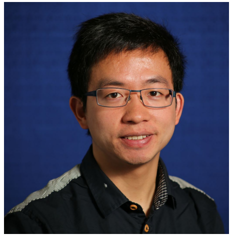
NC STATE UNIVERSITY

Exploring the Impact of Worked Examples in a Novice Programming Environment

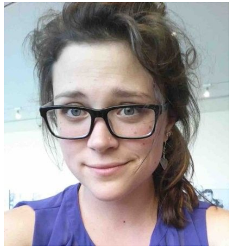
Rui Zhi



Thomas W. Price

Samiha Marwan

Alexandra Milliken



Tiffany Barnes

Min Chi

Center for Educational Informatics

Introduction

Task 1: Calculate $(x+4)(3x-2)$

$$(x+4)(3x-2)$$

$$= x \cdot 3x + x \cdot (-2) + 4 \cdot (3x) + 4 \cdot (-2)$$

$$= 3x^2 - 2x + 12x - 8$$

$$= 3x^2 + 10x - 8$$

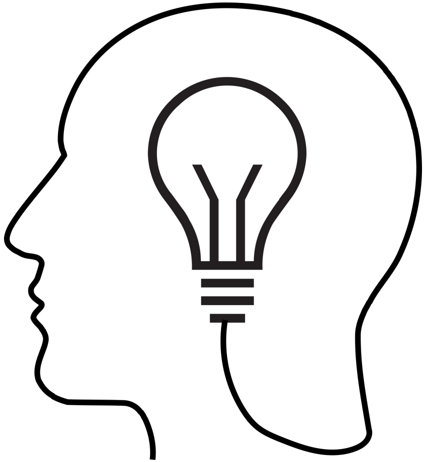
A worked example for a math problem (Chen et al., 2018)

Introduction

- Worked examples have been studied in a variety of domains and can increase learning efficiency (Sweller et. al, 1985; McLaren et. al., 2014)
- However, only a few studies have compared worked examples to traditional problem solving in novice programming environments (Van Merriënboer & De Croock, 1992)

Cognitive Load Theory

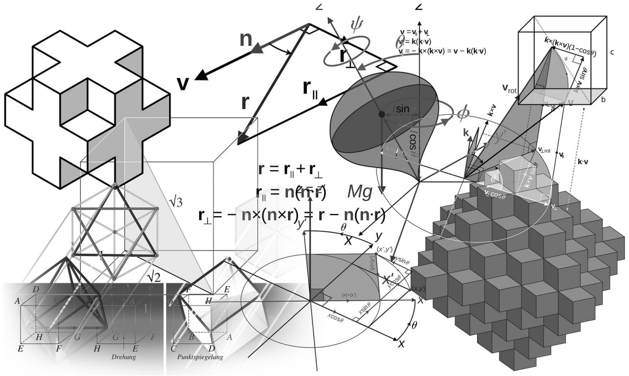
- Cognitive Load Theory (Sweller et al., 1998)

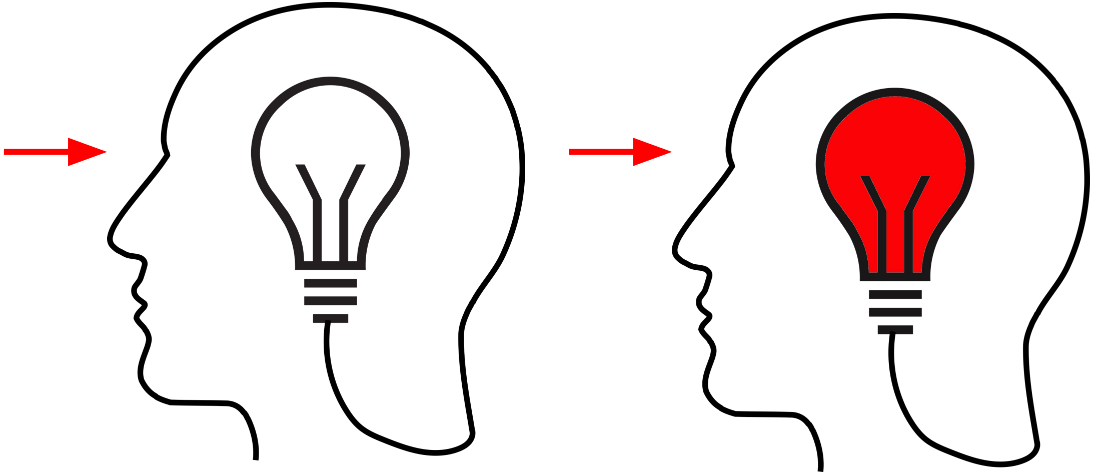


Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Cognitive Load Theory

- Cognitive Load Theory (Sweller et al., 1998)





CEI Center for Educational Informatics

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Cognitive Load Theory

Intrinsic

```
int a;
a = 5;
```

vs.

```
for (int i = 0; i < 5; i++) {
    …
}
```

Cognitive Load

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Cognitive Load Theory

Intrinsic

```
int a;
a = 5;
```

vs.

```
for (int i = 0; i < 5; i++) {
    …
}
```

Extraneous

"A triangle is a polygon with three edges and three vertices." - Wikipedia

Cognitive Load

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Cognitive Load Theory

Intrinsic

```
int a;
a = 5;
```

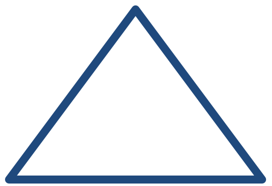
vs.

```
for (int i = 0; i < 5; i++) {
    …
}
```

Extraneous

“A triangle is a polygon with three edges and three vertices.” - Wikipedia

vs.



Cognitive Load

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Cognitive Load Theory

Intrinsic

```
int a;
a = 5;
```

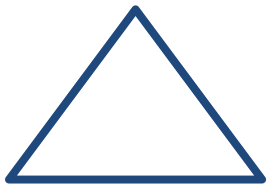
vs.

```
for (int i = 0; i < 5; i++) {
    …
}
```

Extraneous

"A triangle is a polygon with three edges and three vertices." - Wikipedia

vs.



