Making Bugs for Learning

How youth can benefit from designing (and fixing) buggy projects with programmable microcontrollers

Luis Morales-Navarro, University of Pennsylvania
Deborah A. Fields, Utah State University
Yasmin B. Kafai, University of Pennsylvania

Yasmin Mia Gayithri Luis

Debbie Michael Justice

+ amazing teachers and youth!

Mike

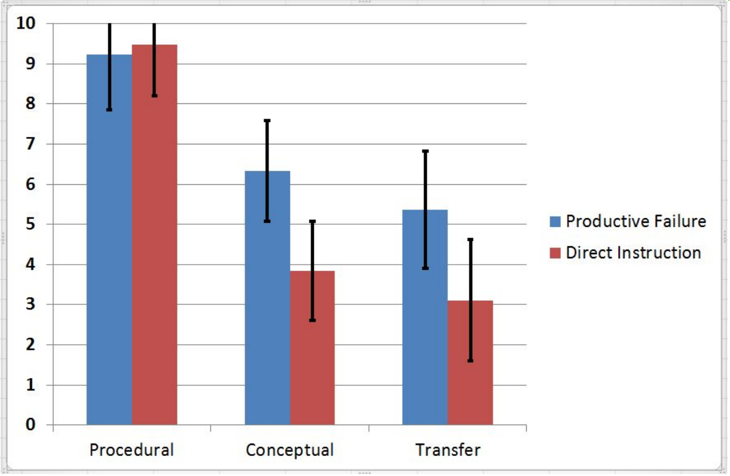
A team effort/project!

"You have to learn how to handle failure. But you can learn from that."

Katalin Karikó

"Making mistakes is beautiful, mistakes are pleasures."

Bad Bunny




SCIENTIFIC AMERICAN.

Sign in

SHAR

COGNITION

How to Turn Failure into Success

Research reveals strategies for staying motivated in the face of challenges

By Rachel Nuwer on April 1, 2019

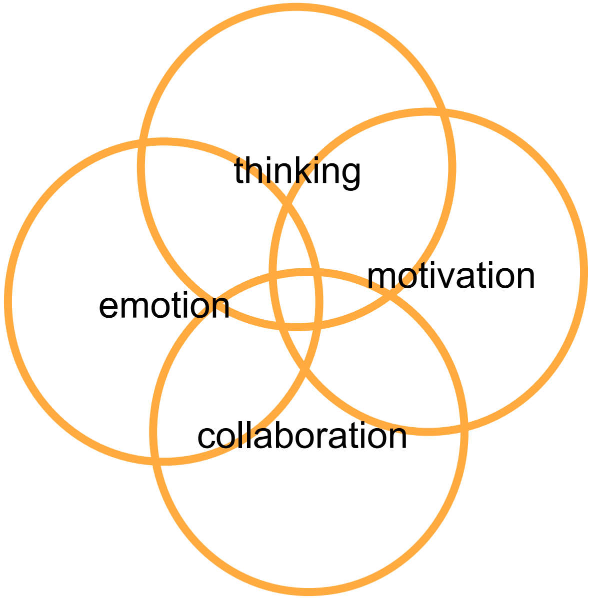
PRODUCTIVE FAILURE


I love a good challenge!
I wonder how I can fix this bug! Maybe if I try...
I keep making mistakes. I'll never be good at coding!
I'm just not smart enough for CS, I'm not taking it next year.

Traditional approaches to debugging:

- Checklists [1]
- Strategies [2]
- Problem sets [1]

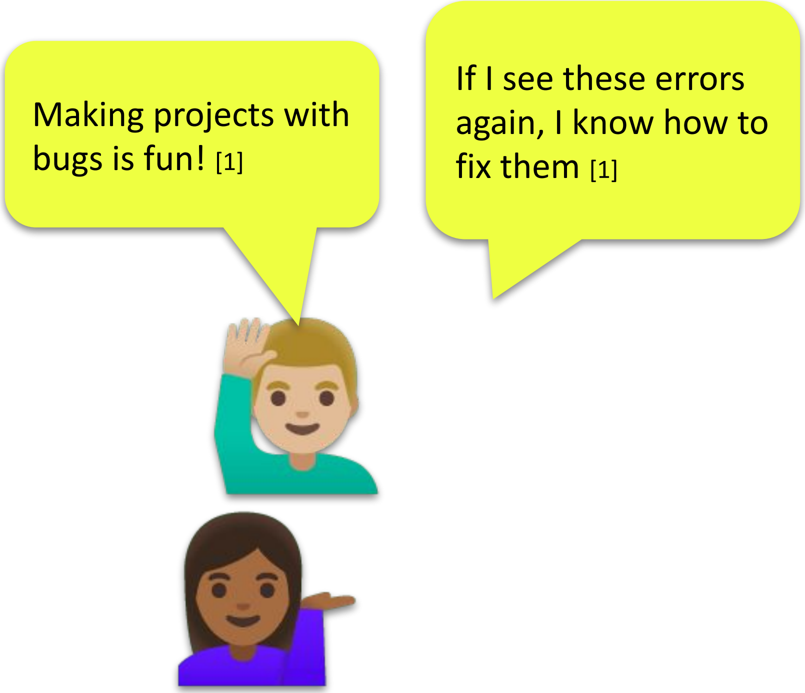
Debugging as a holistic process [3]:



[1] McCauley et al., 2008.
[2] Silva, 2011.
[3] Dahn & DeLiema, 2020.

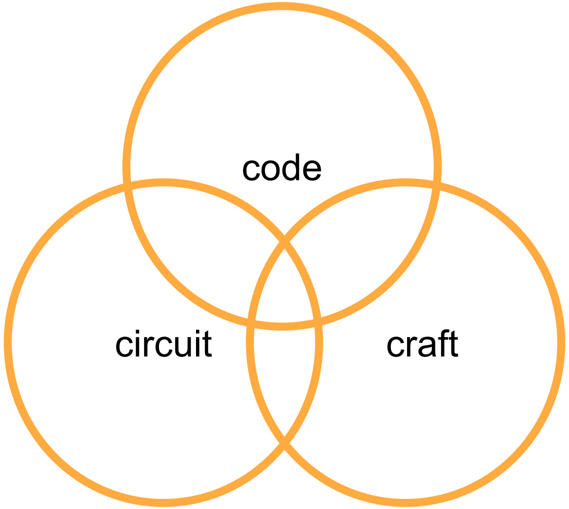
Debugging by Design:

