



# Enabling Independent Learning of Programming Concepts through Programming Completion Puzzles

**Kyle J. Harms**, Noah Rowlett, Caitlin Kelleher



Washington University in St. Louis



# Novices Learning Programming - Classroom

## Secondary schools offering introductory (or pre-AP) Computer Science courses, change from 2005 baseline

---

	2007	2009
Yes	-6%	-17%

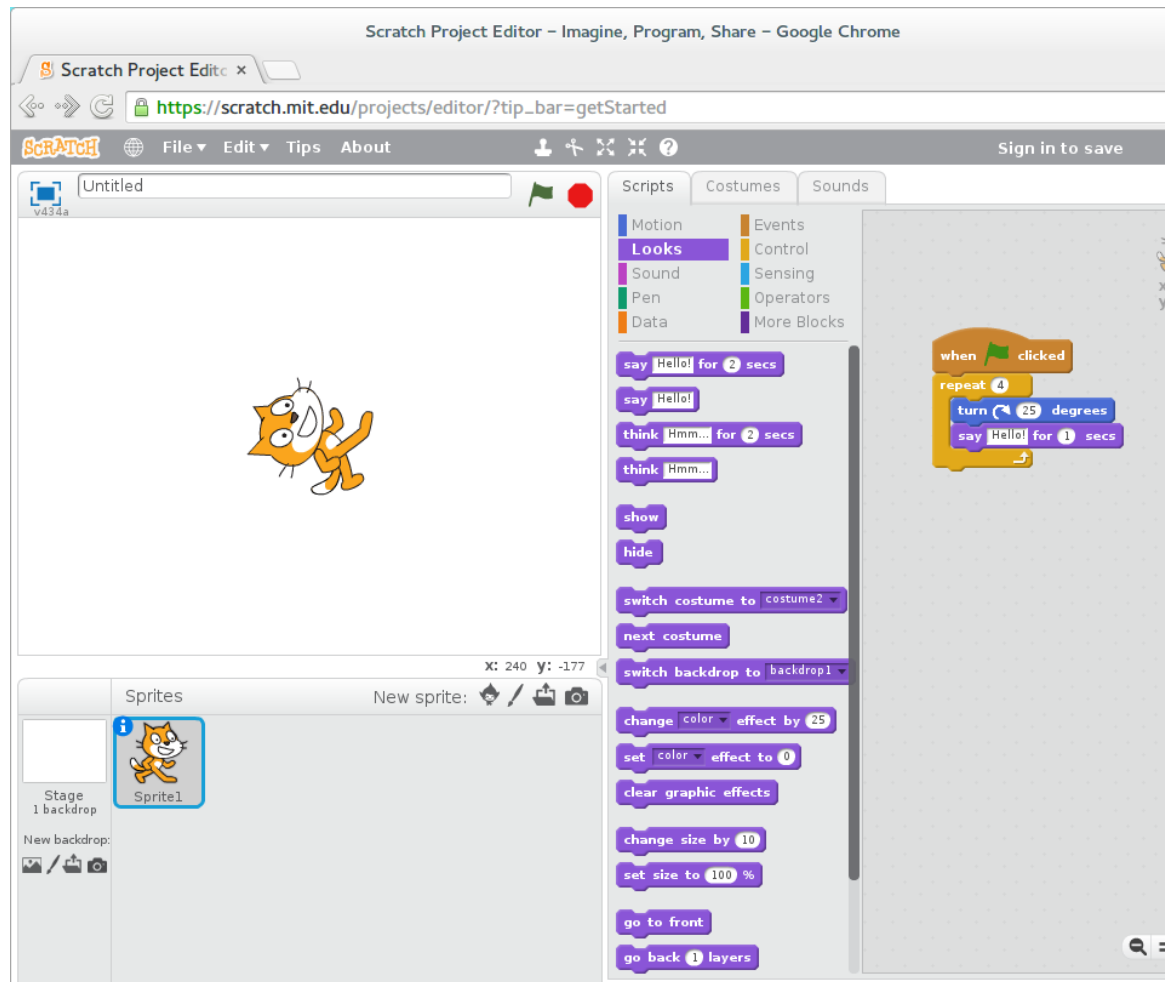
## Secondary offering AP Computer Science courses, change from 2005 baseline

---

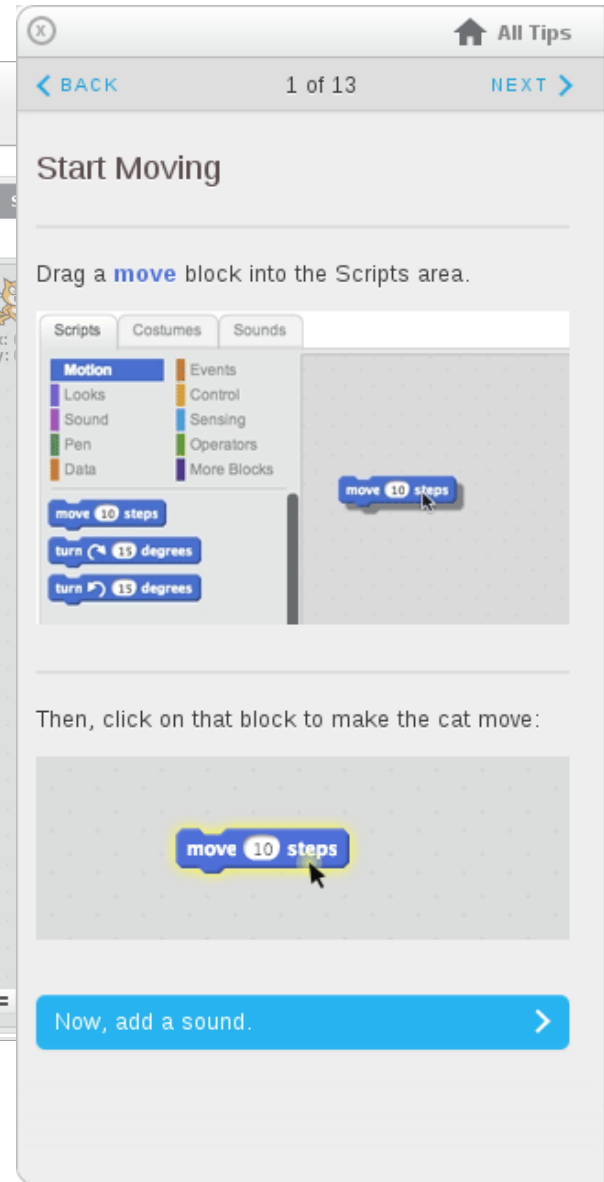
	2007	2009
Yes	-20%	-35%



# Learning Programming Independently - Tutorials



**Scratch**





# Learning Programming Independently - Puzzle-Like Systems

The screenshot displays the code.org - Hour of Code interface. At the top, the browser address bar shows "Code.org - Hour of Code #8" and "http://studio.code.org/hoc/8". Below this, the "Hour of Code" progress bar shows 8 out of 10 steps completed. The main interface is divided into three sections: a "STUDIO" area on the left, a "code" editor in the center, and a "world" grid on the right.

