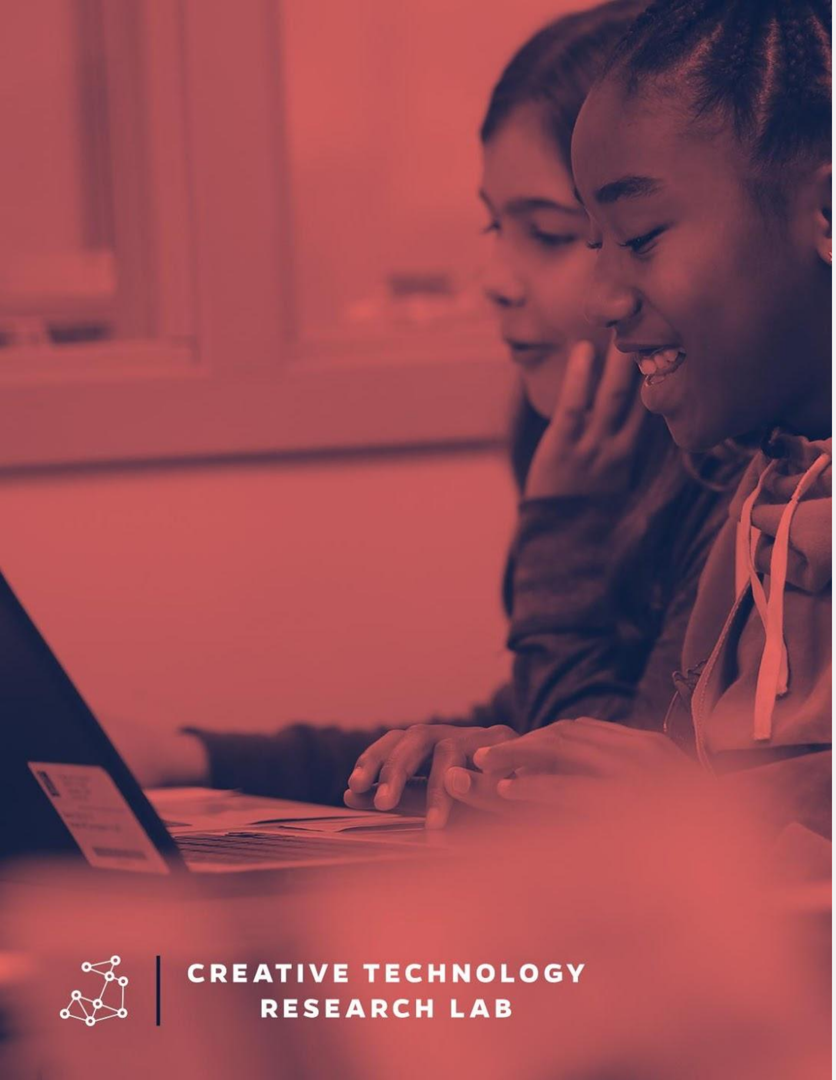
Example of what we see when we look at the data



Paper in preparation

What are Student-Level Barriers?

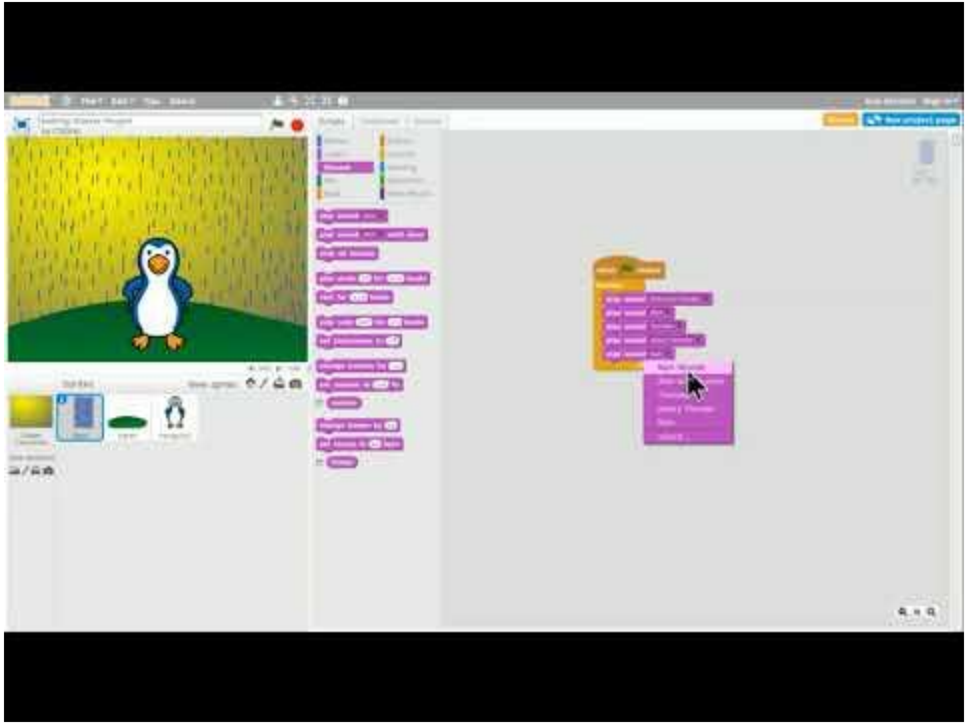
- Programming **languages**
  - Decoding/comprehending code
- Multi-step complex **problem solving**
  - Debugging
  - Strategically planning programs from the beginning to end
- **Frustration** and task abandonment (Israel et al., 2020)



What are the Tool and Curricular Accessibility Barriers?