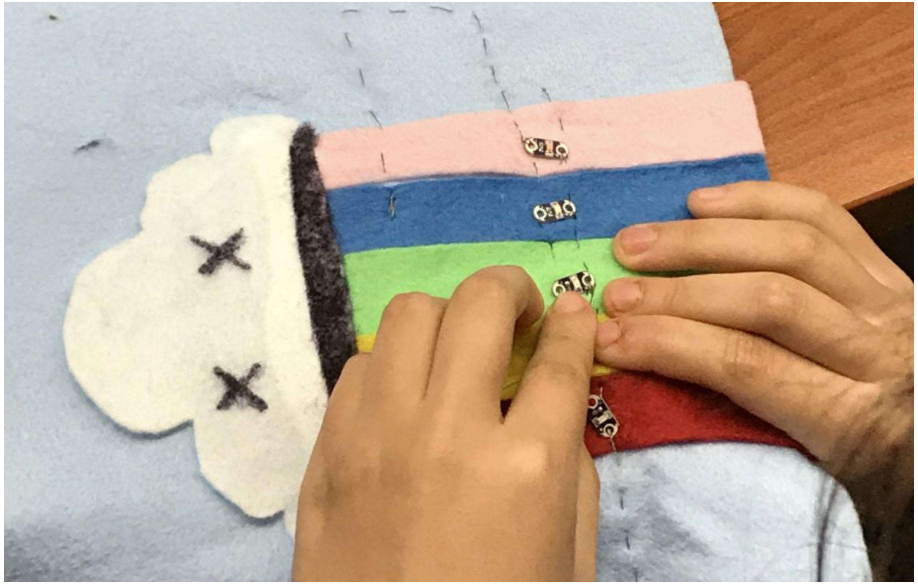
Having students design **buggy** (rather than functional) projects for their peers to solve



[1] Fields, Kafai, Morales-Navarro, & Walker, 2021.

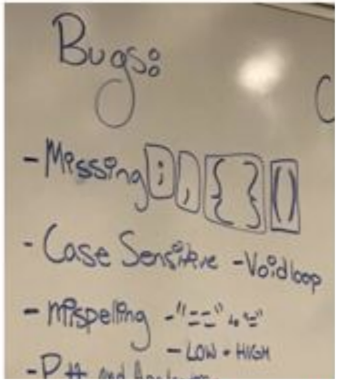
E-textiles/Physical Computing



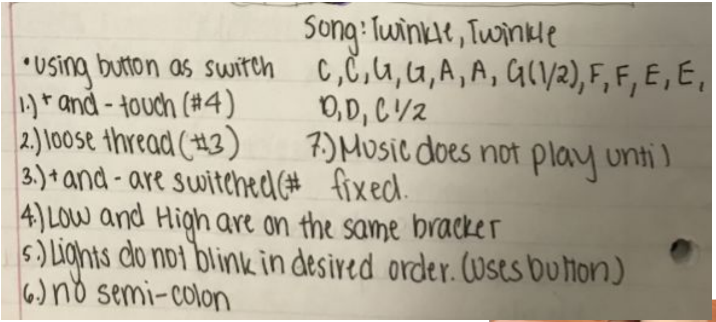


Personally relevant/creative projects!

Debugging by Design



1. Hall of Bugs



2. Planning buggy projects

3. Making buggy projects

4. Solving buggy projects

Reflection

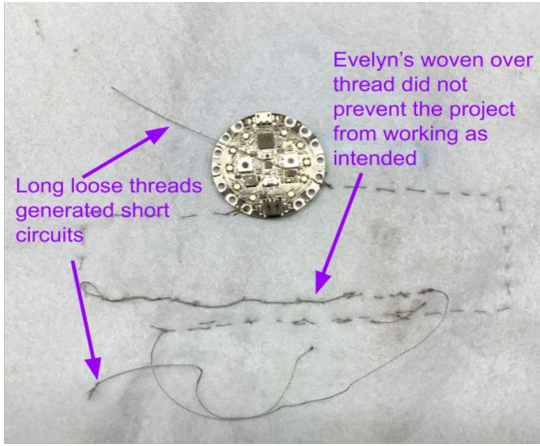
Fields, D. A., Kafai, Y. B., Morales-Navarro, L., & Walker, J. T. (2021). Debugging by design: A constructionist approach to high school students' crafting and coding of electronic textiles as failure artefacts. British Journal of Educational Technology, 52(3), 1078-1092.

Context

Exploring Computer Science

- Spring in 2019 - 1 classroom (25 students)
- Spring 2020 - 10 classrooms cancelled due to COVID
- Spring 2021 - 11 **online** classrooms (8 DbD n=158, 3 E-textiles n=93)
- Implementing Electronic Textiles unit within *Exploring Computer Science*
- Eight 50 minute long class sessions

E-Textiles Buggy Project:



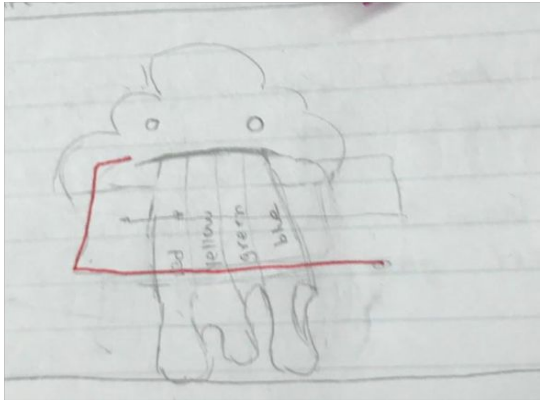

```
int rainbow = 5;
int yellow = 7;
int green = 8;
int blue = 9;
int pink = 10;
int button1 = 4;

void setup()
{
    pinMode(rainbow, OUTPUT);
    pinMode(button2, INPUT)
    pinMode(button1, INPUT);
}

void loop()
{
```

"rainbow" should be assigned to pin 6

Nuisance variables

"button1" should be assigned to pin 19

Unnecessary code: "button2" is not used in the project and it is not declared



Sick Cloud Throwing up a Rainbow by Evelyn & Nicolás

What types of bugs did students design?