**STUDIO:** This area contains a visual representation of the puzzle world. It shows a grid with various objects like a piggy bank, a basket, a goop, a rock, a kitten, and a bucket. A "Run" button is visible below the grid. A message from the Angry Birds character says: "Help me banish this bad piggy using the fewest number of blocks. Try using more than one 'repeat times' block."

**code:** This section contains a code editor with two tabs: "Original Code" and "Clear Code". The code is as follows:

```
goto /goop/  
grab /goop/  
left 4  
down  
drop /kitten/  
goto /basket/  
left  
up 2  
drop /rock/  
drop /goop/  
  
ensure /kitten/:position = /basket/:position  
ensure /piglet/:position = /basket/:position  
ensure /goop/:position = /bucket/:position  
ensure /rock/:position = /bucket/:position
```

Below the code editor are four buttons: "One step", "One line", "To end", and "Stop!".

**world:** This section shows a 5x5 grid representing the puzzle world. The grid contains various objects and their positions:

	0	1	2	3	4
0					
1		piglet			
2			basket	goop	
3	rock				
4			kitten		bucket

A speech bubble from the kitten character says: "Try running my starting code to see how goto works! My goal is to get everything into the correct containers, but remember that I use more energy as I carry more things."

code.org – Hour of Code

Gidget



# More Puzzles...

Lab 1

Syntax practise

Drag the statements on the right into the correct order. Do not use the incorrect statements.  
Should produce four lines of text- read them to get them in the correct order.

Check

1

2

3

4

5

6

7

8

WriteIn('This is the first line');

WriteIn('Don't use a line that's not correct' as line 2);

WriteIn(5.67:2);

Begin  
InitOurCrt;

WriteIn('Don't use a line that's not correct as line 2');

WriteIn(5.67234:5:2);

Program Pick\_the\_correct\_lines;  
{ \$APPTYPE CONSOLE }

End.

End;

uses SysUtils, OurCrt;

WriteIn('The line below should show the number 5.67');

## Parson's Programming Puzzles

5

D. Parsons and P. Haden, "Parson's Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses," in Proceedings of the 8th Australasian Conference on Computing Education - Volume 52, Darlinghurst, Australia, Australia, 2006, pp. 157–163.





# Learning Programming Independently

## - Completion Problems

- Generation
  - Write programs from scratch
- Completion
  - Complete partially written programs






# Learning Programming Independently Picture

**Completion  
Problems**

**Puzzle-like**

**Tutorials**



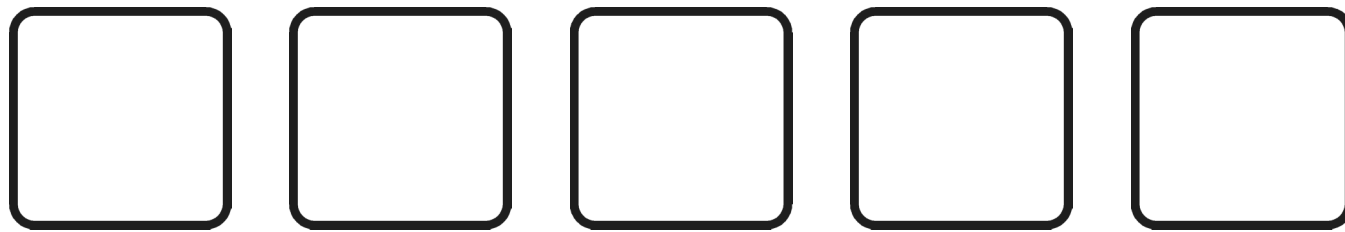


How do we effectively use  
puzzles to support novices  
learning programming  
independently?



# Cognitive Load Theory

- Completion Problems
- Extraneous Cognitive Load

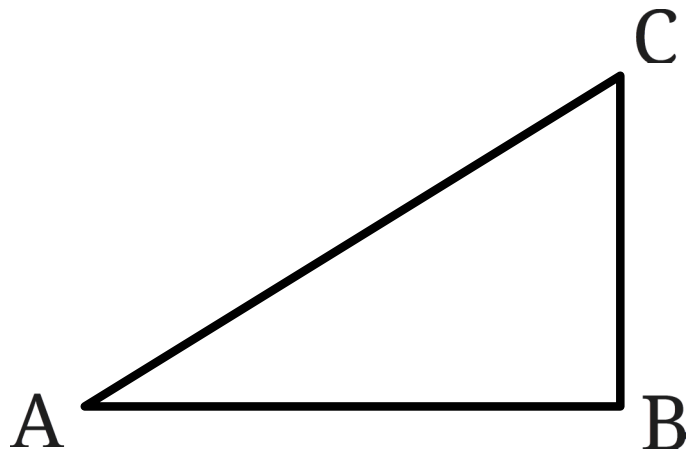


**Working Memory**



# Extraneous Cognitive Load

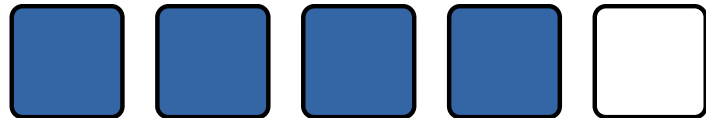
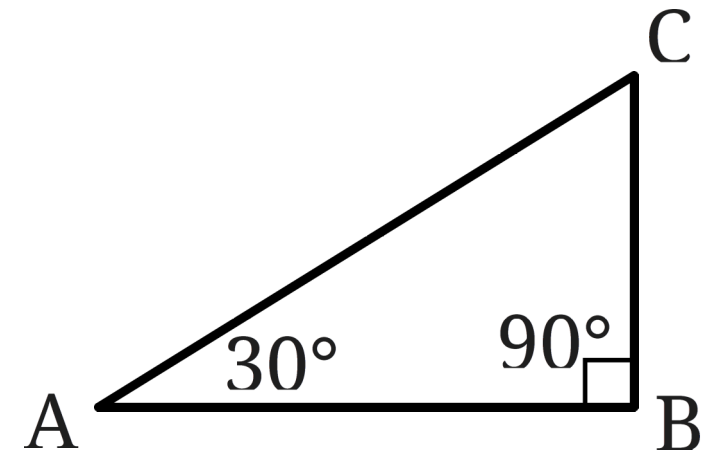
What is angle ACB?