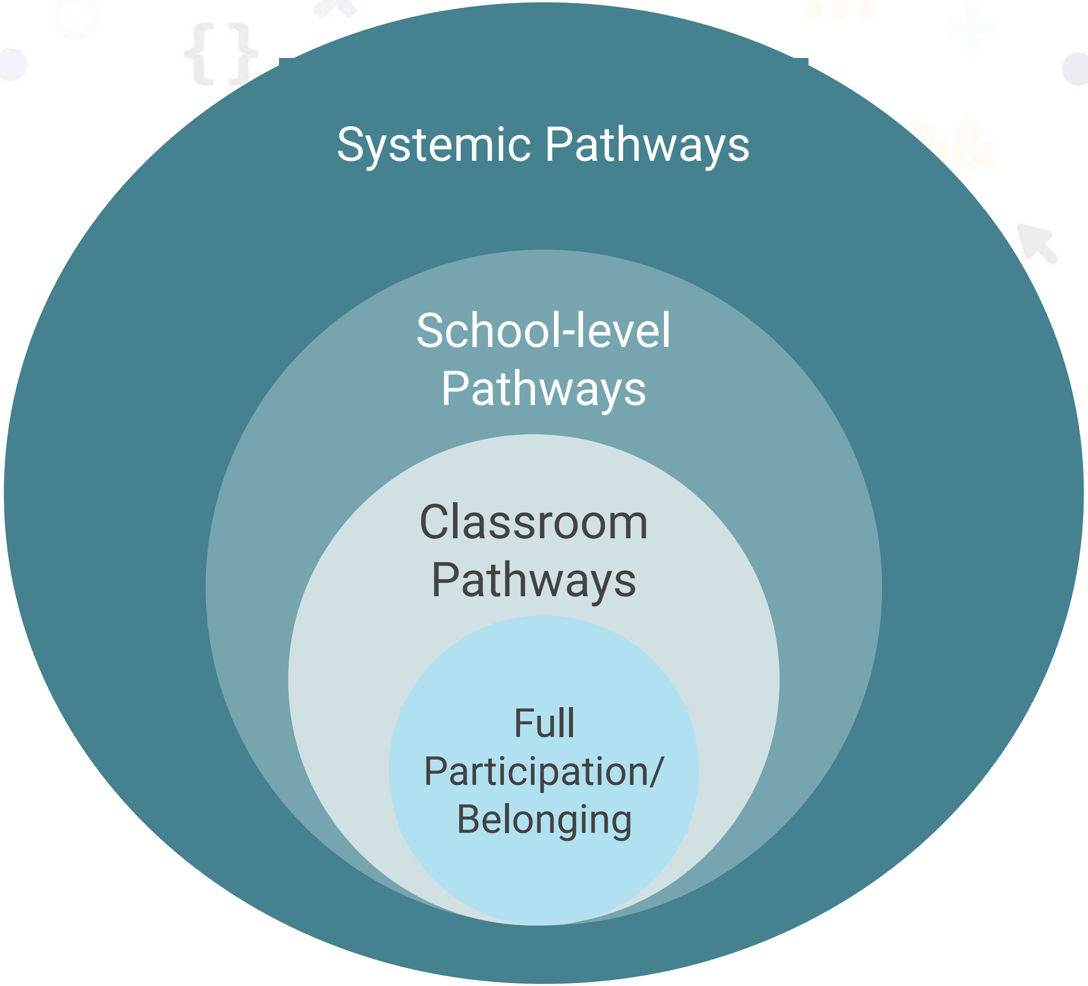
- There is no such thing as “fully accessible” programming platforms:
  - Limited access with screen readers
  - Rely on visual representations
- Cognitive supports are typically outside the system (i.e., teachers)
- Many activities are open-ended with limited structure

Video Example: “It’s Glitching!”



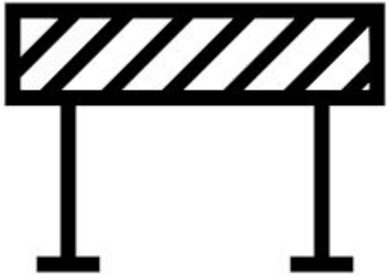
Pathways to Inclusion

Just like with barriers, we must consider pathways in a systematic way.



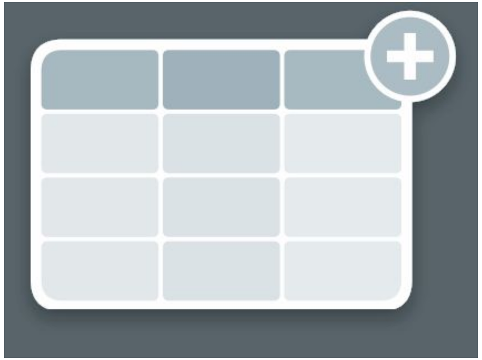
CREATIVE TECHNOLOGY RESEARCH LAB

The UDL Core Concepts



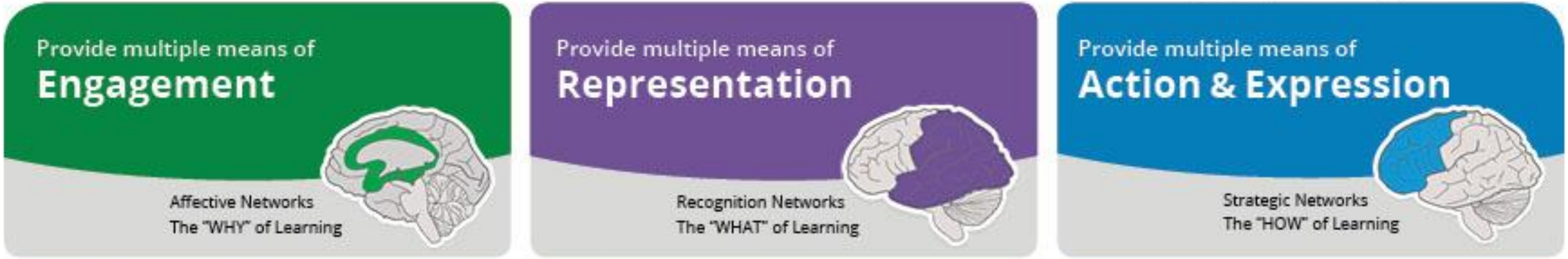
Barriers exist in the learning Environment

Variability is the Norm/Jagged Learning Profiles



The Goal is Expert Learning

Enter Universal Design for Learning



Access

Build

Internalize

Expert learners who are...