Morales-Navarro, L., Kafai, Y. B., Brennan, K., Haduong, P., Venkatasubramanian, V., Hennig, H., Michaeli, T., Weintrop, D., Tsan, J., Franklin, D., Jimenez, Y., Gardner-McCune, C., Hennessy Elliott, C., Schneider, M., Bush, J. B., Fields, D. A., Recker, M., Nixon, J., Castro, F., DesPortes, K., Tissenbaum, M., Smith, C., Bawankule, A., Hopping, D., Holbert, N., Correa, I., Danzig, B., Zikovitz, D., Blikstein, P., Berland, M. (2023). Designing for Successful Failures: Constructionist Perspectives on Supporting Personally Meaningful and Culturally Empowered Learning and Teaching. In Proceedings of Fablearn/Constructionism 2023.

circuit and design bugs

Circuity and design bugs		Examples	Instances	Groups
Connections	Short circuits or loose electrical connections	Leaving long loose threads that create a short when they flop onto other conductive thread connections, Crossing a positive and negative line	20	11
	Open circuit	Leaving a gap in circuitry, for instance with one connection unsewn	1	1
	Connection on the artifact not matching pin number declared *Overlaps with coding errors*	`int lid = 12;` where "lid" LED actually connects to pin 10	11	5
	Reversed polarity issues	Connecting a line from a positive pin to the negative side of an LED, Flipping an LED to be misaligned	13	9
Design	Adding unneeded components	Using non conductive threads, Using glue in a way that made deconstruction difficult }	3	2

semantic bugs

Semantic bugs		Examples	Instances	Groups

Issues with sequence of code	Missing or misplaced delay, or wrong order of functions	void flash(){
		digitalWrite(Pin12,HIGH);
		digitalWrite(Pin6,HIGH);
		//missing a delay here
		digitalWrite(Pin12,LOW);
		digitalWrite(Pin6,LOW);
		delay(100);}
			8	5

Issues with logical expressions	Reverse conditions	if(butt1Val == LOW && butt2Val == HIGH)	1	1
	Redundant logical expressions	if(leverVal == HIGH && leverVal == HIGH)	2	2
	Contradictory logical expressions	if(leverVal == HIGH && leverVal == LOW)	1	1

	Gap or overlap in conditions	if(brightness >900){
		light10();
		}else if(brightness >750 && brightness <= 950){
		light9();}
			1	1

Issues with built-in function calls	Using INPUT instead of OUTPUT or vice versa in pinMode() parameters	pinMode(button1, OUTPUT);	13	6
	Confusing digitalWrite(), analogWrite(), digitalRead(), or analogRead()	int butt1Val = digitalWrite(button4);	20	3

	Use pinMode() more than once on the same pin	void setup(){
		pinMode(blue, OUTPUT);
		pinmode(lever,INPUT);
		pinMode(pink, OUTPUT;
		pinMode(lever,INPUT);
		}
			1	1

	Out of range value in analogWrite	for (int i = 256; i >0; i = i-15){
		analogWrite(press, i);
		}
			1	1

syntax bugs

Syntax bugs		Examples	Instances	Groups
Syntax issues in variable use and declaration	Spaces in variable names	`Light 1 = 3;`	4	1

	Typos or mismatching variable names	`int rainbow = 5;`
		`digitalWrite(ranbow, HIGH);`
		`digitalWrite(rinbow, HIGH);`
			13	6

	Variables declared without data type	`Light 1 = 3;` `//missing “int”`		1
	Uses pin number instead of variable (without reading the pin)	`if( 1 == HIGH && butt2Val == HIGH)`	4	1
	Using undeclared variables	`pinMode(beep, OUTPUT);` `//beep is undefined`	3	2
Syntax issues in student defined functions	Calling undefined functions	`hi() //without hi() being defined`	1	1

	Typos in when calling function	`capture();`
		`//...`
		`void captre() { }`
			1	1

	Missing or extra () when calling functions	`free));`	2	2
Other syntax issues	Missing ;	`int butt1Val = digitalRead(button1)`	29	7

	Missing { or }	`void loop()`
		`//...`
		`}`
			10	2

	Extra ;	`void setup();`	2	1
	Case-sensitivity of constants or built-in functions	`else if(butt1Val == HIGH && butt2Val == low)`	12	3

What growth mindset practices did students demonstrate while engaging in DbD?

Morales-Navarro, L., Fields, D. A., & Kafai, Y. B. (2024). Understanding growth mindset practices in an introductory physical computing classroom: high school students' engagement with debugging by design activities. Computer Science Education, 1-31.

What growth mindset practices did students demonstrate while engaging in DbD?

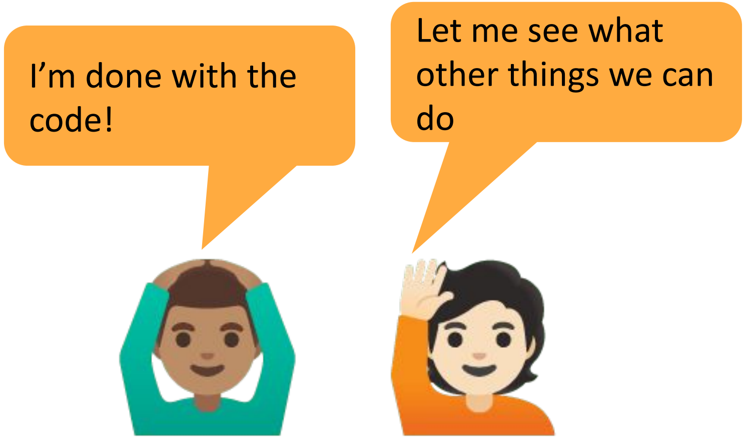
Choosing challenges that lead to more learning