Germane

Cognitive Load

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Cognitive Load Theory

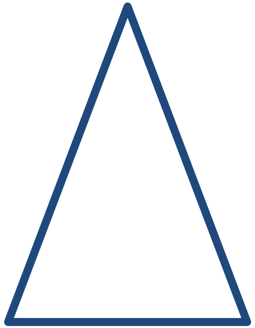
Intrinsic

```
int a;
a = 5;
```

vs.

```
for (int i = 0; i < 5; i++) {
    …
}
```

Extraneous



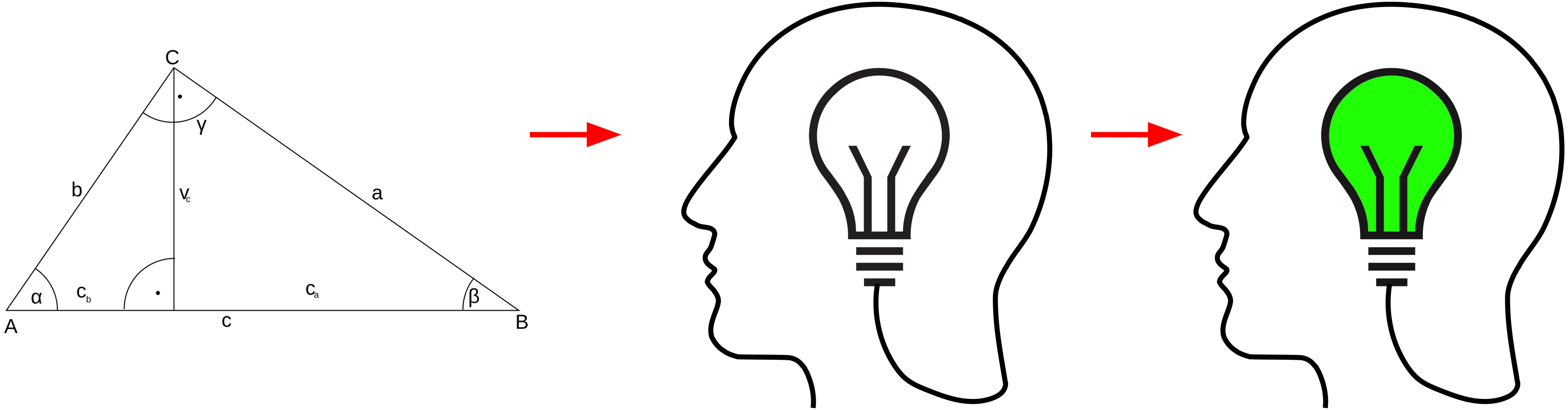
Germane

Cognitive Load

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Cognitive Load Theory

- Cognitive Load Theory (Sweller et al., 1998)



Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Worked Examples

- Teaches problem-solving procedure by showing solutions step by step

$$c(a+b) = \frac{af}{a}$$
$$c(a+b) = f$$
$$a+b = \frac{f}{c}$$
$$a = \frac{f}{c} - b$$

$$\frac{af+e}{b} = c$$
$$af + e = bc$$
$$af = bc - e$$
$$a = \frac{bc-e}{f}$$

(Sweller & Cooper, 1985)

Worked Examples in Programming

- Worked examples are one of the fundamental principles of programming education (Caspersen and Bennedsen, 2007)
- Suggest using worked examples in study materials and lectures (Vihavainen et al., 2011)
- Interleaving worked examples with practice problems can maximize students learning gains, compared to blocking WEs with problems, or solving equivalent problems (Trafton and Reiser, 1993)
- Incomplete worked examples improved novice's programming performance and post-test scores, compared with those who only had the WEs as a reference (MerrienBoer & Croock, 1992)
- It has been shown that combining self-explanation with WEs can be especially beneficial to students' learning (berthold, 2009)

Worked Examples in Programming

- Worked examples are one of the fundamental principles of programming education (Caspersen and Bennedsen, 2007)
- Suggest using worked examples in study materials and lectures (Vihavainen et al., 2011)
- Interleaving worked examples with practice problems can maximize students learning gains, compared to blocking WEs with problems, or solving equivalent problems (Trafton and Reiser, 1993)
- Incomplete worked examples improved novice's programming performance and post-test scores, compared with those who only had the WEs as a reference (MerrienBoer & Croock, 1992)
- It has been shown that combining self-explanation with WEs can be especially beneficial to students' learning (berthold, 2009)

Worked Examples in Programming

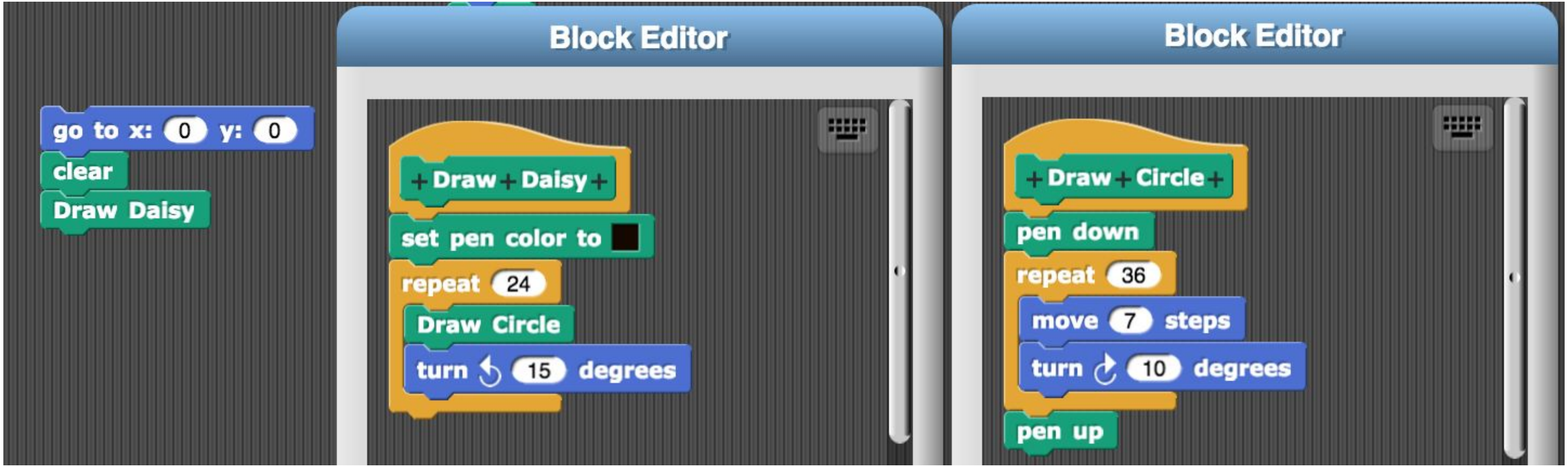
- Worked examples are one of the fundamental principles of programming education (Caspersen and Bennedsen, 2007)
- Suggest using worked examples in study materials and lectures (Vihavainen et al., 2011)
- Interleaving worked examples with practice problems can maximize students learning gains, compared to blocking WEs with problems, or solving equivalent problems (Trafton and Reiser, 1993)
- Incomplete worked examples improved novice's programming performance and post-test scores, compared with those who only had the WEs as a reference (MerrienBoer & Croock, 1992)
- It has been shown that combining self-explanation with WEs can be especially beneficial to students' learning (berthold, 2009)

Research Questions

How does having access to WEs during a programming problem impact:

- RQ1: Students' learning during the problem?
- RQ2: Students' perceived difficulty and cognitive load with respect to the problem?
- RQ3: Students' programming efficiency?