Purposeful & Motivated	Resourceful & Knowledgeable	Strategic & Goal-Directed

The UDL guidelines:

Strategies applied to any discipline so all learners can access & participate in meaningful, challenging learning opportunities.

	Provide multiple means of **Engagement** ➔ Affective Networks The "WHY" of learning	Provide multiple means of **Representation** ➔ Recognition Networks The "WHAT" of learning	Provide multiple means of **Action & Expression** ➔ Strategic Networks The "HOW" of learning
Access	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) ›	Provide options for **Perception** (1) • Offer ways of customizing the display of information (1.1) › • Offer alternatives for auditory information (1.2) › • Offer alternatives for visual information (1.3) ›	Provide options for **Physical Action** (4) • Vary the methods for response and navigation (4.1) › • Optimize access to tools and assistive technologies (4.2) ›
Build	Provide options for **Sustaining Effort & Persistence** (8) • Heighten salience of goals and objectives (8.1) › • Vary demands and resources to optimize challenge (8.2) › • Foster collaboration and community (8.3) › • Increase mastery-oriented feedback (8.4) ›	Provide options for **Language & Symbols** (2) • Clarify vocabulary and symbols (2.1) › • Clarify syntax and structure (2.2) › • Support decoding of text, mathematical notation, and symbols (2.3) › • Promote understanding across languages (2.4) › • Illustrate through multiple media (2.5) ›	Provide options for **Expression & Communication** (5) • Use multiple media for communication (5.1) › • Use multiple tools for construction and composition (5.2) › • Build fluencies with graduated levels of support for practice and performance (5.3) ›
Internalize	Provide options for **Self Regulation** (9) • Promote expectations and beliefs that optimize motivation (9.1) › • Facilitate personal coping skills and strategies (9.2) › • Develop self-assessment and reflection (9.3) ›	Provide options for **Comprehension** (3) • Activate or supply background knowledge (3.1) › • Highlight patterns, critical features, big ideas, and relationships (3.2) › • Guide information processing and visualization (3.3) › • Maximize transfer and generalization (3.4) ›	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) ›
Goal	**Expert Learners** who are... **Purposeful & Motivated**	**Resourceful & Knowledgeable**	**Strategic & Goal-Directed**

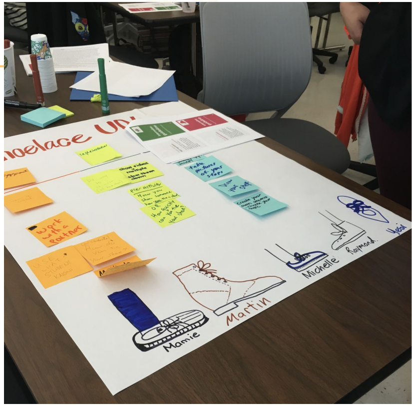
UDL in preK-12 CS Education

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.

CREATIVE TECHNOLOGY RESEARCH LAB



UDL principle	Broad UDL Application	UDL in CS/CT digital & unplugged activities
The "Why" of Learning	⇨ Offer kids choice in activities. Ensure activities are based on ideas that are important to them and relate to their diverse lived experiences. ⇨ Make computing learning goals clear (to the kids and also in materials for parents) and regularly check for understanding in all media. ⇨ Provide multiple entry points for computing activities with varying levels of challenge. Encourage kids to experiment with different entry points. Consider the use/modify/create as one framework for multiple entry points	⇨ CS/CT games are flexible and include activities that are familiar to kids' daily lives. For example, in Scratch Jr. kids are encouraged to pick sprites, actions, and backgrounds to make movies. ⇨ Embed "I can" statements in computational activities such as "I can make the cat jump four times" or "I can sequence the story into beginning, middle, and end". ⇨ Use icons like thumbs up/down or a gauge that moves towards increased understanding throughout activities so kids can check for understanding. ⇨ Offer varied options such as (a) Play the game/program, (b) add to the game/program, (c) debug game/program, (d) sequence existing code in the game/program.

Example: Continuum of Guided to Open-ended Exploration to Teach Debugging

Play and Remix Existing Code	Debug "buggy" program	Construct "exploded" code
Students play & remix with code that has been constructed	Students debug a program that is not working	Students reconstruct a code that has been deconstructed