Persisting after setbacks

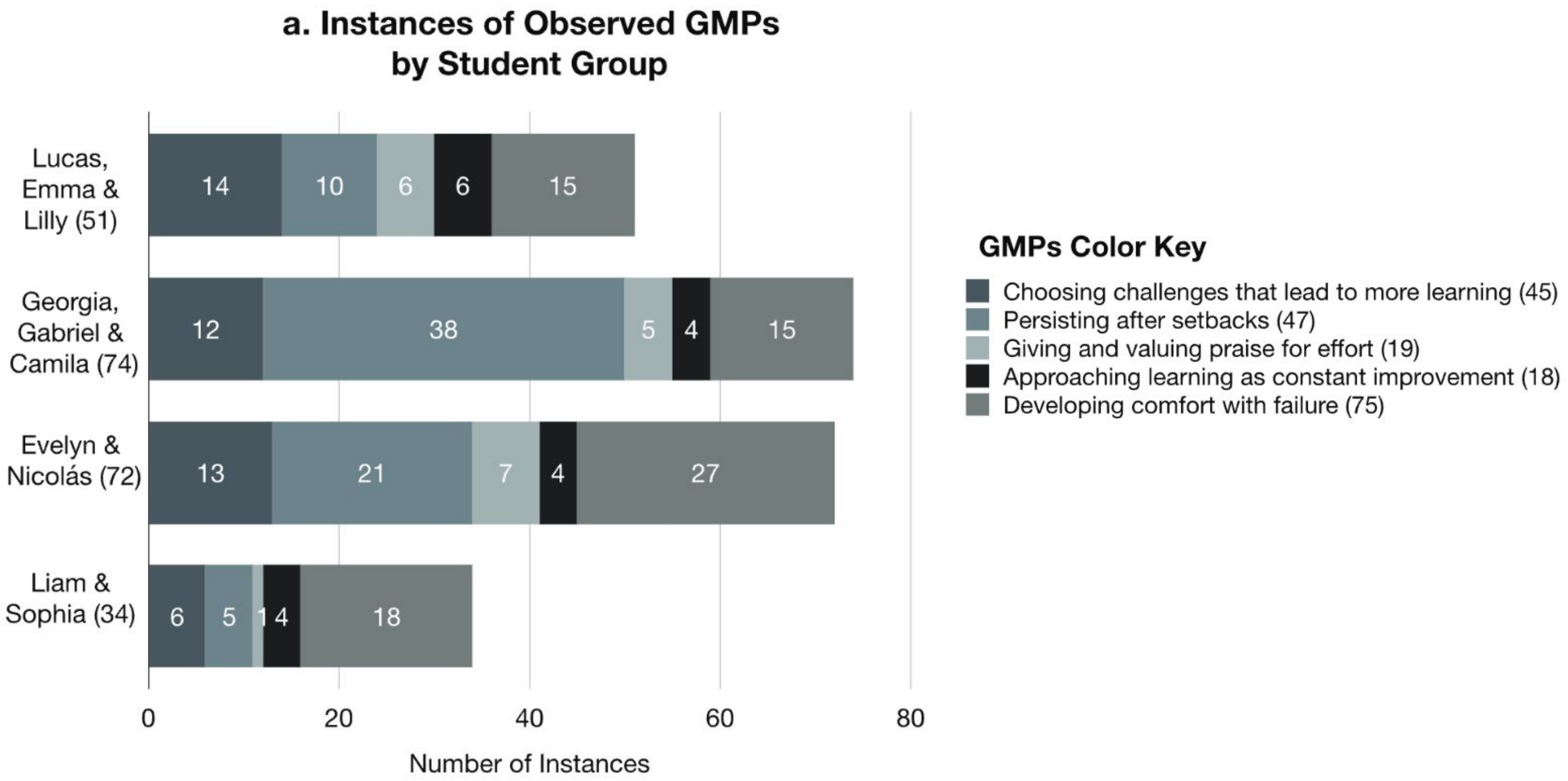
Giving and valuing praise for effort

Approaching learning as constant improvement

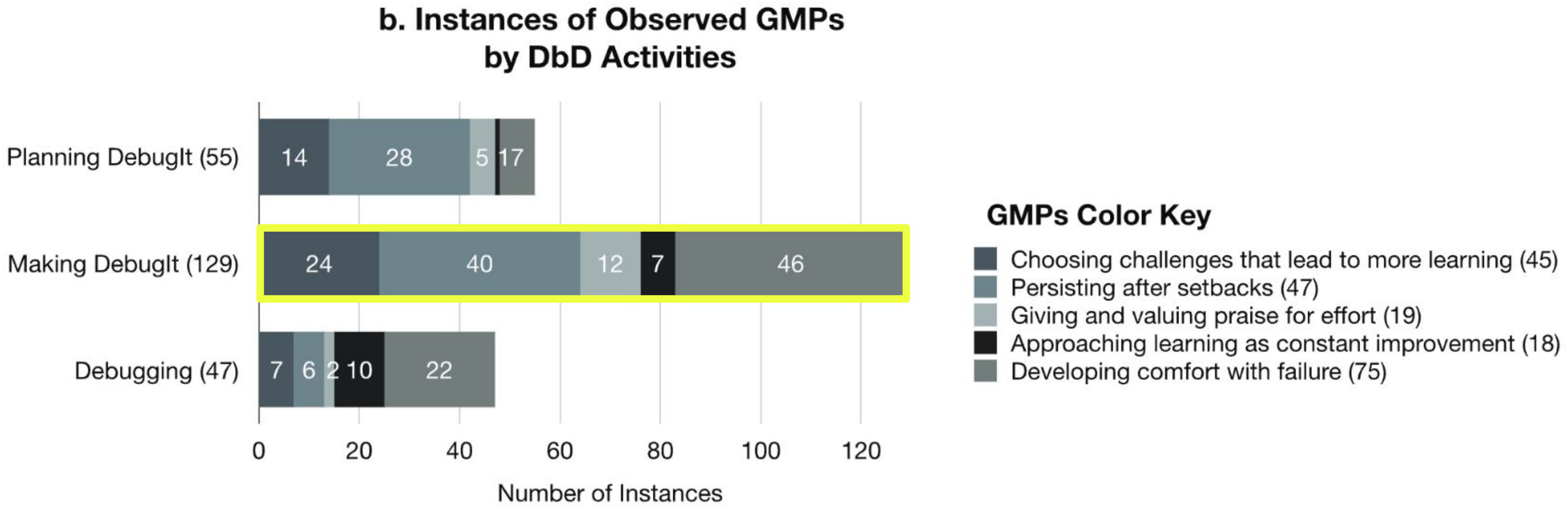
Developing comfort with failure



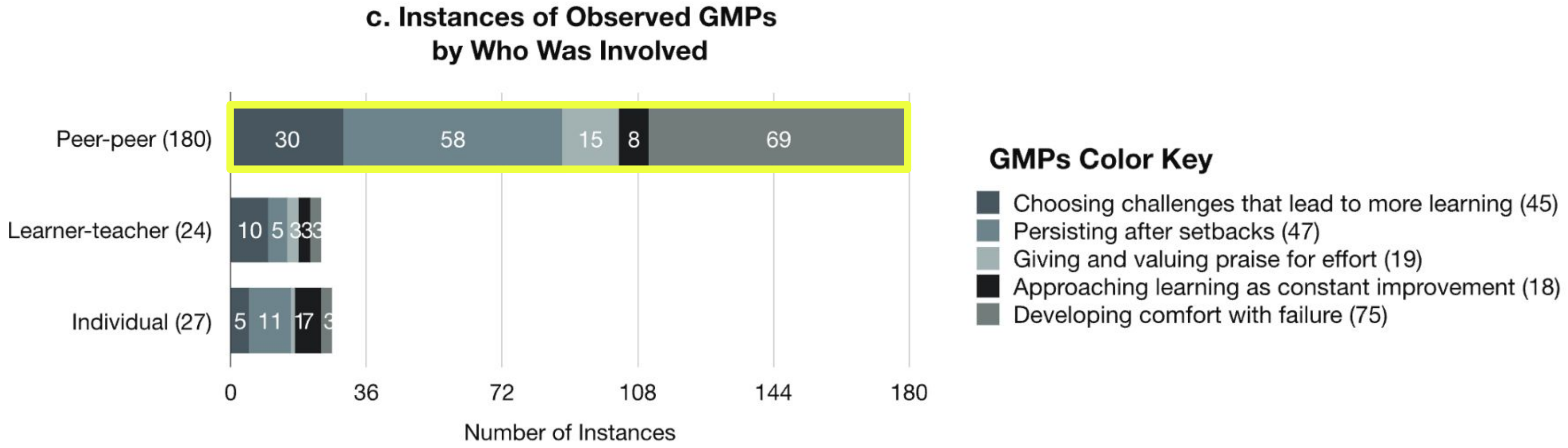
Growth Mindset Practices in Debugging by Design



Growth Mindset Practices in Debugging by Design



Growth Mindset Practices in Debugging by Design



What are the effects of DbD on Students' Self-Beliefs in Computing?