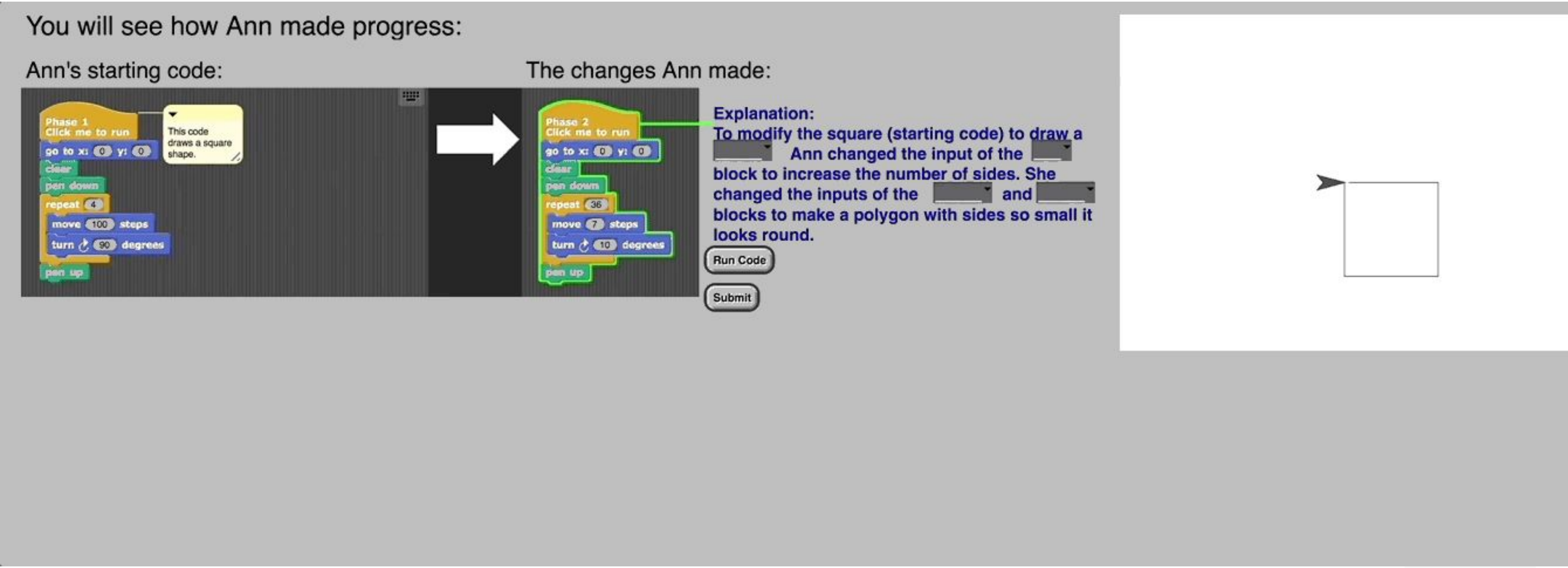
Peer Code Helper



Chunk expert solution procedure into meaningful steps and present to students

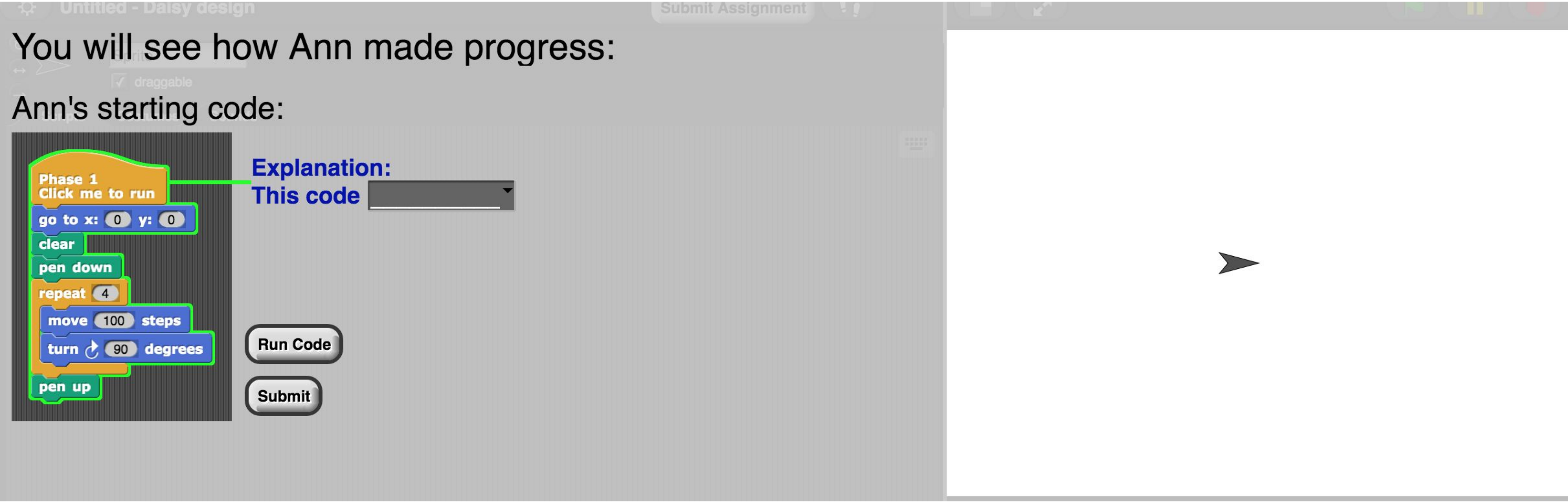
Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Peer Code Helper



Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Peer Code Helper



Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

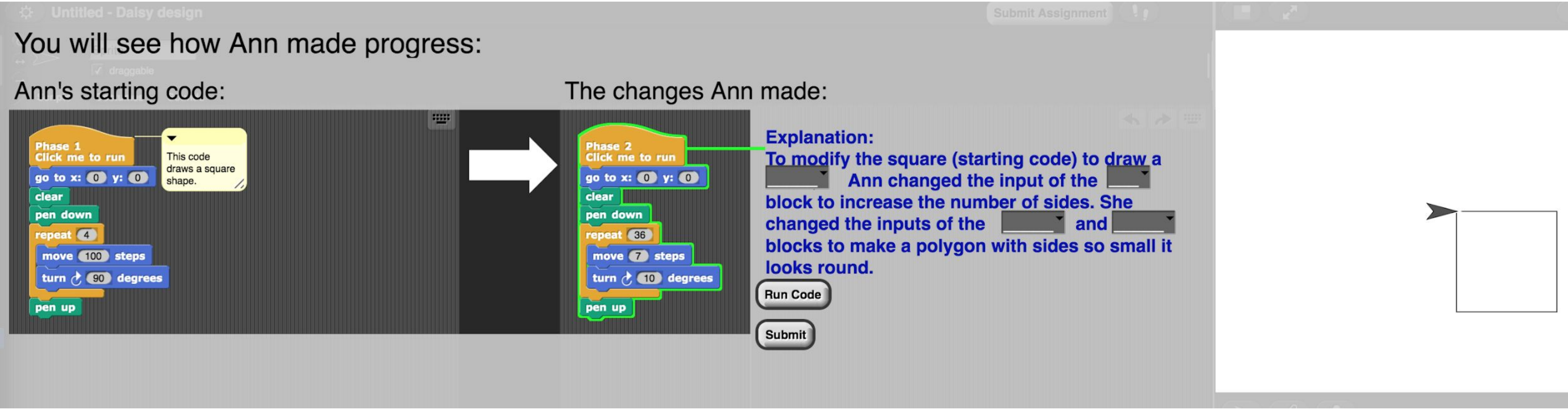
Peer Code Helper

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Peer Code Helper

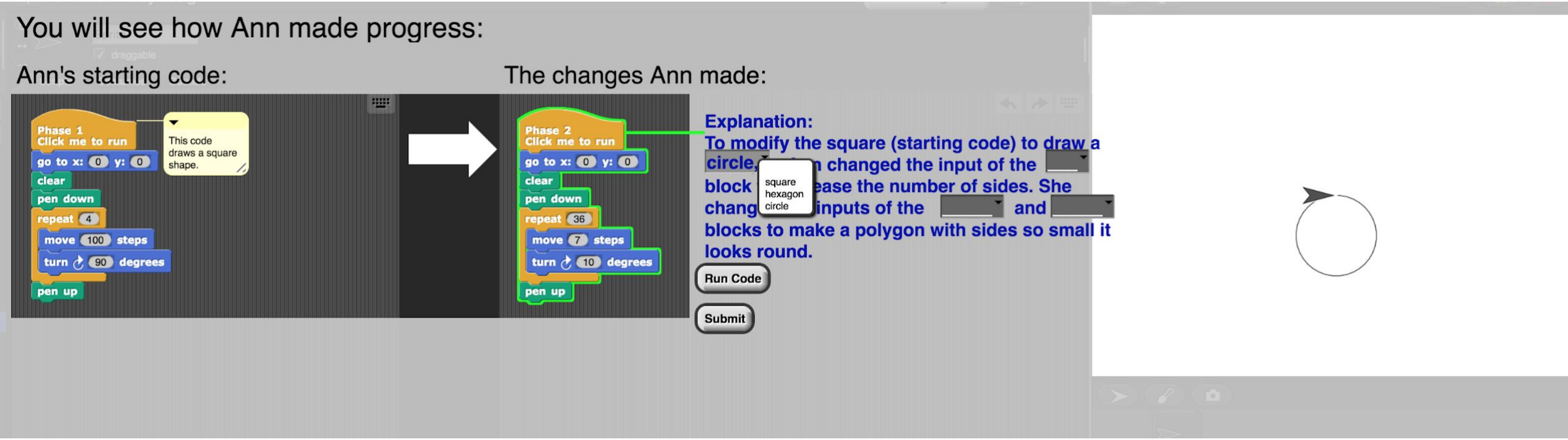
Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Peer Code Helper



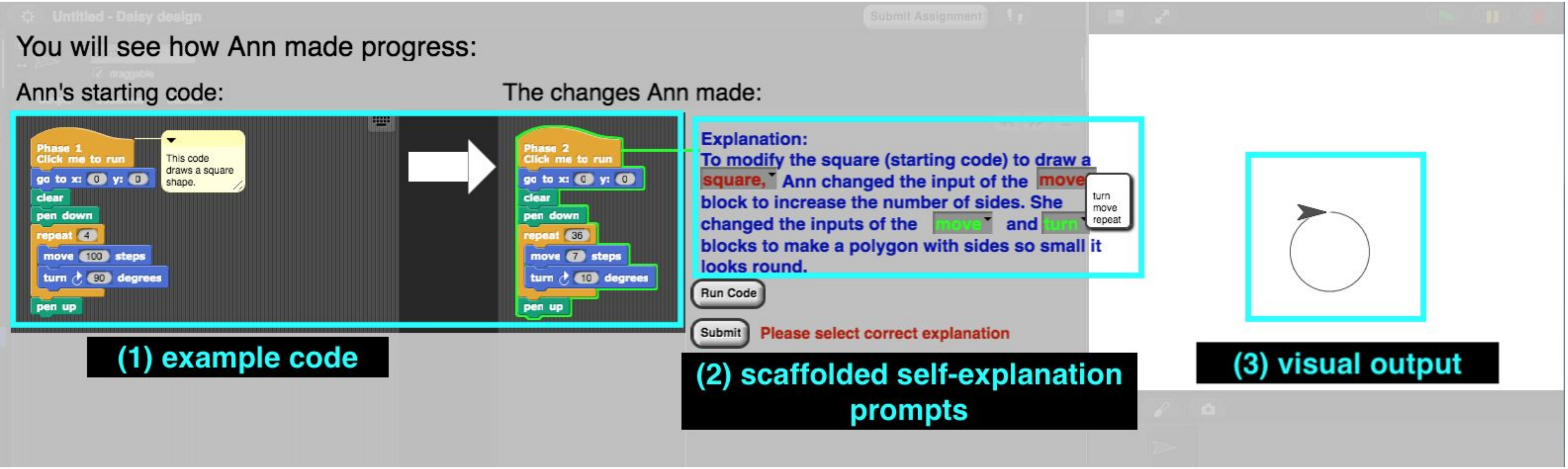
Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Peer Code Helper