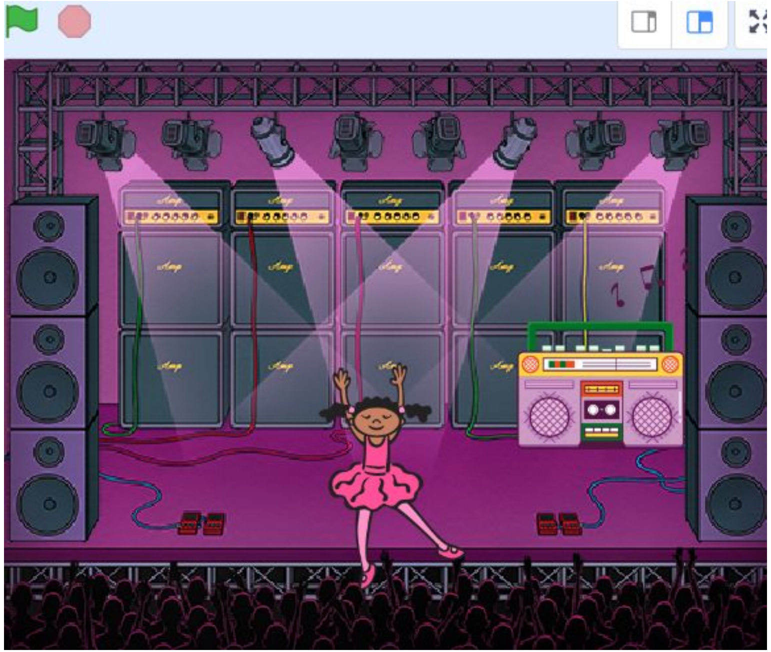
Scaffolded Scratch Projects

Dancing Ballerina Play and remix

Dancing Ballerina Exploded Code

Dancing Ballerina Buggy Program

Dancing Ballerina Spicy Expansion



CREATIVE TECHNOLOGY
RESEARCH LAB

Our Research on UDL-based Inclusive Practices

Study 1: Special Education Teachers in Promoting Access to CS Education: BrowardCODES-for-All

APPLIED LEARNING DEPARTMENT, STEM+CS PROGRAMS

Established 1915 BROWARD County Public Schools

BROWARD S.T.E.M. + COMPUTER SCIENCE

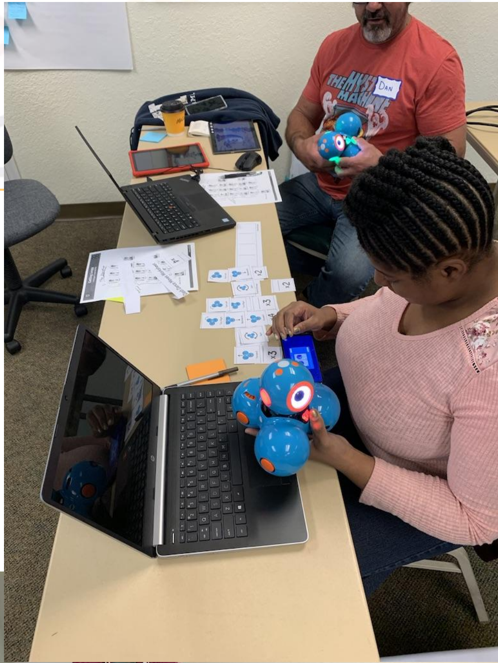
BROWARDCODES FOR ALL- INCLUSION IN COMPUTER SCIENCE

CS FOR ALL

This course is for special education teachers and staff to prepare them to support students in com science (CS) coursework.

This course and the resources provided in it can utilized by general education teachers in support students with disabilities in CS education, includ of Universal Design for Learning (UDL) principles planning and development, the use of assistive t in CS education, and other high-leverage practic supporting all learners.

Overview
Cohort Info
Computer Science (CS)
Universal Design for Le...
CS Materials



CREATIVE TECHNOLOGY RESEARCH LAB

Research Questions

1. What resources do special education teachers need in order to provide CS education to students with disabilities?

2. What features of professional development empower teachers to provide meaningful CS education to students with disabilities?

CREATIVE TECHNOLOGY RESEARCH LAB

APPLIED LEARNING DEPARTMENT, STEM+CS PROGRAMS

BROWARDCODES FOR ALL- INCLUSION IN COMPUTER SCIENCE

CS FOR ALL

This course is for special education teachers and support staff to prepare them to support students in computer science (CS) coursework.

This course and the resources provided in it can also be utilized by general education teachers in supporting students with disabilities in CS education, including the use of Universal Design for Learning (UDL) principles in lesson planning and development, the use of assistive technologies in CS education, and other high-leverage practices for supporting all learners.

Professional Development Content

1. Inclusive and accessible CS instruction based on Universal Design for Learning (UDL; CAST, 2011)
2. Individualizing CS for students with more significant needs
3. Exploration of accessibility features within available CS software and hardware

[Example Agenda]

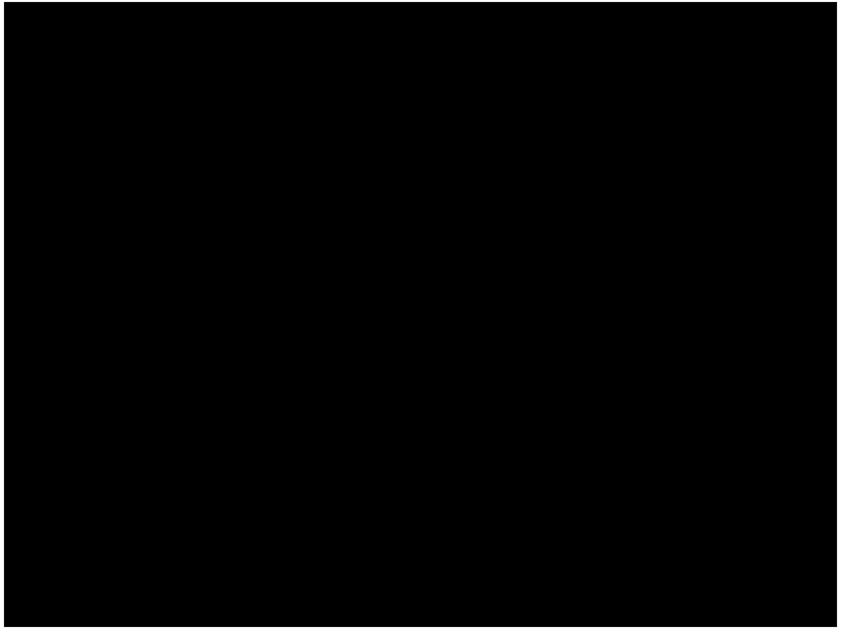
CREATIVE TECHNOLOGY RESEARCH LAB

Time to brainstorm challenges and strategies was key

Barriers/Challenges	Strategies to overcome
Navigator not engaged/ Collaboration channels	Make role very clear; model; possibly switch partners; collaboration rubric
Persevere; frustration from failure/not getting it	• Show failures; show your failures • Celebrate what we learn from "failures" • "Found a bug" visual • Don't make a competition
Not enough robots	Code.org puzzles Unplugged centers Pair programming Borrow materials from STEM+CS
Puzzlets and students with low vision	Brailing puzzlets