Morales-Navarro, L., Giang, M., Fields, D. A., & Kafai, Y. B. (2023). Connecting Beliefs, Mindsets, Anxiety, and Self-Efficacy in Computer Science Learning: An Instrument for Capturing Secondary School Students' Self-Beliefs. Computer Science Education. https://doi.org/10.1080/08993408.2023.2201548

Morales-Navarro, L., Fields, D.A., Giang. M., & Kafai, Y. B. (2023). Designing Bugs or Doing Another Project: Effects on Secondary Students' Self-Beliefs in Computer Science. In Blikstein, P., Van Aalst, A., Kizito, R., & Brennan, K. (Eds.). Proceedings of the 17th International Conference of the Learning Sciences - ICLS 2023. Montréal, Canada: International Society of the Learning Sciences.

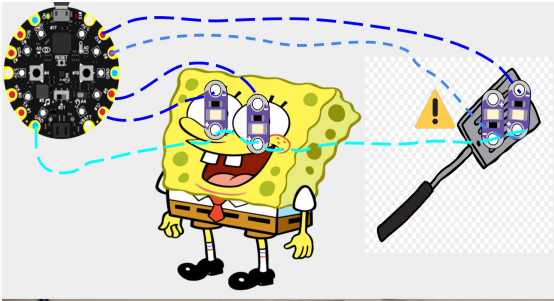
DbD: E-Textiles Buggy Projects (n = 158)

Hall of Problems: Discussion of errors and mistakes

Debug-it design: 5 bugs in code and 1 bug in circuit diagram. Provided a project statement, circuit diagram, and code.

Project construction: Built and solved each others projects.

```
int eye1 = 12;
int eye2 = 6;
int spatula1 = 9;
int spatula2 =10;
//Name 2 switches/buttons
int button1 = 4;
int button2 = 19;

void setup(){
   pinMode(eye1, OUTPUT);
   pinMode(eye2, OUTPUT);
   pinMode(spatula1,OUTPUT);
   pinMode(spatula2,INPUT)

   pinMode(button1,INPUT);
   pinMode(button2,OUTPUT)

}
```



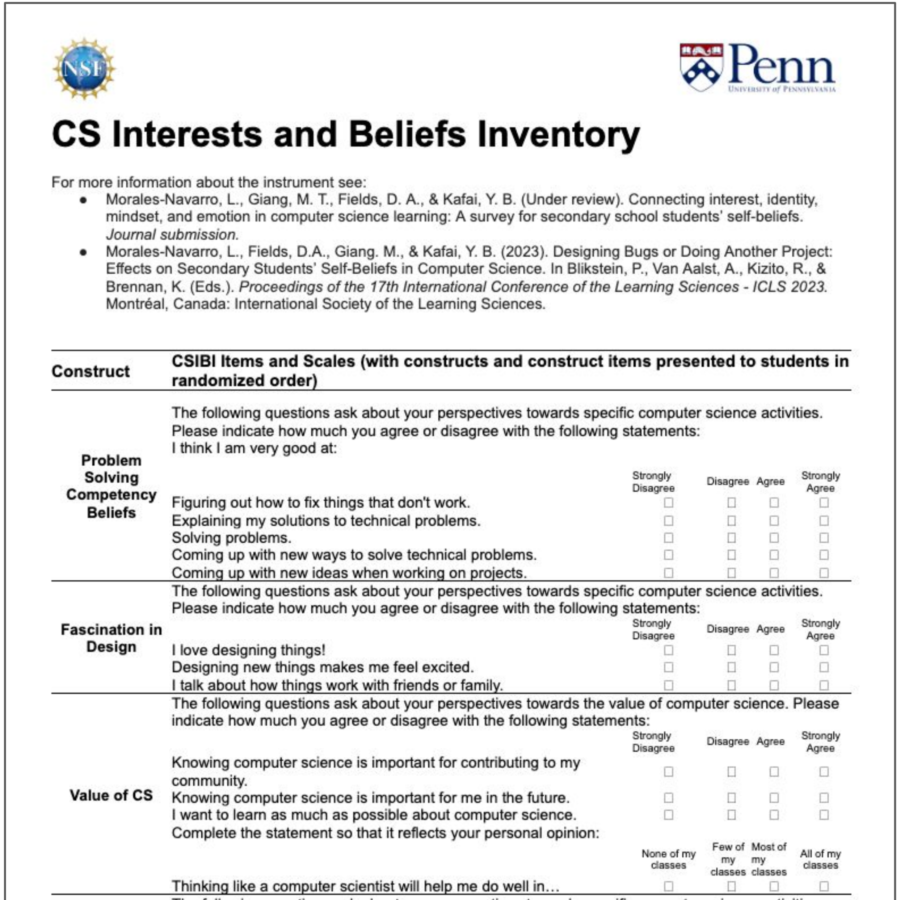
Comparison: Music projects (n=93)

Students made a project with programmed music (by coding tones and rhythms, using arrays, for loops, and conditionals).

```
int speaker = 5;
int pace = 2200;
float MyDurations[] = {3.0/8.0, 3.0/8.0, 1.0/4.0, 1.0/8.0, 3.0/
1.0/4.0, 1.0/8.0, 3.0/4.0};
int MyPitches[] = {294, 294, 294, 330, 370, 370, 330, 370, 392,

void setup(){
pinMode(speaker, OUTPUT);
}

void loop(){
 for (int i = 0; i < 10; i = i+1) {
tone (speaker, MyPitches[i]);
delay(pace * MyDurations[i]);
notone(speaker);
delay(10);
 }
}
```

Data collection

- Project completion variable ("Didn't do it" (1) to "Finished" (5))
- CS self-beliefs constructs four point likert scale (factor loadings average of .742 and range between .491 to .875; reliabilities between .713 and .889)
  - Problem solving competency beliefs
  - Fascination in design
  - Value of CS
  - CS creative expression
  - E-Textiles coding self-efficacy
  - Programming fixed mindset
  - Programming growth mindset
  - Programming anxiety
  - Programming self-concept



CS Interests and Beliefs Inventory

For more information about the instrument see:

- Morales-Navarro, L., Giang, M. T., Fields, D. A., & Kafai, Y. B. (Under review). Connecting interest, identity, mindset, and emotion in computer science learning: A survey for secondary school students' self-beliefs. *Journal submission.*
- Morales-Navarro, L., Fields, D.A., Giang. M., & Kafai, Y. B. (2023). Designing Bugs or Doing Another Project: Effects on Secondary Students' Self-Beliefs in Computer Science. In Blikstein, P., Van Aalst, A., Kizito, R., & Brennan, K. (Eds.). *Proceedings of the 17th International Conference of the Learning Sciences - ICLS 2023.* Montréal, Canada: International Society of the Learning Sciences.