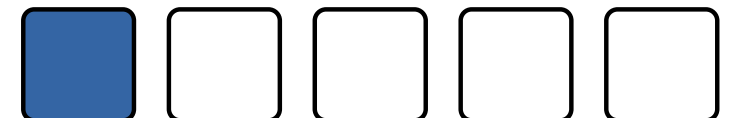
$$\angle ABC = 90^\circ$$
$$\angle CAB = 30^\circ$$



What is angle ACB?



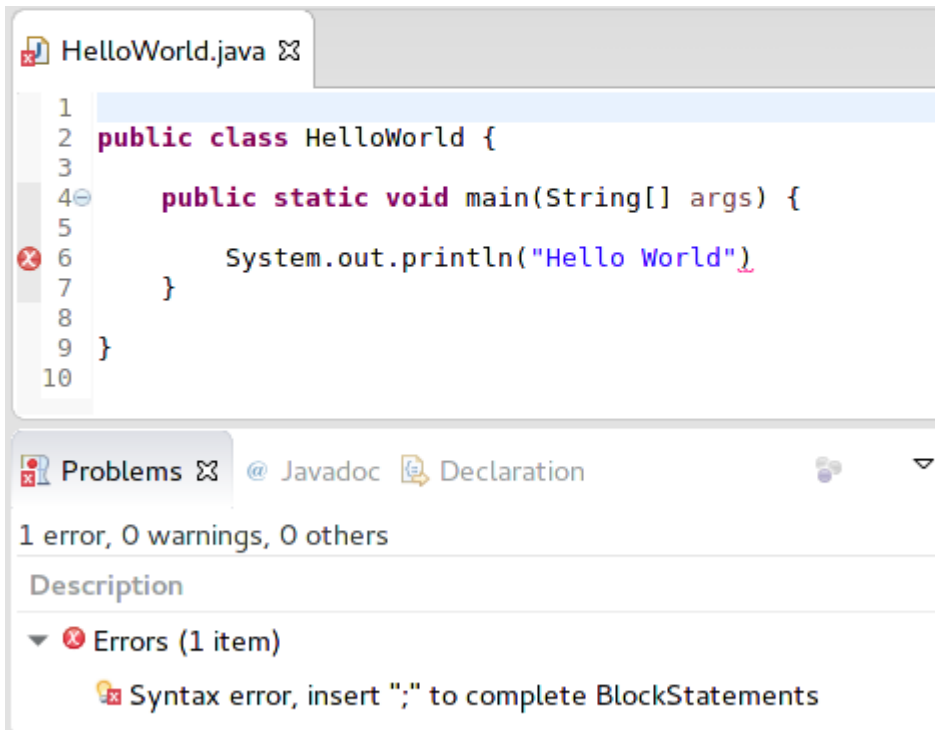
**High Extraneous Cognitive Load**



**Low Extraneous Cognitive Load**



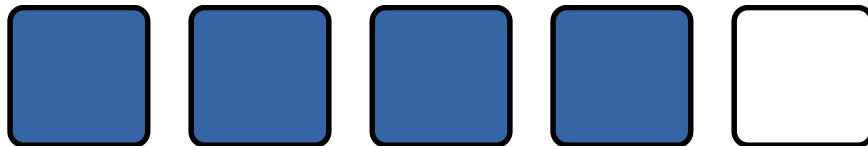
# Example: CS1 First Program



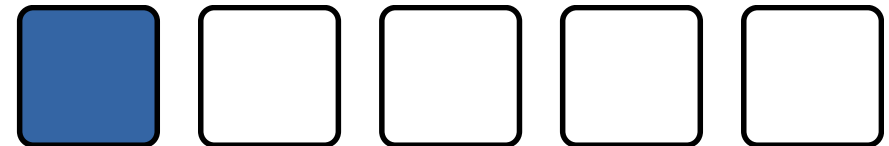
The screenshot shows an IDE with a file named `HelloWorld.java`. The code is as follows:

```
1  
2 public class HelloWorld {  
3  
4     public static void main(String[] args) {  
5  
6         System.out.println("Hello World")  
7     }  
8  
9 }  
10
```

Below the code editor, the **Problems** panel shows 1 error, 0 warnings, and 0 others. The error description is: **Syntax error, insert ";" to complete BlockStatements**. The error points to the end of the `main` method block.



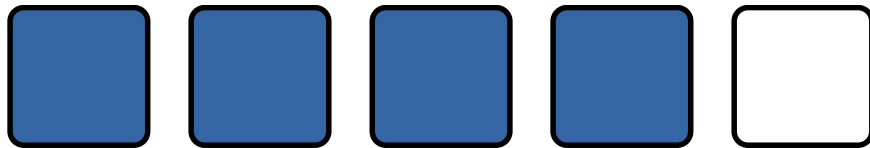
**High Extraneous Cognitive Load**



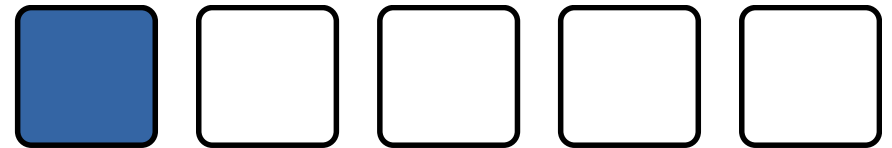
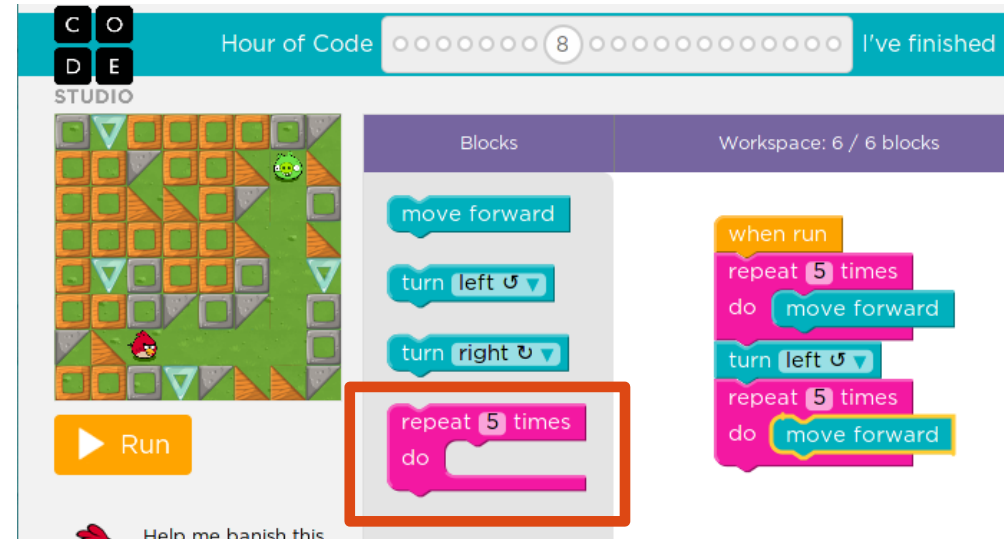
**Low Extraneous Cognitive Load**



# Example: Drag 'n Drop



**High Extraneous Cognitive Load**



**Low Extraneous Cognitive Load**






# Completion Problems

- Generation (write from scratch)
  - High extraneous cognitive load
- Completion (complete partial program)
  - More working memory resources available for learning





Can we use completion problems and also leverage the strengths of puzzle-like systems to provide an effective way to help novices learn programming independently?