Data includes teaching artifacts

One example: Teaching Shapes with Dash Robot



Study 2: Instructional Coaching Study (Collaboration with Cornell Tech)

Purpose: Investigate teachers' comfort in teaching CS to students with disabilities & how CS coaches provide support to teachers in this effort.

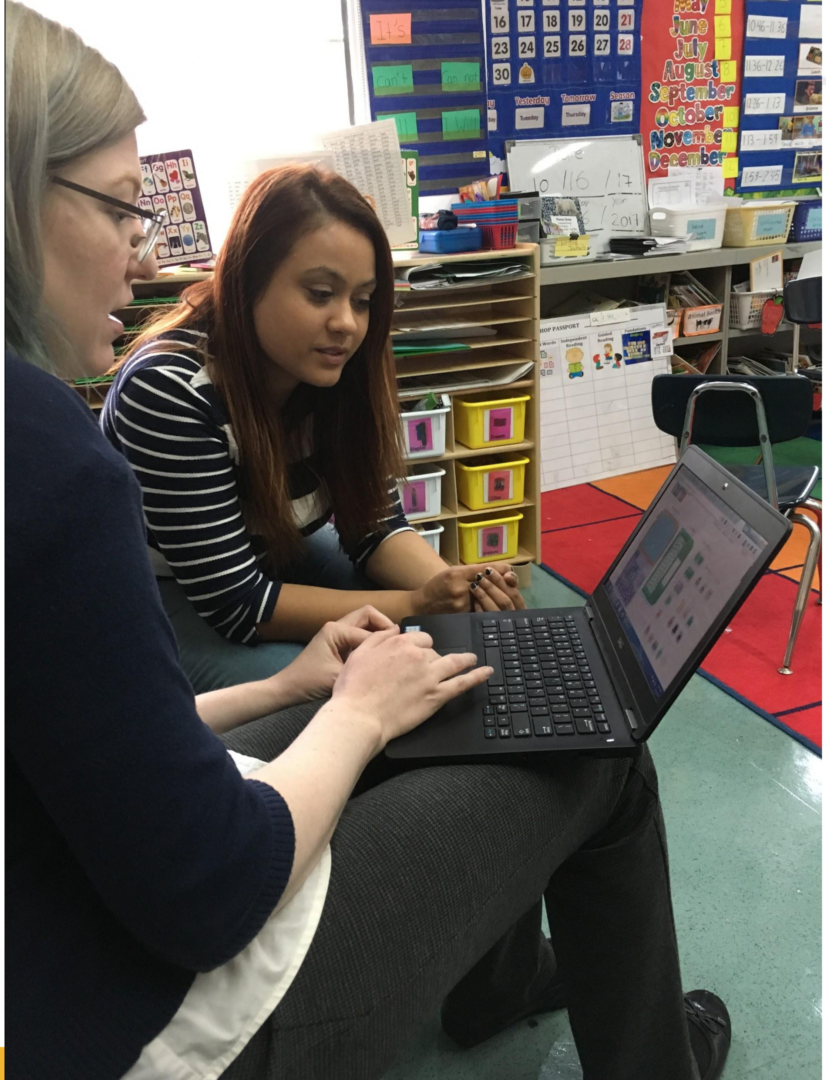
Research questions:

1. How confident are teachers in supporting students with disabilities in CS education?
2. To what extent does instructional coaching influence teacher confidence in teaching CS to students with disabilities?
3. How familiar are teachers in using UDL during CS education?
4. How are CS instructional coaches addressing inclusion and UDL in their coaching practices?

Paper under review

Participants

- 66 teachers from two K- 6 schools
  - 37 general education
  - 16 special education
  - 6 school-based coaches
  - 3 gifted and talented
  - 4 English as a second language
- 4 Computer Science Instructional Coaches



Teacher in Residence (TIR)

Content Coaching

- Systems-based approach
- Co-planning
- Observation/co-teaching
- Reflection & Feedback
- Supplemental PD

Research Design

- Quantitative analysis of pre/post teacher surveys
- Qualitative analysis of open-ended post-survey items
- Qualitative analysis of focus group with CS coachers

Quantitative Survey Results

Paired Samples t-test Results:

Table 1. Survey Likert Scale Means and t-values

Item	Pretest	Posttest	t-value
Confidence in teaching CS	M=3.51, SD=.85	M=3.86, SD=.788	t=2.97, p=.004
Confidence in teaching CS to students with disabilities	M=3.25, SD=1.22	M=3.5, SD=.93	t=2.16, p=.035
Looking forward to teaching CS	M=4.06; SD=.86	M=4.11, SD=.78	t=.344, p=.732
Understanding how to apply UDL in the context of CS education	M=1.41, SD=.95	M=1.56, SD=1.03	t=1.196, p=.236

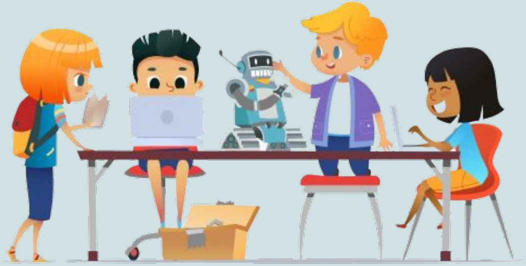
Multiple linear regression: $F(3, 57) = 11.54$, $p < .001$ with an $R^2 = 0.35$.