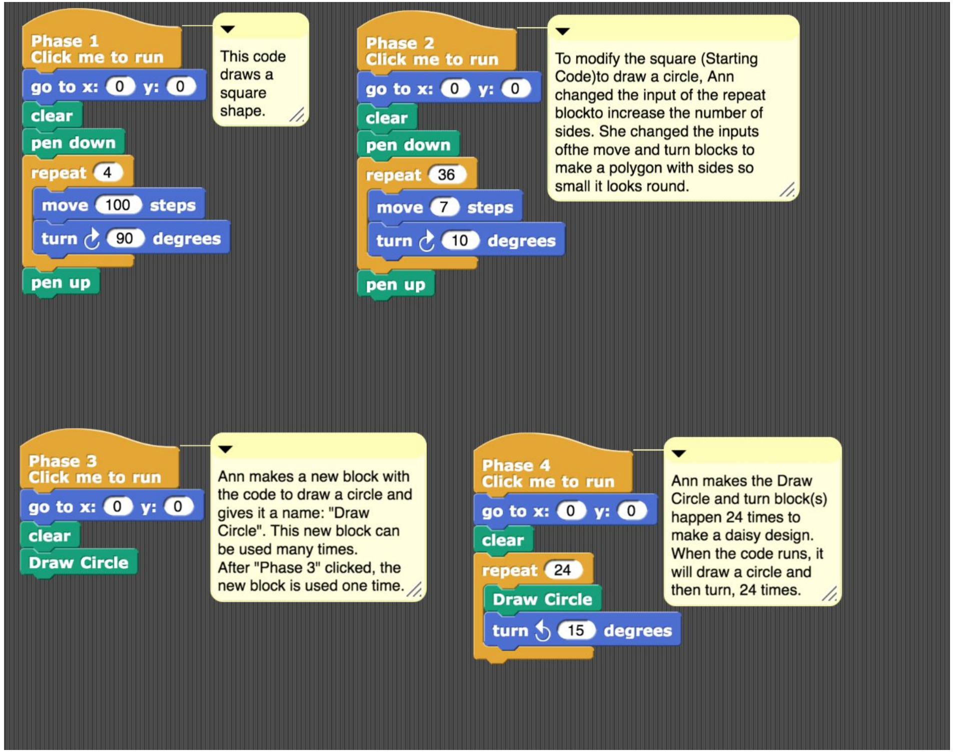
Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Peer Code Helper



Center for Educational Informatics

Peer Code Helper

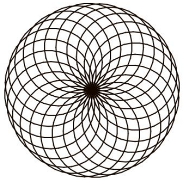


Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Participants & Procedure

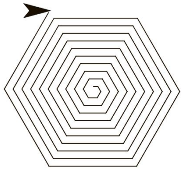
- Participants
  - 22 female high school students (ages 13 ~15)
  - Assigned to one of the two groups via matched pairs according to pre-test score
- Two groups

E1	E2
Problem 1 (with WEs)	Problem 1 (without WEs)
Problem 2 (without WEs)	**Problem 2 (with WEs)**



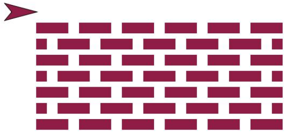
Procedures & Loops

Problem 1: Daisy Design



Procedures & Loops & Variables

Problem 2: Spiral Polygon



Procedures & Loops & Variables & Conditionals

Problem 3: Brick Wall

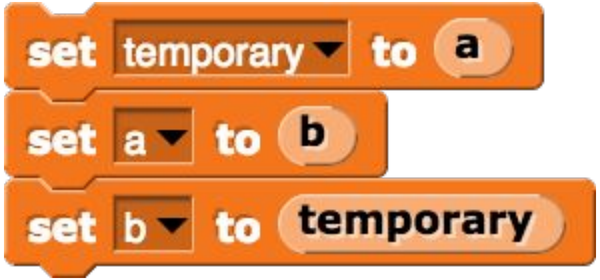
Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Study Outline

Step	Group E1	Group E2	Time
0	Snap*!* Introduction (taught by camp instructor)		90 minutes
1	Experience pre-survey + **Knowledge pre-test**		35 minutes
2	Introduce the Peer Code Helper		10 minutes
3	E1: Problem 1 (WEs)	E2: Problem 1 (no WEs)	45 minutes
4	**Post-test1** + Cognitive load survey		25 minutes
Second Day			
5	Re-introduce the Peer Code Helper		5 minutes
6	E1: Problem 2 (no WEs)	E2: Problem 2 (WEs)	45 minutes
7	**Post-test2** + Cognitive load survey		25 minutes
8	Problem 3 (Brick Wall, no WEs)		45 minutes
9	Demographics (post-survey) + Cognitive load survey		15 minutes

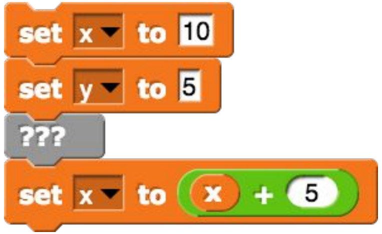
Pre-test and Post-tests Examples

a, b, and temporary are variables. What does this program do?



1. Makes a and b equal to each other
2. Rearranges the variables a, b, and temporary
3. This script does not do anything
4. Swaps the values of a and b

To ensure the value of x is 15 and y is 10 after running this script, which block is missing in the blocks below?



1. 

2. 

3. 

4. 

Adapted from the Commutative Assessments (Weintrop & Wilensky, 2015)

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Study Outline

Step	Group E1	Group E2	Time
0	Snap*!* Introduction (taught by camp instructor)		90 minutes
1	Experience pre-survey + Knowledge pre-test		35 minutes
2	Introduce the Peer Code Helper		10 minutes
3	E1: Problem 1 (WEs)	E2: Problem 1 (no WEs)	45 minutes
4	Post-test1 + **Cognitive load survey**		25 minutes
Second Day			
5	Re-introduce the Peer Code Helper		5 minutes
6	E1: Problem 2 (no WEs)	E2: Problem 2 (WEs)	45 minutes
7	Post-test2 + **Cognitive load survey**		25 minutes
8	Problem 3 (Brick Wall, no WEs)		45 minutes
9	Demographics (post-survey) + **Cognitive load survey**		15 minutes

Cognitive Load Survey (CS CLCS)

Intrinsic Load

1. The topics covered in the activity were very complex.
2. The activity covered program code that I thought was very complex.
3. The activity covered concepts and definitions that I thought were very complex.

Extraneous Load

4. The instructions and/or explanations during the activity were very unclear.
5. The instructions and/or explanations were very unhelpful for my learning.
6. The instructions and/or explanations were full of unclear language.

Germane Load

7.The activity really enhanced my understanding of the topic(s) covered.
8. The activity really enhanced my knowledge and understanding of computing/programming.
9. The activity really enhanced my understanding of the program code covered.
10. The activity really enhanced my understanding of the concepts and definitions.

(Morrison et al., 2014)

Study Outline

Step	Group E1	Group E2	Time
0	Snap*!* Introduction (taught by camp instructor)		90 minutes
1	Experience pre-survey + Knowledge pre-test		35 minutes
2	Introduce the Peer Code Helper		10 minutes
3	E1: Problem 1 (WEs)	E2: Problem 1 (no WEs)	45 minutes
4	Post-test1 + Cognitive load survey		25 minutes
Second Day			
5	Re-introduce the Peer Code Helper		5 minutes
6	E1: Problem 2 (no WEs)	E2: Problem 2 (WEs)	45 minutes
7	Post-test2 + Cognitive load survey		25 minutes
8	**Problem 3 (Brick Wall, no WEs)**		45 minutes
9	Demographics (post-survey) + Cognitive load survey		15 minutes

RQ1: Student Learning

How does having access to WEs during a programming problem impact students’ learning during the problem?