# Completion Problems → Programming Completion Puzzles

The screenshot shows a programming environment window titled "\*alien-and-ufo.lgp - Looking Glass". The interface is divided into several sections:

- Scene View:** A 3D scene showing a green alien character on a grey, cratered surface next to a colorful flying saucer. A tree stump is visible in the background.
- Buttons:** Below the scene view are two buttons: "Play Correct" (green) and "Play Mine" (blue).
- Instructions:** A red lifebuoy icon is followed by the text: "Use all of these actions to put the animation back in the correct order."
- Available Actions:** A dashed box contains a "Do together" block and three individual action blocks:
  - "flying saucer beam up alien"
  - "flying saucer fly away"
- My Story Editor:** A yellow panel on the right titled "My Story" contains a sequence of actions:
  - Undo, Redo, Clear buttons.
  - "alien walk 0.5 meters"
  - "Repeat 2 times" loop containing:
    - "alien kick"
    - "flying saucer shake"
  - "loop" block
  - "flying saucer fly above alien"
  - "flying saucer turn RIGHT 2.0 rotations duration 2.0 seconds"



# Puzzle Curriculum

Easy



**1. Sequential**



**4. Repeated & Parallel**

Challenging



**2. Repeated**



**3. Parallel**



**5. Parallel {  
Repeated  
}**



**6. Repeated {  
Parallel  
}**





▶ Play Correct

▶ Play Mine



Use all of these actions to put the animation back in the correct order.

Repeat 2 times

loop

flying saucer fly above alien

flying saucer shake

flying saucer fly away

alien kick

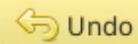
flying saucer beam up alien

flying saucer turn RIGHT 2.0 rotations du

Do together

alien walk 0.5 meters

## custom action My Story



Undo



Redo



Clear

Done X

Drop action here.



- 
- **Lessons Learned**
  - **Programming Completion**  
**Puzzle Effectiveness**



- 
- **Lessons Learned**
  - **Programming Completion**  
**Puzzle Effectiveness**





# Formative Evaluations

- Completion Problem → Puzzle Format & Interface
  - 10 iterations
  - 23 participants - St. Louis Science Center
  - 30 minutes
- Puzzle Curriculum
  - 8 iterations
  - 21 participants - St. Louis Academy of Science
  - 90 minutes



# Lessons Learned:

- 1) Limit the editable dimensions of the puzzle.
- 2) When executing the program, limit distractions and focus the user's attention on the program's output.
- 3) Author puzzle programs with memorable segments.
- 4) Provide a challenge without being tricky.

...





**1) Limit the editable  
dimensions of the puzzle.**



# Editable Dimensions



The screenshot shows the Looking Glass software interface. The top menu bar includes 'World', 'Edit', 'Project', 'Window', and 'Help'. Below the menu is a toolbar with icons for 'Create or Open Project', 'Save', 'Undo', 'Redo', 'Play', 'Play & Explore', 'Find A Remix', 'Share as: World', 'Remix', and 'Te'. The main window is titled '\*alien-and-ufo.lgp - Looking Glass'. The left panel shows a 3D scene with a tree, a UFO, and an alien. Below the scene is a dropdown menu for 'alien'. The right panel is titled 'Scene My Story' and contains a 'custom action My Story' editor. The editor shows a sequence of actions: 'Note' (Your code has been scrambled! Unscramble the code to make your animation work correctly.), 'flying saucer fly away', 'flying saucer turn RIGHT 2.0 rotations duration 2.0 seconds', 'alien walk 0.5 meters', 'Do together' (Drop action here.), 'flying saucer shake', 'flying saucer beam up thing: alien', 'Repeat 2 times' (Drop action here.), 'loop' (alien kick, flying saucer fly above thing: alien). The 'duration 2.0 seconds' value is highlighted with a red box and a mouse cursor. The 'alien kick' action is highlighted with a red box and a mouse cursor. The 'Repeat 2 times' block is highlighted with a red box and a mouse cursor. The 'alien kick' action is highlighted with a red box and a mouse cursor. The 'alien kick' action is highlighted with a red box and a mouse cursor. The 'alien kick' action is highlighted with a red box and a mouse cursor.

**Change Values**

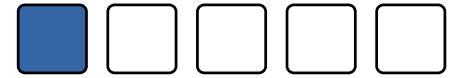
**Moves**

**Insertions**

**Deletions**




# Limit the Possibilities



Statement  
Bin

\*alien-and-ufo.lgp - Looking Glass



Play Correct Play Mine

Use all of these actions to put the animation back in the correct order.

Statement Bin

flying saucer turn RIGHT 2.0 rotations du

Repeat 2 times

loop

flying saucer fly above alien

alien walk 0.5 meters

flying saucer beam up alien

flying saucer fly away

Do together

alien kick


flying saucer shake

custom action My Story

Undo Redo Clear Done X

Drop action here.





**2) When executing the program, limit distractions and focus the user's attention on the program's output.**



# Play Window



\*alien-and-ufo.lgp - Looking Glass

custom action My Story

Undo Redo Clear Done

alien walk 0.5 meters

Repeat 2 times

alien kick

flying saucer shake

loop

flying saucer fly above alien

flying saucer turn RIGHT 2.0 rotations duration 2.0 seconds

Play Correct Play Mine

Use all of these actions to put the animation back in the correct order.