Construct	CSIBI Items and Scales (with constructs and construct items presented to students in randomized order)				
Problem Solving Competency Beliefs	The following questions ask about your perspectives towards specific computer science activities. Please indicate how much you agree or disagree with the following statements: I think I am very good at:	Strongly Disagree	Disagree	Agree	Strongly Agree
	Figuring out how to fix things that don't work.	☐	☐	☐	☐
	Explaining my solutions to technical problems.	☐	☐	☐	☐
	Solving problems.	☐	☐	☐	☐
	Coming up with new ways to solve technical problems.	☐	☐	☐	☐
	Coming up with new ideas when working on projects.	☐	☐	☐	☐
Fascination in Design	The following questions ask about your perspectives towards specific computer science activities. Please indicate how much you agree or disagree with the following statements:	Strongly Disagree	Disagree	Agree	Strongly Agree
	I love designing things!	☐	☐	☐	☐
	Designing new things makes me feel excited.	☐	☐	☐	☐
	I talk about how things work with friends or family.	☐	☐	☐	☐
Value of CS	The following questions ask about your perspectives towards the value of computer science. Please indicate how much you agree or disagree with the following statements:	Strongly Disagree	Disagree	Agree	Strongly Agree
	Knowing computer science is important for contributing to my community.	☐	☐	☐	☐
	Knowing computer science is important for me in the future.	☐	☐	☐	☐
	I want to learn as much as possible about computer science.	☐	☐	☐	☐
	Complete the statement so that it reflects your personal opinion:	None of my classes	Few of my classes	Most of my classes	All of my classes
	Thinking like a computer scientist will help me do well in…	☐	☐	☐	☐

Did participation in the DbD/comparison (Music) activity impact students' project completion?

No significant difference between DbD and comparison in terms of project completion

DbD ($M = 1.537$, SD 2.56)

Comparison ($M = 2.56$, $SD = 1.550$)

ANOVA: $F(1, 142) = 0.000$, $p = 1$, partial $\eta2 = 0$

No difference by gender or teacher

Did completing either class project relate to students’ CS self-beliefs and was this influenced by whether students were in the DbD or the comparison (Music) activities?

Construct	DbD	Comparison
Problem solving competency beliefs	.191	.364**
Fascination in design	.257*	.229
Value of CS	.201	.266
CS creative expression	.284**	.363**
E-Textiles coding self-efficacy	.321**	.335*
Programming fixed mindset	-.032	-.178
Programming growth mindset	.313**	.230+
Programming anxiety	.042	-.330*
Programming self-concept	.190	.297*

+p < .10, *p < .05, **p < .01

Did completing either class project relate to students' CS self-beliefs and was this influenced by whether students were in the DbD or the comparison (Music) activities?

Construct	DbD	Comparison
Problem solving competency beliefs	.191	.364**
Fascination in design	.257*	.229
Value of CS	.201	.266
CS creative expression	.284**	.363**
E-Textiles coding self-efficacy	.321**	.335*
Programming fixed mindset	-.032	-.178
Programming growth mindset	.313**	.230+
Programming anxiety	.042	-.330*
Programming self-concept	.190	.297*

Significant for both DbD & comparison

+p < .10, *p < .05, **p < .01

Did completing either class project relate to students’ CS self-beliefs and was this influenced by whether students were in the DbD or the comparison (Music) activities?

Significant for DbD

Construct	DbD	Comparison
Problem solving competency beliefs	.191	.364**
Fascination in design	.257*	.229
Value of CS	.201	.266
CS creative expression	.284**	.363**
E-Textiles coding self-efficacy	.321**	.335*
Programming fixed mindset	-.032	-.178
Programming growth mindset	.313**	.230+
Programming anxiety	.042	-.330*
Programming self-concept	.190	.297*

+p < .10, *p < .05, **p < .01

Did completing either class project relate to students’ CS self-beliefs and was this influenced by whether students were in the DbD or the comparison (Music) activities?

Significant for comparison (music)

Construct	DbD	Comparison
Problem solving competency beliefs	.191	.364**
Fascination in design	.257*	.229
Value of CS	.201	.266
CS creative expression	.284**	.363**
E-Textiles coding self-efficacy	.321**	.335*
Programming fixed mindset	-.032	-.178
Programming growth mindset	.313**	.230+
Programming anxiety	.042	-.330*
Programming self-concept	.190	.297*

+p < .10, *p < .05, **p < .01

How did students' approaches to troubleshooting change from pre to post?

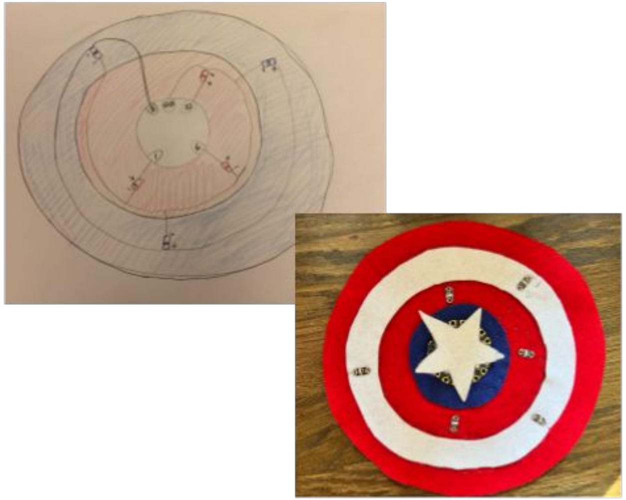
Morales-Navarro, L., Fields, D. A., Barapatre, D., & Kafai, Y. B. (2024, March). Failure Artifact Scenarios to Understand High School Students' Growth in Troubleshooting Physical Computing Projects. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (pp. 874-880).