Pre- and Post-tests Results

Table: Mean (with SD) pre-test, post-test1, and post-test2 scores

	Pre-test	Post-test1	Post-test2
Group E1 (N=8)	8.13 (4.22)	8.38 (3.62)	9.13 (3.83)
Group E2 (N=8)	7.13 (4.02)	7.50 (3.21)	8.50 (4.21)

Pre- and Post-tests Results

Table: Mean (with SD) pre-test, post-test1, and post-test2 scores

	Pre-test	Post-test1	Post-test2
Group E1 (N=8)	8.13 (4.22)	8.38 (3.62)	9.13 (3.83)
Group E2 (N=8)	7.13 (4.02)	7.50 (3.21)	8.50 (4.21)

No significant difference on pre-test scores between groups: $t(13.96) = -0.4$, $p = .64$, $d = .24$

Pre- and Post-tests Results

Table: Mean (with SD) pre-test, post-test1, and post-test2 scores

	Pre-test	Post-test1	Post-test2
Group E1 (N=8)	8.13 (4.22)	8.38 (3.62)	9.13 (3.83)
Group E2 (N=8)	7.13 (4.02)	7.50 (3.21)	8.50 (4.21)

Main effect of test: ($F(2,28) = 5.26$, $\boldsymbol{p < .05}$, partial $\eta^2 = .27$)
Pre-test to post-test2: ($t(15) = 3.05$, $\boldsymbol{p < .01}$, $d = .30$)
Post-test1 to post-test2: ($t(15) = 3.05$, $\boldsymbol{p < .01}$, $d = .30$)

Pre-test to post-test1: ($t(15) = -0.86$, $p = .40$, $d = .08$)
No main effect of group: ($F(1,14) = 0.20$, $p = .66$, partial $\eta^2 = .014$)
No significant interaction between group and test: ($F(2,28) = 0.13$, $p = .88$, partial $\eta^2 = .009$)

RQ1: Student Learning

How does having access to WEs during a programming problem impact students' learning during the problem?

- Most of students' learning occurred during problem 2
  - Having time to reflect and digest the concepts learned in problem 1
- We did not find significant differences in learning between groups on the WE problems
  - Most students completed the core objectives

RQ2: Cognitive Load

How does having access to WEs during a programming problem impact students’ perceived difficulty and cognitive load with respect to the problem?

Cognitive Load Survey Results

Table: Mean (SD) factor score of cognitive load (IL - Intrinsic Load, EL - Extraneous Load, GL - Germane Load)

	Problem 1 - Daisy Design			Problem 2 - Spiral Polygon			Problem 3 - Brick Wall		
	IL	EL	GL	IL	EL	GL	IL	EL	GL
Group E1 (N = 8)	4.3 (2.4)	3.5 (3.7)	6.6 (3.2)	6.4 (2.4)	3.3 (2.8)	6.7 (2.4)	5.3 (3.3)	3.4 (2.5)	7.9 (2.6)
Group E2 (N = 8)	4.9 (2.6)	3.4 (2.6)	8.6 (1.6)	3.8 (1.9)	2.7 (1.7)	8.0 (2.1)	6.3 (2.8)	4.9 (3.5)	7.6 (2.3)

No main effect of group: ($F(1,14) = 0.10$, $p = .76$, partial $\eta^2 = .007$)
No main effect of problem: ($F(2, 28) = 1.78$, $p = .19$, partial $\eta^2 = .011$)

Significant interaction between group and problem: ($F(2,28) = 4.65$, $\boldsymbol{p = .05}$, partial $\eta^2 = .25$)

Cognitive Load Survey Results

Table: Mean (SD) factor score of cognitive load (IL - Intrinsic Load, EL - Extraneous Load, GL - Germane Load)

	Problem 1 - Daisy Design			Problem 2 - Spiral Polygon			Problem 3 - Brick Wall		
	IL	EL	GL	IL	EL	GL	IL	EL	GL
Group E1 (N = 8)	4.3 (2.4)	3.5 (3.7)	6.6 (3.2)	**6.4 (2.4)**	3.3 (2.8)	6.7 (2.4)	5.3 (3.3)	3.4 (2.5)	7.9 (2.6)
Group E2 (N = 8)	4.9 (2.6)	3.4 (2.6)	8.6 (1.6)	**3.8 (1.9)**	2.7 (1.7)	8.0 (2.1)	6.3 (2.8)	4.9 (3.5)	7.6 (2.3)

E1P1 vs. E2P1: $t(13.91) = 0.40$, $p = .69$, $d = 0.20$
E1P2 vs. E2P2: $t(13.19) = -2.33$, $\boldsymbol{p < .05}$, $d = -1.16$

Why?

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

Cognitive Load Survey Results

Table: Mean (SD) factor score of cognitive load (IL - Intrinsic Load, EL - Extraneous Load, GL - Germane Load)

	Problem 1 - Daisy Design			Problem 2 - Spiral Polygon			Problem 3 - Brick Wall		
	IL	EL	GL	IL	EL	GL	IL	EL	GL
Group E1 (N = 8)	4.3 (2.4)	3.5 (3.7)	6.6 (3.2)	**6.4 (2.4)**	3.3 (2.8)	6.7 (2.4)	5.3 (3.3)	3.4 (2.5)	7.9 (2.6)
Group E2 (N = 8)	4.9 (2.6)	3.4 (2.6)	8.6 (1.6)	**3.8 (1.9)**	2.7 (1.7)	8.0 (2.1)	6.3 (2.8)	4.9 (3.5)	7.6 (2.3)

Possible explanations:
- WEs reduce intrinsic load
- WEs represent an inherently different learning task than problem solving
- The self-reported instrument may not be accurate

Cognitive Load Survey Results

Table: Mean (SD) factor score of cognitive load (IL - Intrinsic Load, EL - Extraneous Load, GL - Germane Load)

	Problem 1 - Daisy Design			Problem 2 - Spiral Polygon			Problem 3 - Brick Wall		
	IL	EL	GL	IL	EL	GL	IL	EL	GL
Group E1 (N = 8)	4.3 (2.4)	3.5 (3.7)	6.6 (3.2)	**6.4 (2.4)**	3.3 (2.8)	6.7 (2.4)	5.3 (3.3)	3.4 (2.5)	7.9 (2.6)
Group E2 (N = 8)	4.9 (2.6)	3.4 (2.6)	8.6 (1.6)	3.8 (1.9)	2.7 (1.7)	8.0 (2.1)	**6.3 (2.8)**	4.9 (3.5)	7.6 (2.3)

E1P1 vs. E1P2: $t(7) = -3.51$, $\boldsymbol{p < .01}$, $d = 0.83$
E2P2 vs. E2P3: $t(7) = -4.52$, $\boldsymbol{p < .01}$, $d = 1.04$

Center for Educational Informatics

Cognitive Load Survey Results

Table: Mean (SD) factor score of cognitive load (IL - Intrinsic Load, EL - Extraneous Load, GL - Germane Load)