Do together

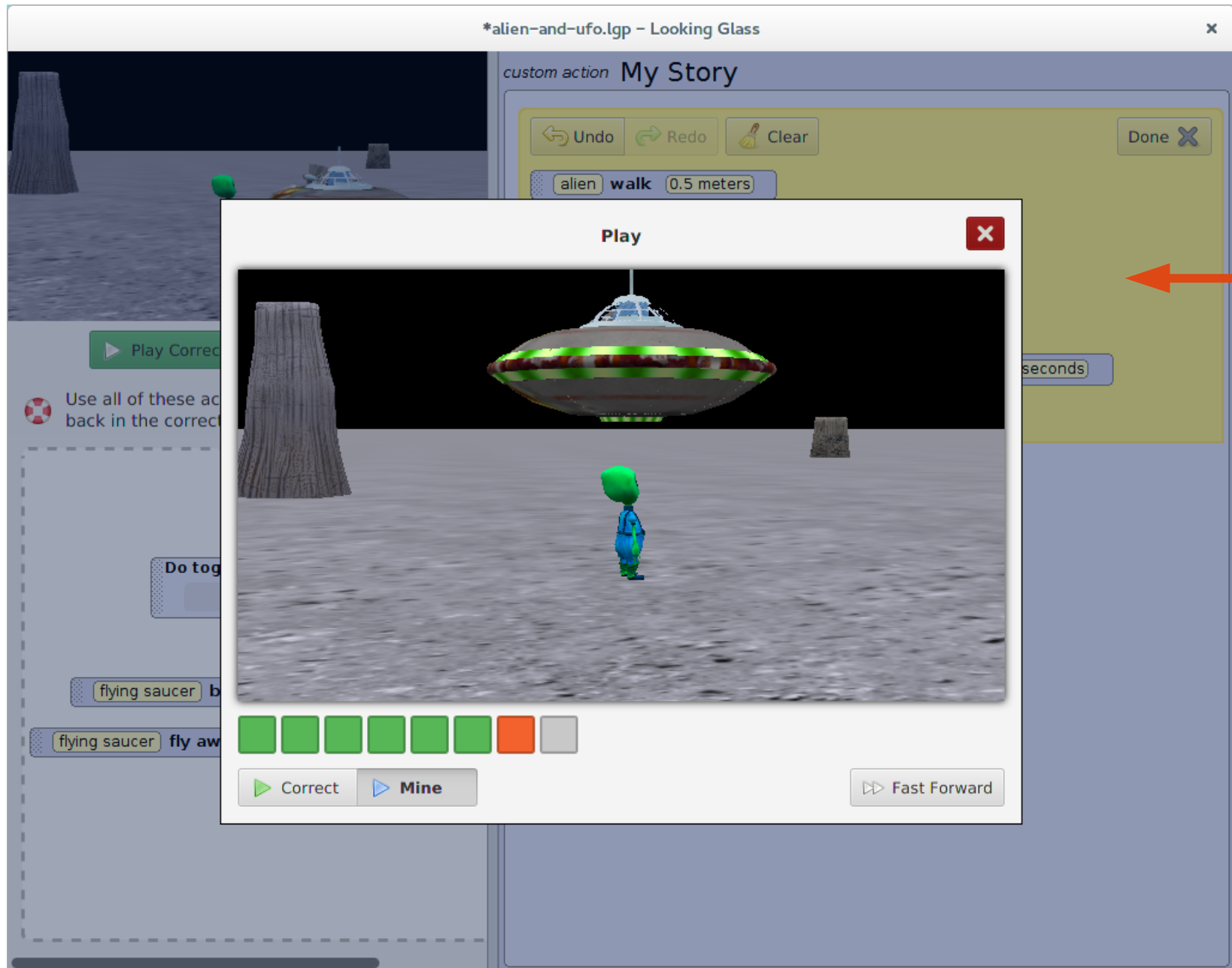
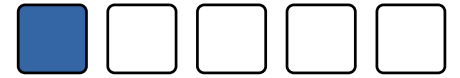
flying saucer beam up alien

flying saucer fly away

Play Fast Forward Full Screen



# Play Overlay







**3) Author puzzle programs  
with memorable segments.**



# Hard to Remember Output ■■■■

Monkey Business

Brown Monkey turn to face Grey Monkey

Grey Monkey turn to face Brown Monkey

Repeat 2 times

Brown Monkey talk

Grey Monkey talk

loop

Brown Monkey move UP 0.5 meters duration 0.5 seconds animationStyle BEGIN ABRUPTLY AND END GENTLY

Brown Monkey move DOWN 0.5 meters duration 0.5 seconds animationStyle BEGIN GENTLY AND END ABRUPTLY

Repeat 3 times

Grey Monkey say "EEK EEK"

Brown Monkey turn RIGHT 4.0 rotations

Grey Monkey hop side to side

loop

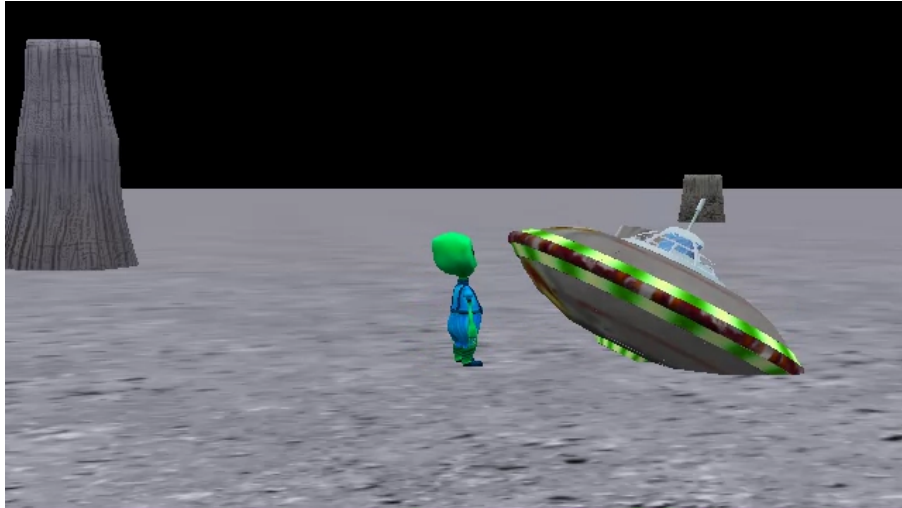
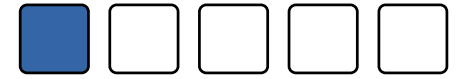
Grey Monkey lay down