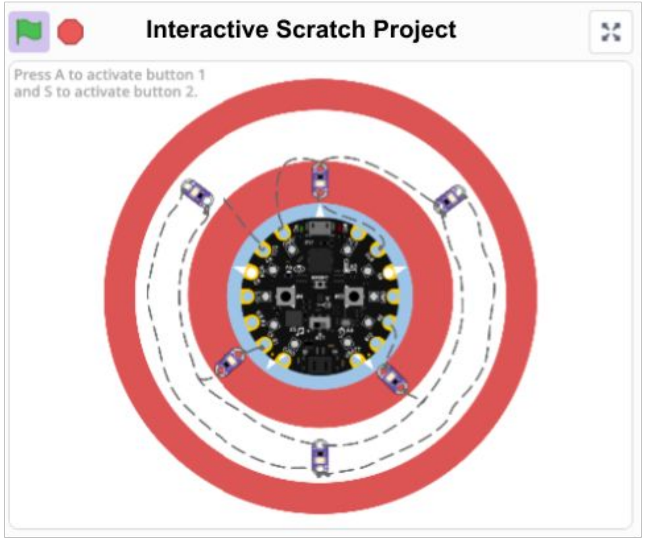
Broken Bird Project

The lights on this toy bird's eyes should light up and the bird should sing a melody whenever someone presses both its wings (silver patches on either side.)

But, one of the lights isn't turning on and the bird sings the melody only half way through whenever someone presses both its wings (silver patches on either side).

Faulty Captain America Project

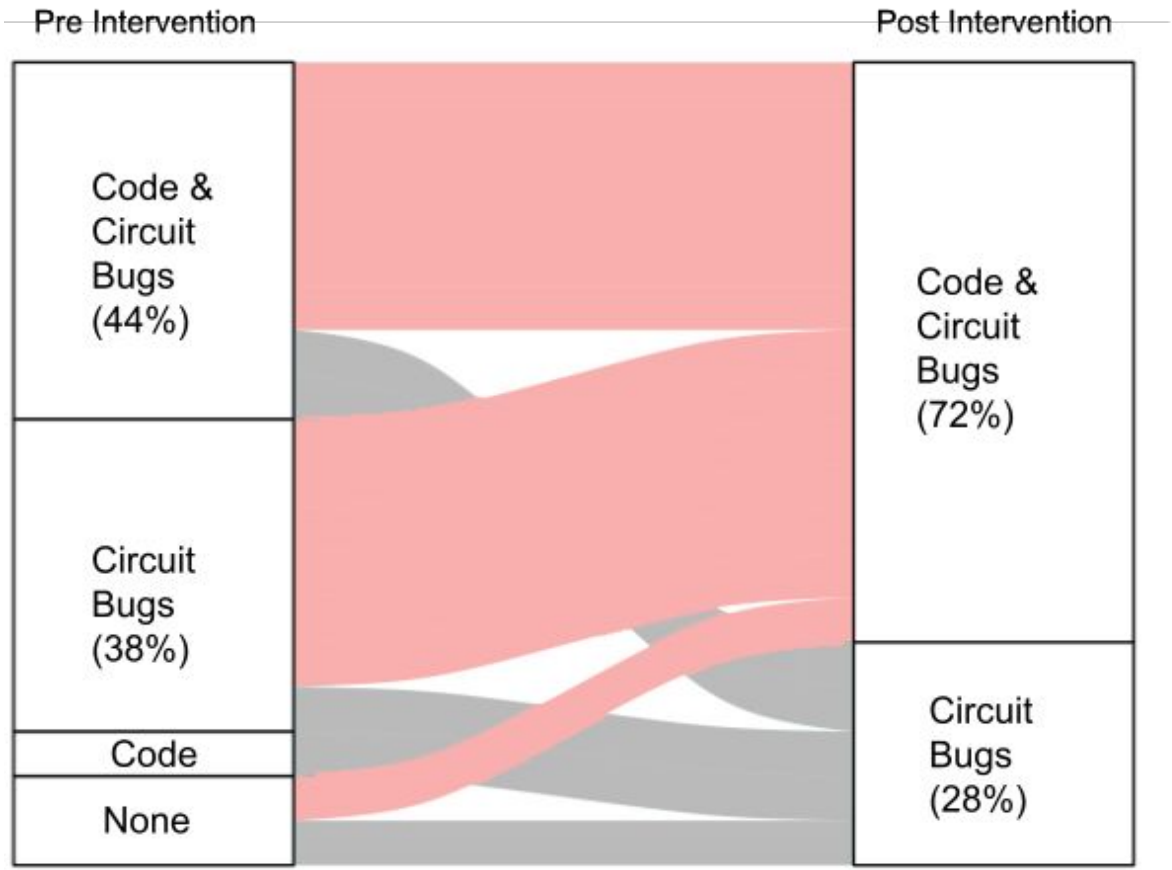




My project isn't working. It's supposed to do this:

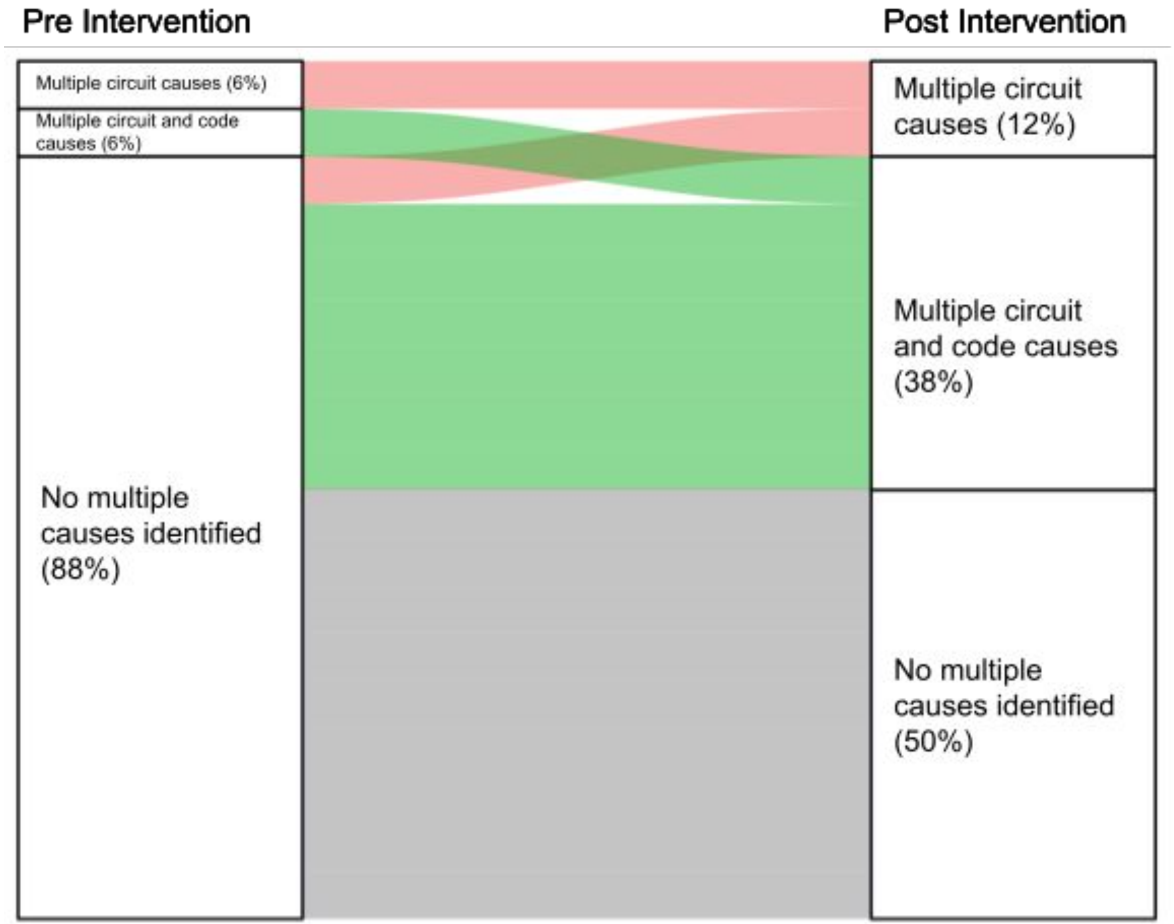
- if both buttons are pushed, the outer LEDs (on white ring) turn on, while the inner LEDs (on red ring) do a chase pattern; [condition partially worked: *inner lights ran alternate blinking pattern, outer lights did not turn on*]
- if button1 is pushed and button2 is not pushed, the outer LEDS (on the white ring) are off, and the inner LEDs (on the red ring) blink; [condition worked: *inner lights blinked*]
- if button1 is not pushed and button2 is pushed, the outer LEDs (on the white ring) blink, and the inner LEDS (on the red ring) are off; [condition did not work; *no lights turned on*]
- if neither button is pushed, then all LEDs are off. [condition worked: *no lights on*]