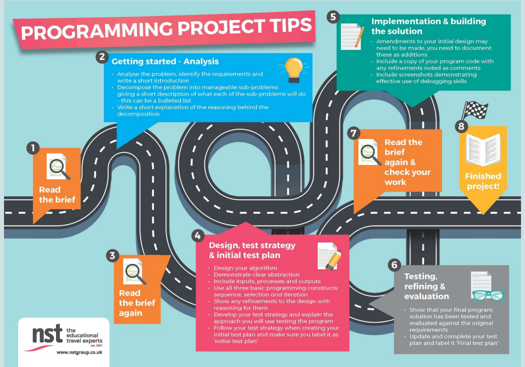
Confidence in teaching CS to students with disabilities based on the number of years the teachers taught CS, confidence in teaching CS in general, and understanding of UDL in the context of CS

CREATIVE TECHNOLOGY RESEARCH LAB

Survey Questions

What strategies support students with disabilities in CS instruction?





Exploratory, hands-on learning

Visual supports

Step-by-step instructions

Qualitative Results - Coach Focus Group

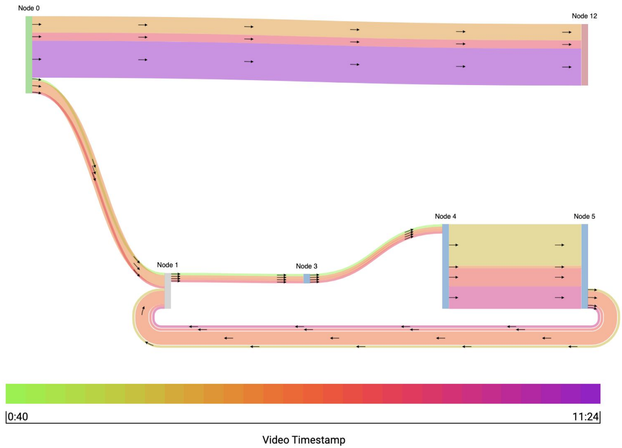
Equity

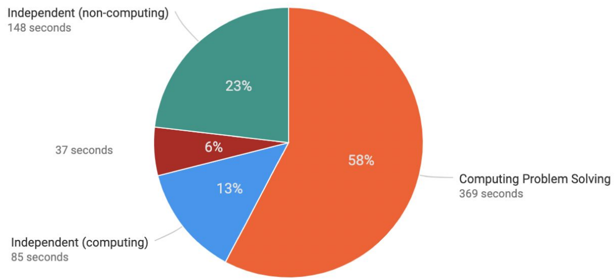
Teacher Agency

Universal Design for Learning

Student Performance through Video Analysis

- Collaborative Computing Observation Instrument (CCOI)
- Goal: Track student challenges and strategies
- Codes for
  - Independent/Collaborative problem solving
  - Challenges/barriers
  - Computational discussions
  - Non-computing behaviors
- https://ccoi.education.ufl.edu/





CREATIVE TECHNOLOGY RESEARCH LAB

Studying Student Engagement and Learning

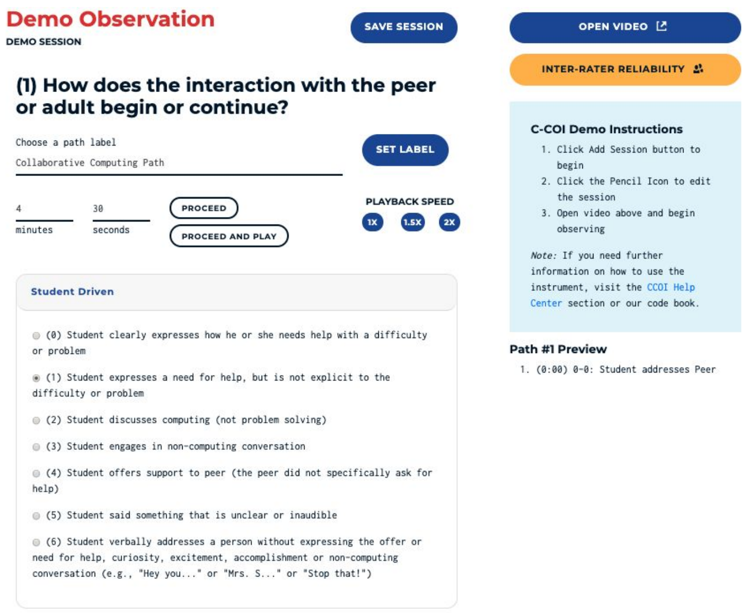
<u>Data includes:</u>

-Videos of students' computational behaviors

-Student observations

-Addition of data analytics (sequential data mining, Hidden Markov Modeling)

-Cognitive interviews/think-aloud protocols



CREATIVE TECHNOLOGY RESEARCH LAB

So how can we measure student participation?