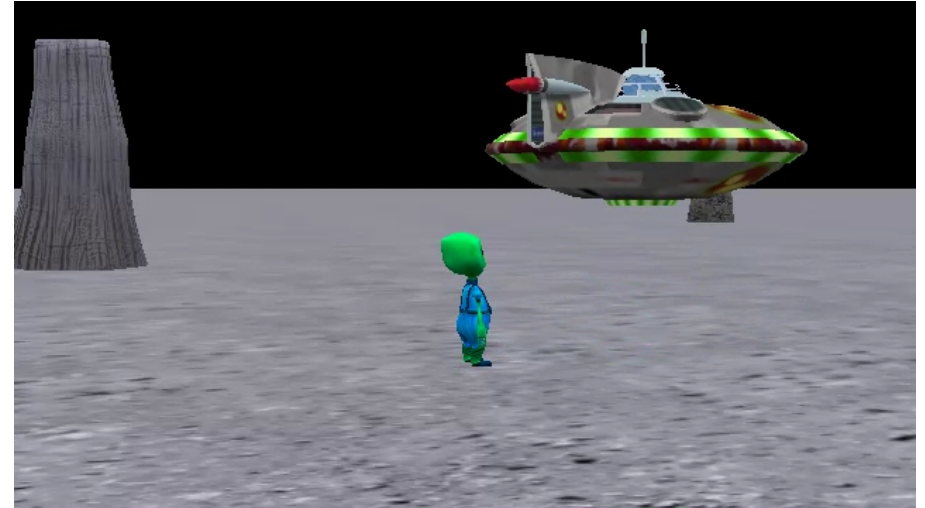
Monkey Business



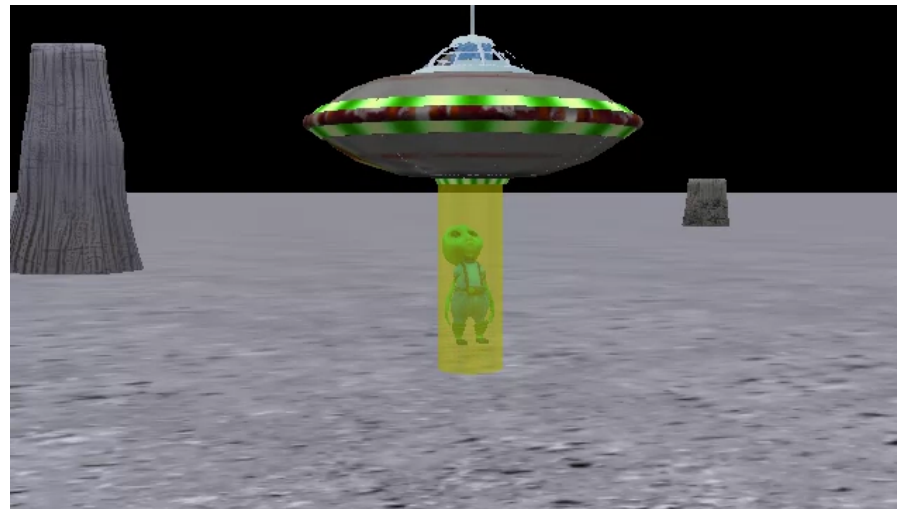
# Memorable Segments



Segment I – The alien repairs the flying saucer.



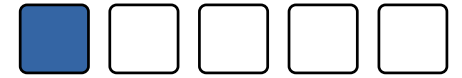
Segment II – The flying saucer starts up.



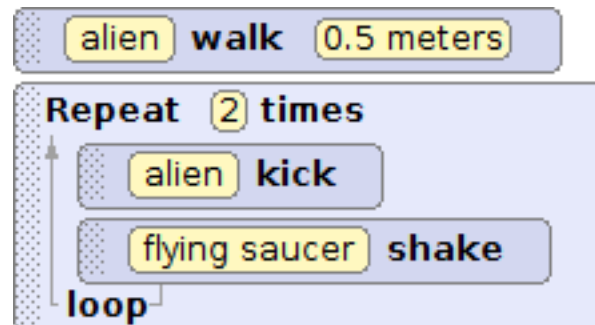
Segment III - Alien drives flying saucer.



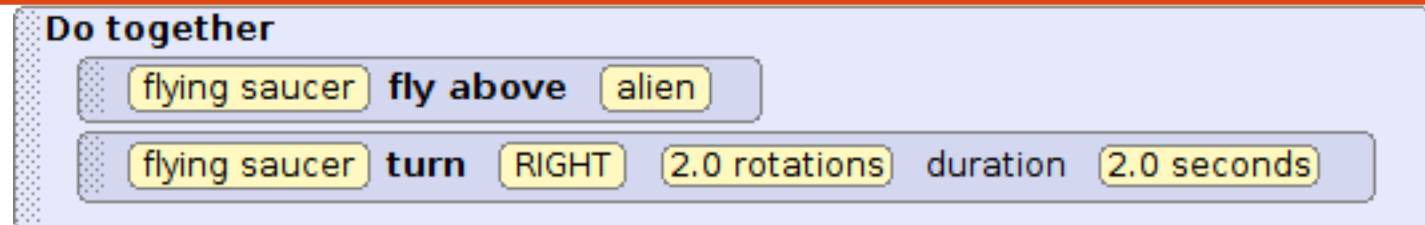
# Memorable Segments



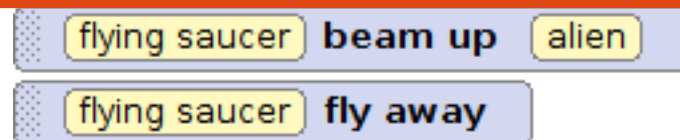
Segment I



Segment II



Segment III







**4) Provide a challenge  
without being tricky.**





# Tricky & Challenging

“It was tricky, but not harder.”

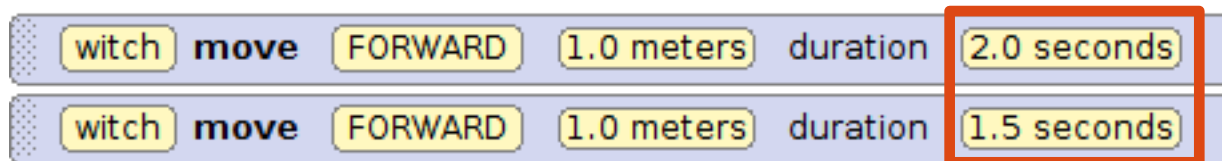
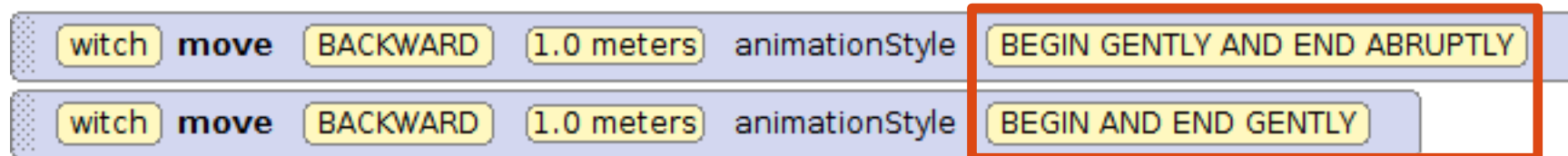
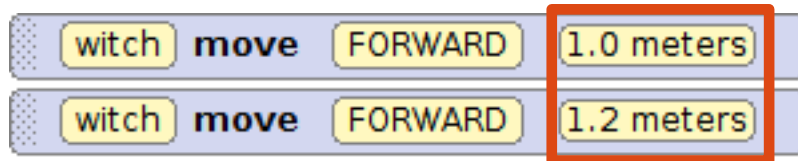
“I thought this one was a little challenging, but I liked it!”