	Problem 1 - Daisy Design			Problem 2 - Spiral Polygon			Problem 3 - Brick Wall		
	IL	EL	GL	IL	EL	GL	IL	EL	GL
Group E1 (N = 8)	4.3 (2.4)	3.5 (3.7)	6.6 (3.2)	**6.4 (2.4)**	3.3 (2.8)	6.7 (2.4)	5.3 (3.3)	3.4 (2.5)	7.9 (2.6)
Group E2 (N = 8)	4.9 (2.6)	3.4 (2.6)	8.6 (1.6)	3.8 (1.9)	2.7 (1.7)	8.0 (2.1)	**6.3 (2.8)**	4.9 (3.5)	7.6 (2.3)

WEs may increase students' perceived difficulty of problem solving immediate following WEs

RQ2: Cognitive Load

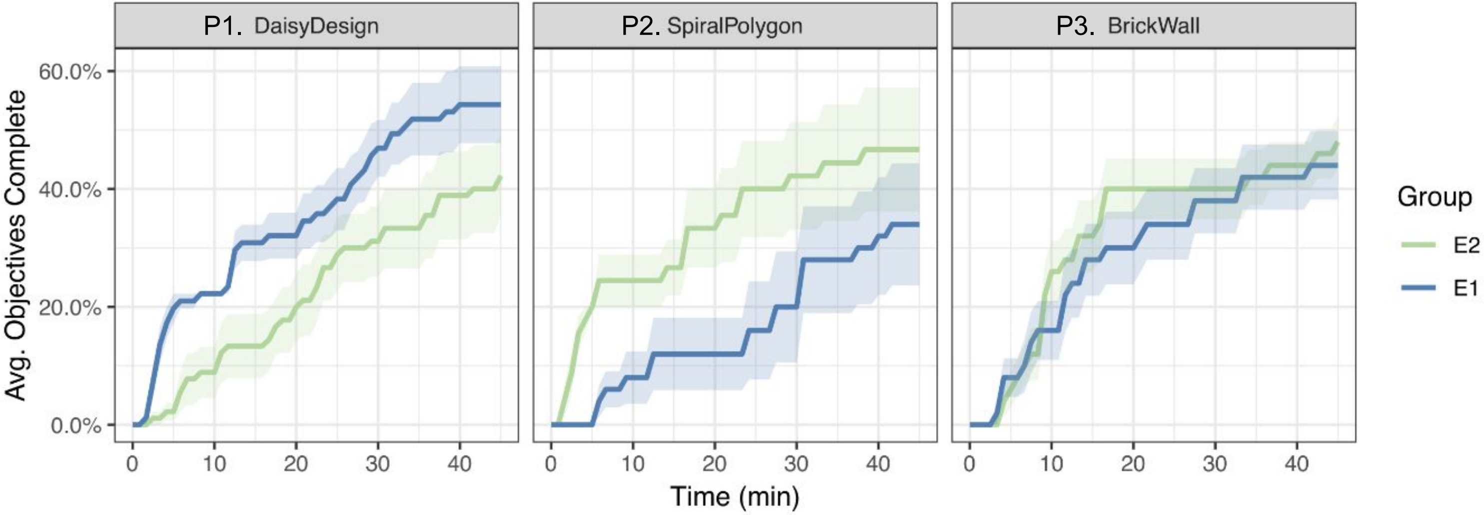
How does having access to WEs during a programming problem impact students' perceived difficulty and cognitive load with respect to the problem?

- We found significant differences between the groups' **intrinsic cognitive load** for problem 2 but not for problem 1
- We also found both groups experienced higher intrinsic load on problems *without WEs* that followed problems *with WEs*

RQ3: Programming Efficiency

How does having access to WEs during a programming problem impact students' programming efficiency?

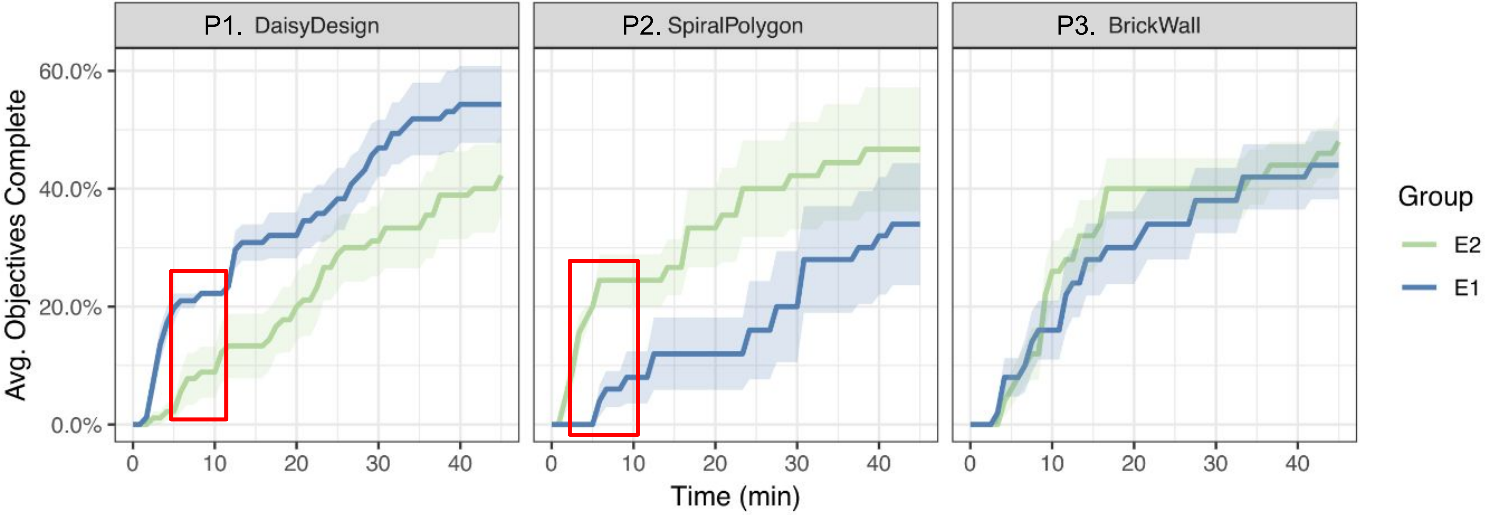
Objectives Completed Over Time



The average number of objectives completed by each group over time, with shading indicating ±1 standard error.