-Interactions with peers

-Persistence on difficult tasks

-Adaptive help seeking

-Difficulties faced

-Successes and failure in completing CS activities

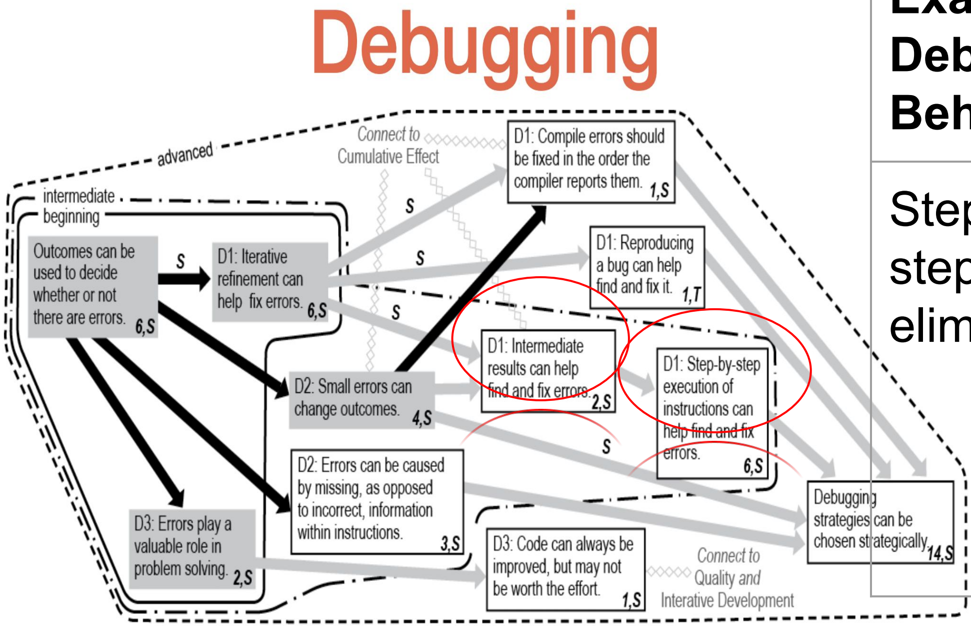
-Metacognitive strategy use

Study 3: Examination of Debugging

- Video of students engaged in computational tasks.
- Code for challenges and debugging strategies
- Most common debugging approaches:
  - Trial and error
  - Step-by-step testing/error elimination
  - Abandoning code and starting over
- Most common “other” strategies
  - Using system-embedded supports (e.g., tutorials)
  - Help seeking from peers and adults

Yan et al., (in preparation)

Mapping Debugging to Learning Trajectories



Example Debugging Behavior	Example Consensus Goals in the Learning Trajectory
Step-by-step error elimination	-Intermediate results can find and fix errors -Step-by-step execution of instructions can help fix and find bugs.

Yan et al., (in preparation)

Study 4: Samples of Human Labeled Sequences Related to Problem-Solving Phases

Trial & Error

Event	Object_Attribute_Selected
SELECT_ICE_CREAM_TOPPING	Ice Cream: Whip
SELECT_PIZZA_TOPPING	Pizza: Pineapple
DELIVER_PIZZA_ICE_CREAM	.
TROLL_WANTS_MORE	.
TROLL_DISLIKES_SOMETHING	.
TROLL_REJECTS_DELIVERY	.
SELECT_ICE_CREAM_TOPPING	Ice Cream: Sprinkles
SELECT_PIZZA_TOPPING	Pizza: Pineapple
SELECT_PIZZA_TOPPING	Pizza: Pepper
SELECT_ICE_CREAM_TOPPING	Ice Cream: Whip
DELIVER_PIZZA_ICE_CREAM	.
TROLL_DISLIKES_SOMETHING	.
TROLL_DISLIKES_SOMETHING	.
SELECT_PIZZA_TOPPING	Pizza: Pepperoni
SELECT_PIZZA_TOPPING	Pizza: Pineapple
TROLL_REJECTS_DELIVERY	.

Systematic Testing

Event	Object_Attribute_Selected
SELECT_PIZZA_TOPPING	Pizza: Mushroom
DELIVER_PIZZA_ICE_CREAM	.
TROLL_DISLIKES_SOMETHING	.
TROLL_REJECTS_DELIVERY	.
SELECT_ICE_CREAM_TOPPING	Ice Cream: Whip
DELIVER_PIZZA_ICE_CREAM	.
TROLL_DISLIKES_SOMETHING	.
TROLL_REJECTS_DELIVERY	.
SELECT_PIZZA_TOPPING	Pizza: Pepper
DELIVER_PIZZA_ICE_CREAM	.
TROLL_DISLIKES_SOMETHING	.
TROLL_REJECTS_DELIVERY	.

Event	Object_Attribute_Selected
SELECT_PIZZA_TOPPING	Pizza: Mushroom
SELECT_PIZZA_TOPPING	Pizza: Pepperoni
SELECT_PIZZA_TOPPING	Pizza: Pepper
SELECT_PIZZA_TOPPING	Pizza: Pineapple
SELECT_PIZZA_TOPPING	Pizza: Cheese
REMOVE_PIZZA_TOPPING	Pizza: Cheese
DELIVER_PIZZA_ICE_CREAM	.
TROLL_DISLIKES_SOMETHING	.
TROLL_REJECTS_DELIVERY	.
SELECT_PIZZA_TOPPING	Pizza: Mushroom
SELECT_PIZZA_TOPPING	Pizza: Pepperoni
SELECT_PIZZA_TOPPING	Pizza: Pepper
SELECT_PIZZA_TOPPING	Pizza: Cheese
DELIVER_PIZZA_ICE_CREAM	.
TROLL_DISLIKES_SOMETHING	.

Event	Object_Attribute_Selected
SELECT_PIZZA_TOPPING	Pizza: Pepper
SELECT_PIZZA_TOPPING	Pizza: Mushroom
DELIVER_PIZZA_ICE_CREAM	.
TROLL_WANTS_MORE	.
~~TROLL_REJECTS_DELIVERY~~	
SELECT_PIZZA_TOPPING	Pizza: Cheese
SELECT_PIZZA_TOPPING	Pizza: Mushroom
SELECT_PIZZA_TOPPING	Pizza: Pepperoni
DELIVER_PIZZA_ICE_CREAM	.
TROLL_DISLIKES_SOMETHING	.