Identifying potential bugs across domains.



Alluvial diagram shows the distribution of students by types of bugs identified.

Identifying multiple causes for potential bugs.

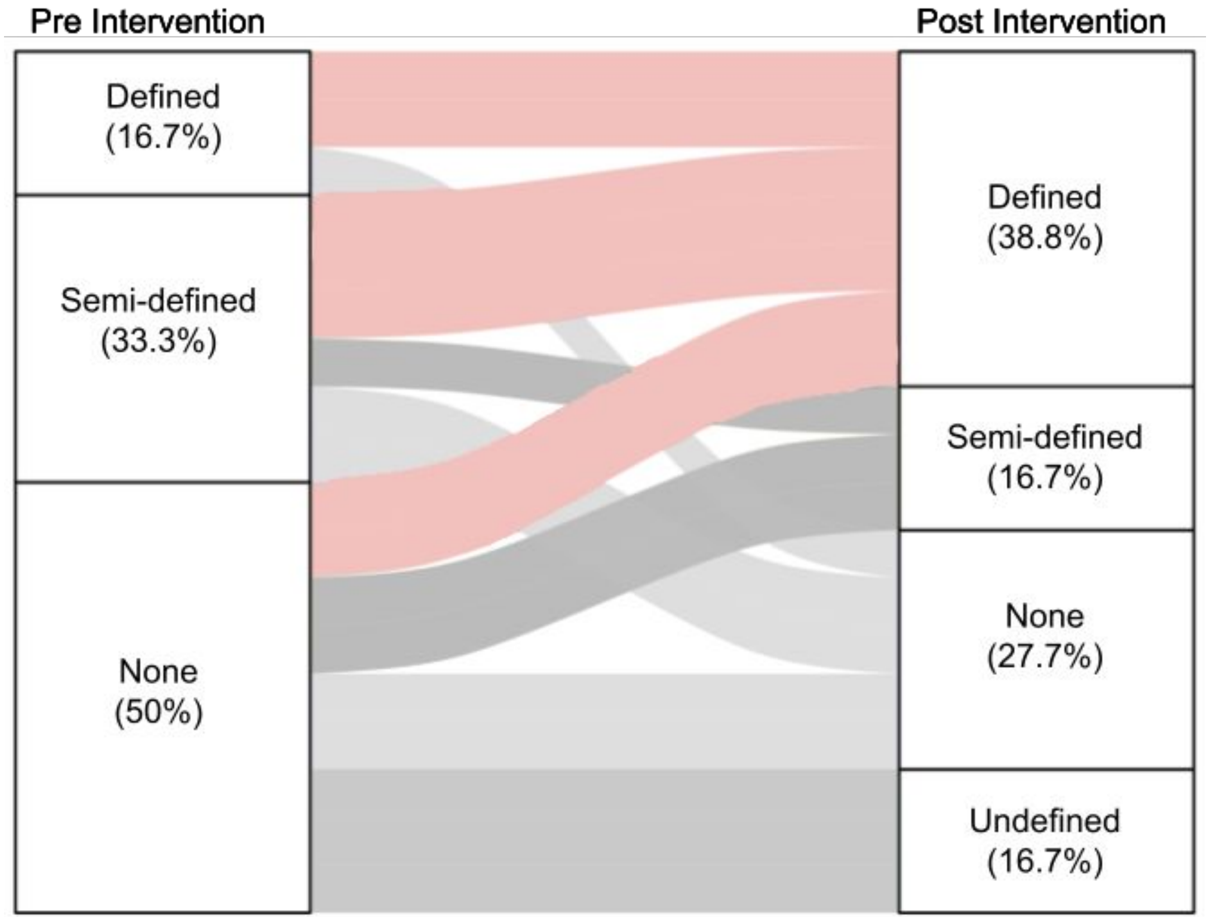


Alluvial diagram showing distribution of students that identified that a bug may be caused by more than one problem in a single domain or across domains.

Becoming more specific in identifying bugs

Bug Identification Categories	**Description: Identify coding as the relevant domain with varying levels of clear definition**	**Example**
Defined coding bugs	Identify a bug location, and identify a specific error.	“Using digitalRead() instead of digitalWrite() to change that LED to HIGH.” Damian
Semi-defined coding bugs	Identify either a bug location or a specific error, but not both.	“There’s something wrong in the for loop” Viviana
Undefined coding bugs	No bug locations or specific errors identified.	“Since there's two it's, maybe one of the lights isn't coded in yet” Amelia

Becoming more specific in identifying bugs



Alluvial diagram showing how student distribution by coding bugs identification category changed.

Takeaways

Designing buggy projects for peers to solve may support students in:

- reflecting, creating and documenting mistakes
- developing growth mindsets
- becoming more competent in identifying bugs across domains

Making Bugs for Learning

How youth can benefit from designing (and fixing) buggy projects with programmable microcontrollers

Luis Morales-Navarro, University of Pennsylvania
luismn@upenn.edu
luismn.com

Penn
UNIVERSITY of PENNSYLVANIA

NSF