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

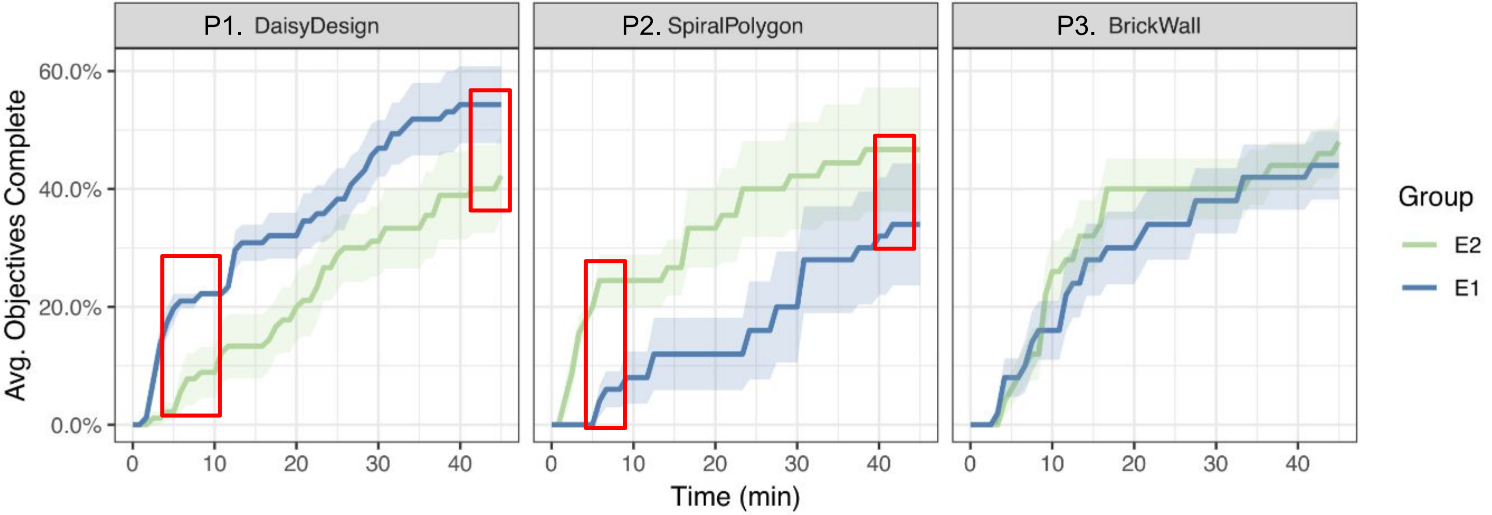
Objectives Completed Over Time



The average number of objectives completed by each group over time, with shading indicating ±1 standard error.

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

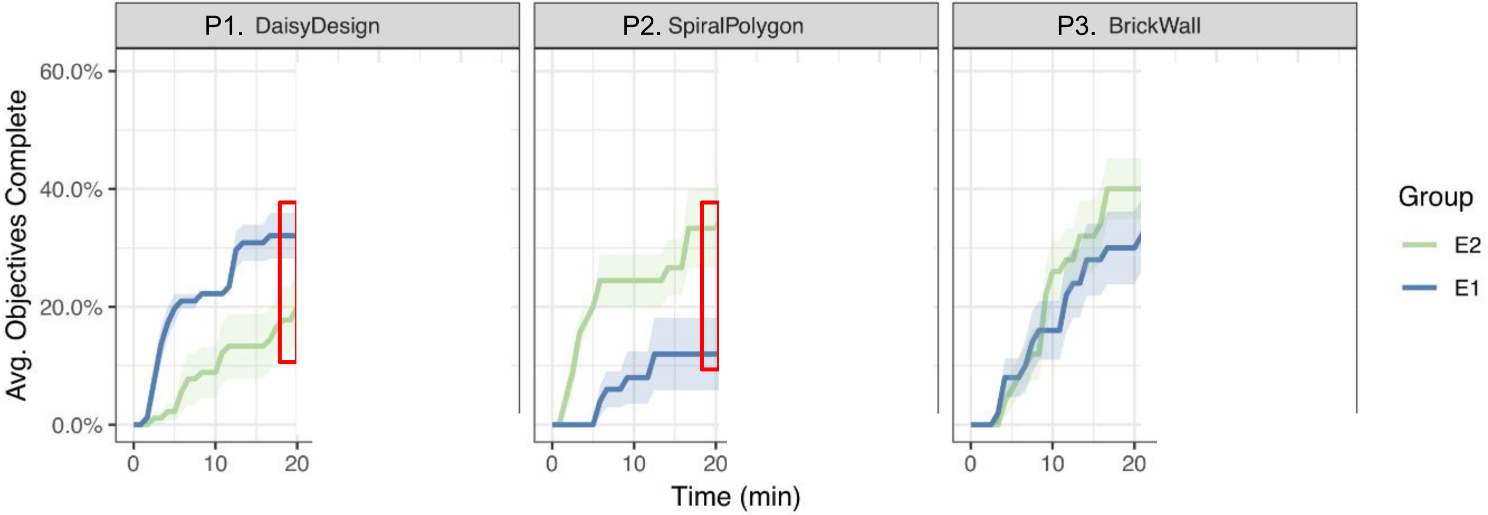
Objectives Completed Over Time



The average number of objectives completed by each group over time, with shading indicating ±1 standard error.

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

If We Cut the Time ...



The average number of objectives completed by each group over time, with shading indicating ±1 standard error.

Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, Min Chi

RQ3: Programming Efficiency

How does having access to WEs during a programming problem impact students' programming efficiency?

- Our analysis suggests that WEs save students considerable time in completing programming objectives, but that students take longer to complete later objectives

Post-survey Feedback

Would you like to have the Peer Code Helper on future programming activities?

- 35% yes
  - Appreciate the PCH
    - “see how to go from one step to the next”
- 12% no
  - had very high pre-test scores (over 75%)
    - More advanced students may not appreciate worked examples (Kalyuga et al., 2003)
- 53% uncertain
  - Prefer the challenge of working independently
    - “It’s good to have a challenge, but it’s also nice... to make it a little bit easier”

Conclusion

- Worked examples may have an effect on students' intrinsic cognitive load
- Programming worked examples may improve students' programming efficiency in the short term, but that students do require additional time to process WEs before they can construct their own code

Thank you for your time!

Questions?

Rui Zhi
rzhi@ncsu.edu

Advisor:
Dr. Tiffany Barnes
Dr. Thomas W. Price

Center for Educational Informatics

Cognitive Load Survey (CS CLCS)

Intrinsic Load
1. The topics covered in the activity were very complex.
2. The activity covered program code that I thought was very complex.
3. The activity covered concepts and definitions that I thought were very complex.

Extraneous Load
4. The instructions and/or explanations during the activity were very unclear.
5. The instructions and/or explanations were very unhelpful for my learning.
6. The instructions and/or explanations were full of unclear language.

Germane Load
7.The activity really enhanced my understanding of the topic(s) covered.
8. The activity really enhanced my knowledge and understanding of computing/programming.
9. The activity really enhanced my understanding of the program code covered.
10. The activity really enhanced my understanding of the concepts and definitions.

(Morrison et al., 2014)