Systematic Testing with Partial Solution

Event	Object_Attribute_Selected
SELECT_PIZZA_TOPPING	Pizza: Pepper
SELECT_PIZZA_TOPPING	Pizza: Pepperoni
SELECT_PIZZA_TOPPING	Pizza: Cheese
SELECT_PIZZA_TOPPING	Pizza: Mushroom
DELIVER_PIZZA_ICE_CREAM	.
TROLL_DISLIKES_SOMETHING	.

Full Solution

Event	Object_Attribute_Selected
SELECT_PIZZA_TOPPING	Pizza: Cheese
SELECT_PIZZA_TOPPING	Pizza: Pepperoni
SELECT_PIZZA_TOPPING	Pizza: Pepper
DELIVER_PIZZA_ICE_CREAM	.
TROLL_ACCEPTS	.

Test one at a time

Winnowing

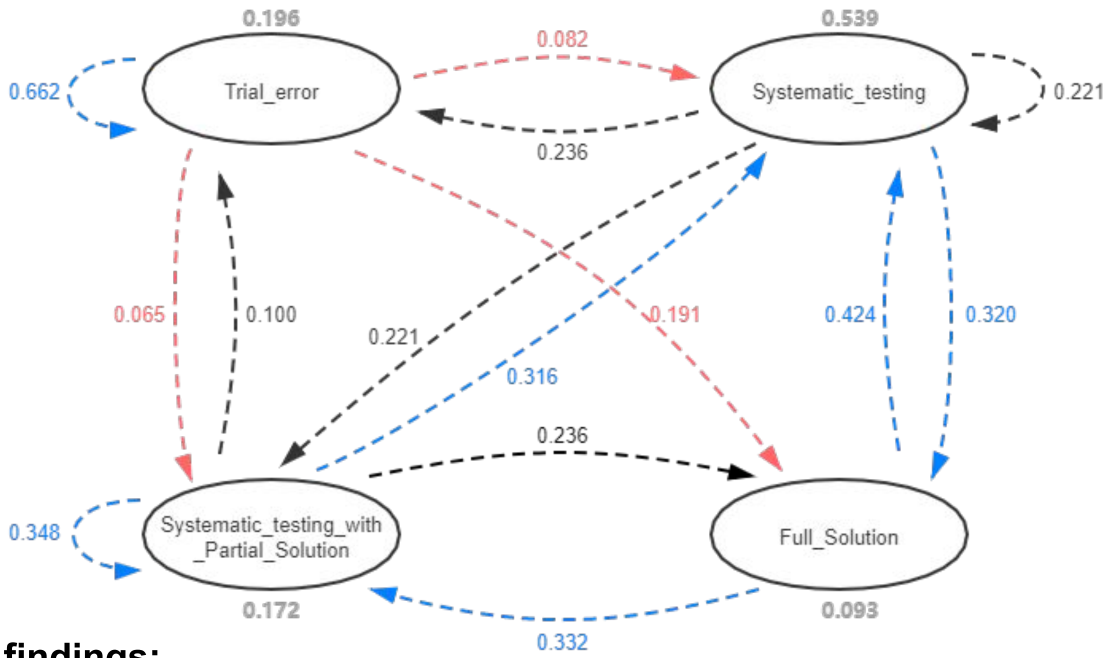
Additive (most used, 59.10%)

Three strategies

Circled Toppings' attributes are correct

Liu et al. (in preparation);
U.S. Dept of Education, EIR; INFACT Project (PI: TERC)

Probability of Transitions among Different Phases in Problem-Solving Progress



Gray: probability as initial states

Red: lower probability

Blue: higher probability

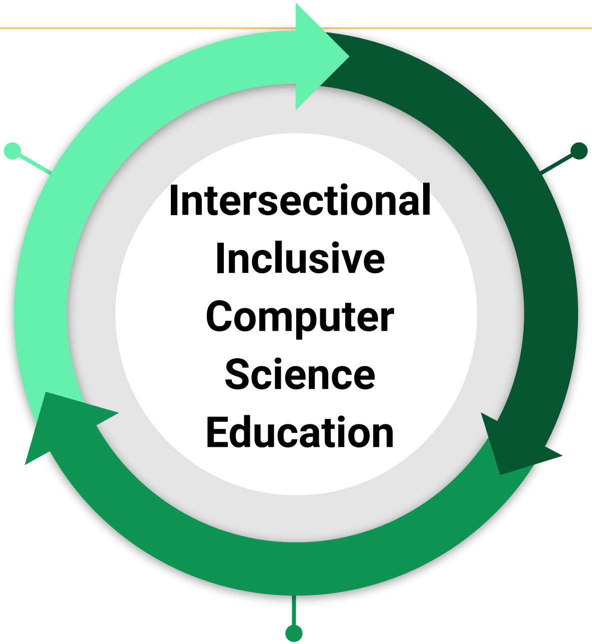
Raw findings:

- Systematic_testing has highest probability as an initial phase (0.539).
- Trial_error → Trial_error has highest probability of transition (0.662).
- If student get a full_solution in previous puzzles, always start with Systematic_testing phase in next puzzle (0.424).

What's Next?

Universal Design for Learning

Culturally Responsive and Sustaining Pedagogy



Translanguaging/ multilingual education

CREATIVE TECHNOLOGY RESEARCH LAB

[CSTA Inclusive Teaching Practices]

Key Takeaways

1. Please include students with disabilities in K-12 CS education. They WILL succeed when given accessible, engaging activities.
2. Consider goals, anticipated barriers, and the UDL principles when designing instruction for all learners.
3. Disaggregate your data to see who is meeting instructional goals and who is not.
4. Share successes of students with disabilities in CS education so we can start shifting the discourse to better inclusion.

Thank you so much!