# Tricky

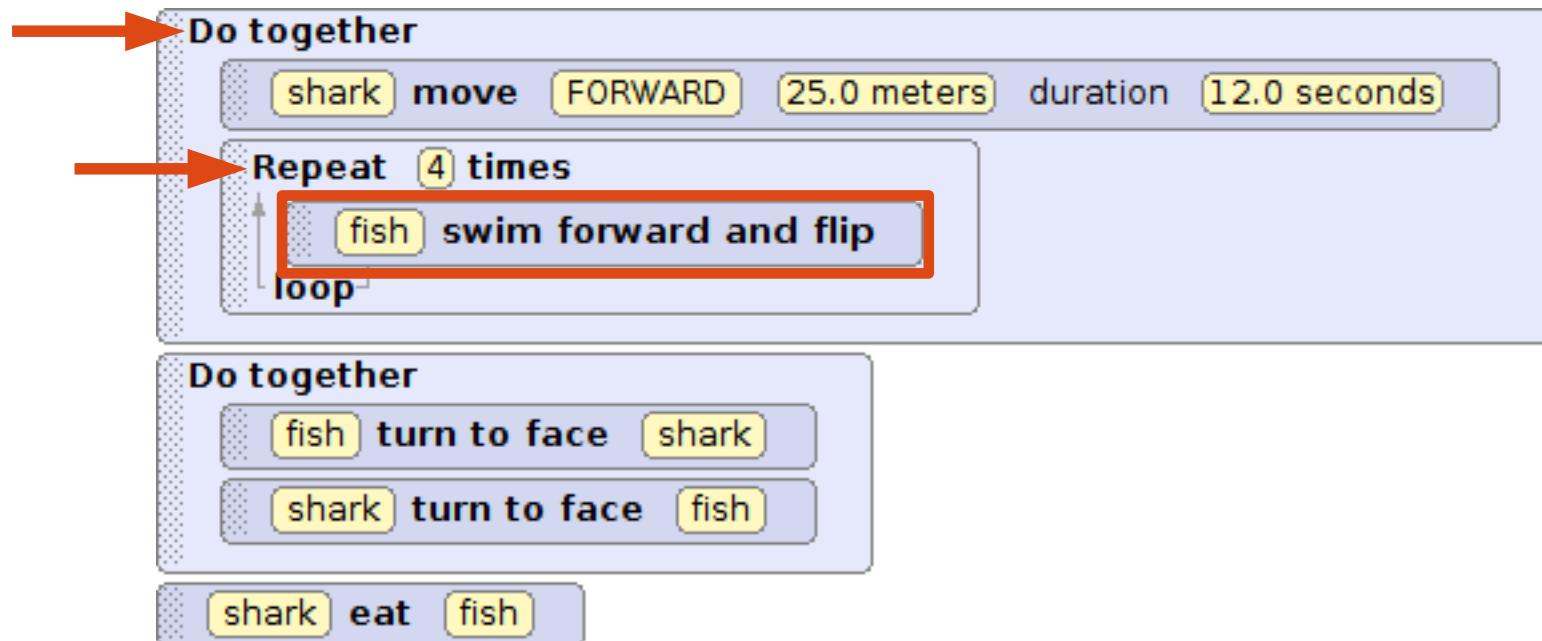
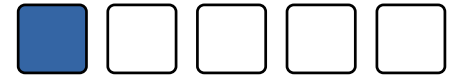


## Nearly Identical Statements





# Challenging





- 
- **Lessons Learned**
  - **Programming Completion**  
**Puzzle Effectiveness**





# Summative Evaluation

- 27 participants
  - 12 Female, 15 Male
  - Average Age: 11.59
  - Minimal Programming Experience (< 3 hours)
- 2 hours



# Independent Learning: Tutorials

The image shows the Looking Glass software interface with a tutorial window open. The main window, titled "two-walruses.lgp - Looking Glass (1)", displays a 3D scene with two walrus characters in a snowy environment. The left sidebar contains a dropdown menu with "baby walrus" selected, and a list of actions for "baby walrus's Actions". The right sidebar shows a "Scene" tab with "My Story" as the custom action. The tutorial window, titled "Tutorial", has a header "Baby walrus say weeeeeeeee!" and a main area with the instruction "Drag in the following action: baby walrus say 'weeeeeeeeeee!'". The tutorial window also includes a "Next »" button and a progress indicator showing 2/9.

two-walruses.lgp - Looking Glass (1)

World Edit Help

Create or Open Project Save Undo Redo Play

Scene My Story

custom action My Story

Drop action here.

baby walrus

baby walrus's Actions

+ new action...

Custom Actions (2)

edit baby walrus shake head

edit baby walrus turn back to normal color

say, think

? baby walrus say text

? baby walrus think text

position

? baby walrus move direction amount

? baby walrus move toward target amount

? baby walrus move away from target amount

? baby walrus move to target

? baby walrus place spatial relation target

Tutorial

Baby walrus say weeeeeeeee!

two-walruses.lgp - Looking Glass

Scene My Story

custom action My Story

baby walrus say text

text: "Hello."

Custom Text...

Drag in the following action:

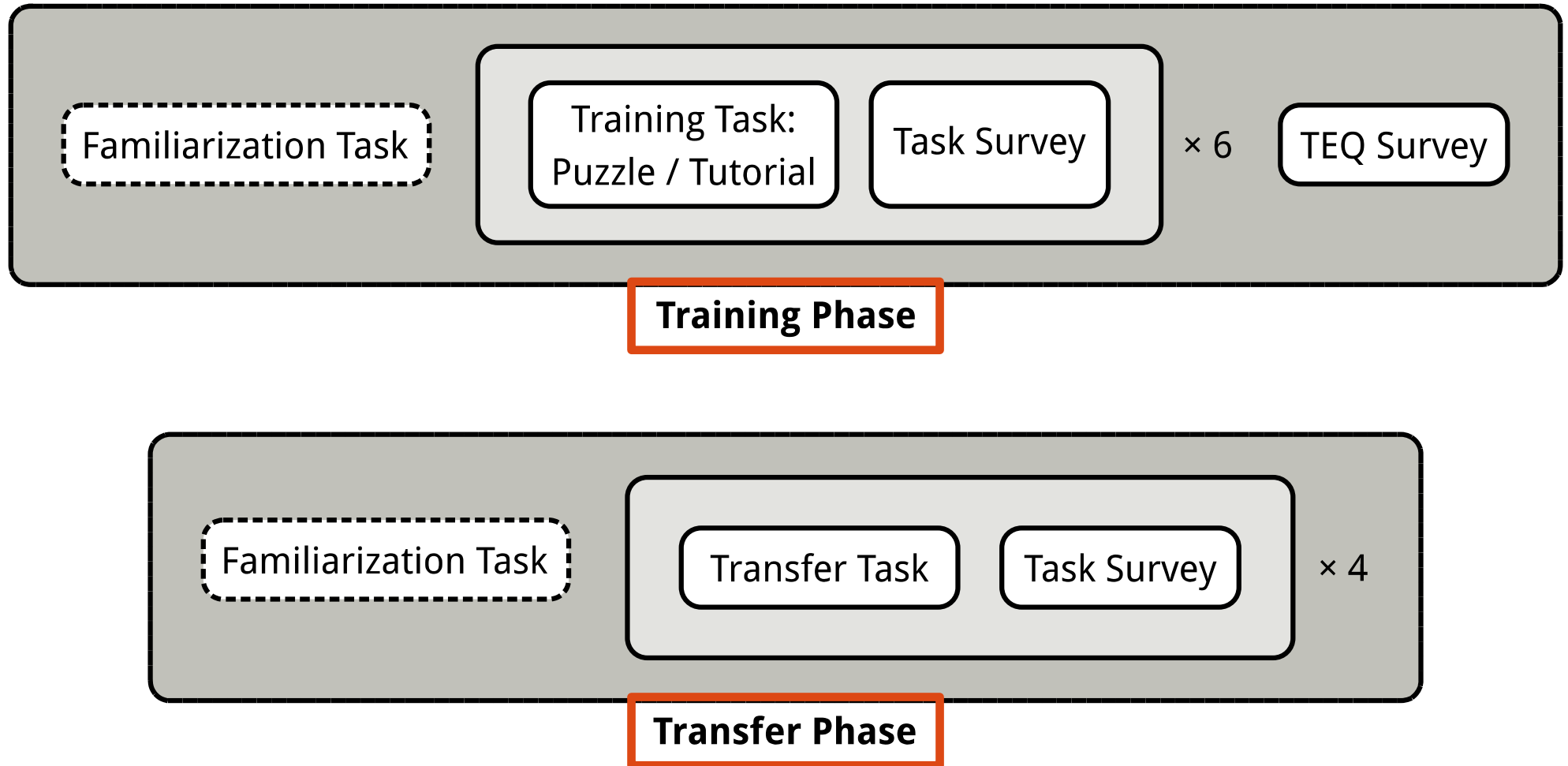
**baby walrus say "weeeeeeeeeee!"**

Next »

2/9

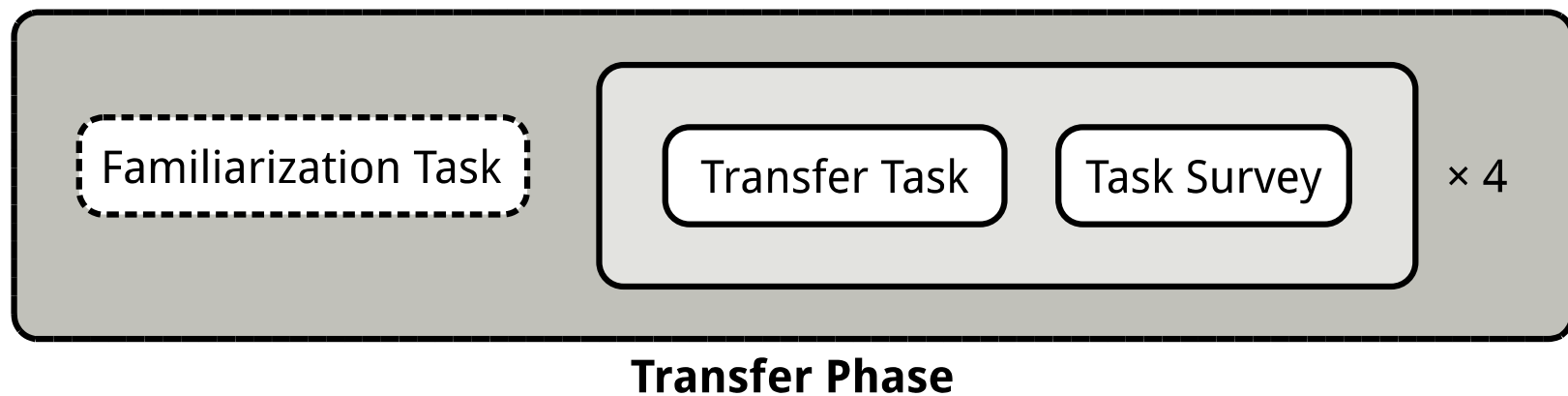
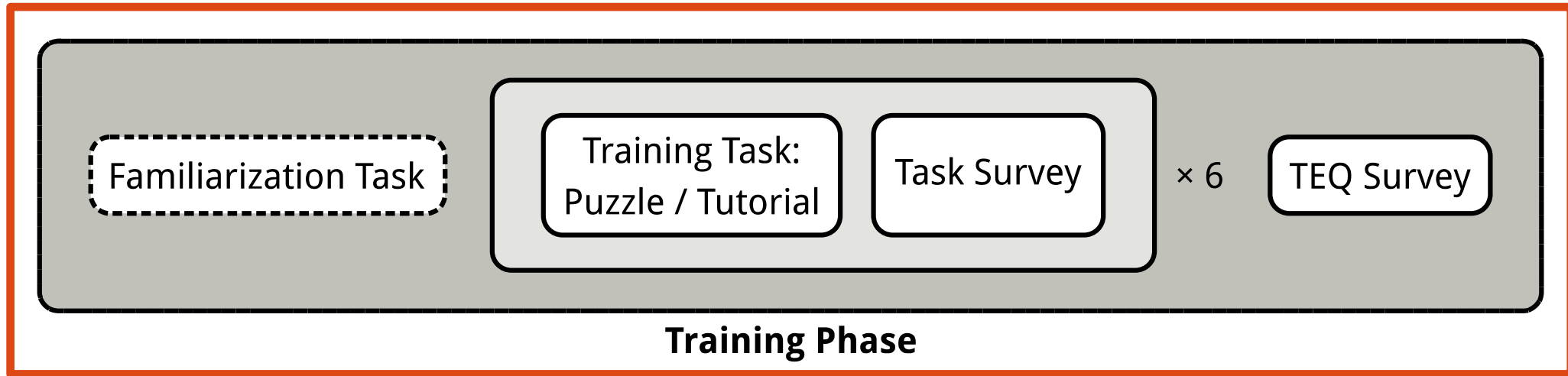


# Study Design





# Study Design





# Training Phase



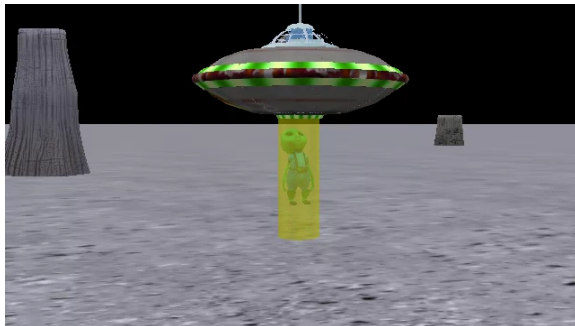
**1. Sequential**



**2. Repeated**



**3. Parallel**



**4. Repeated & Parallel**



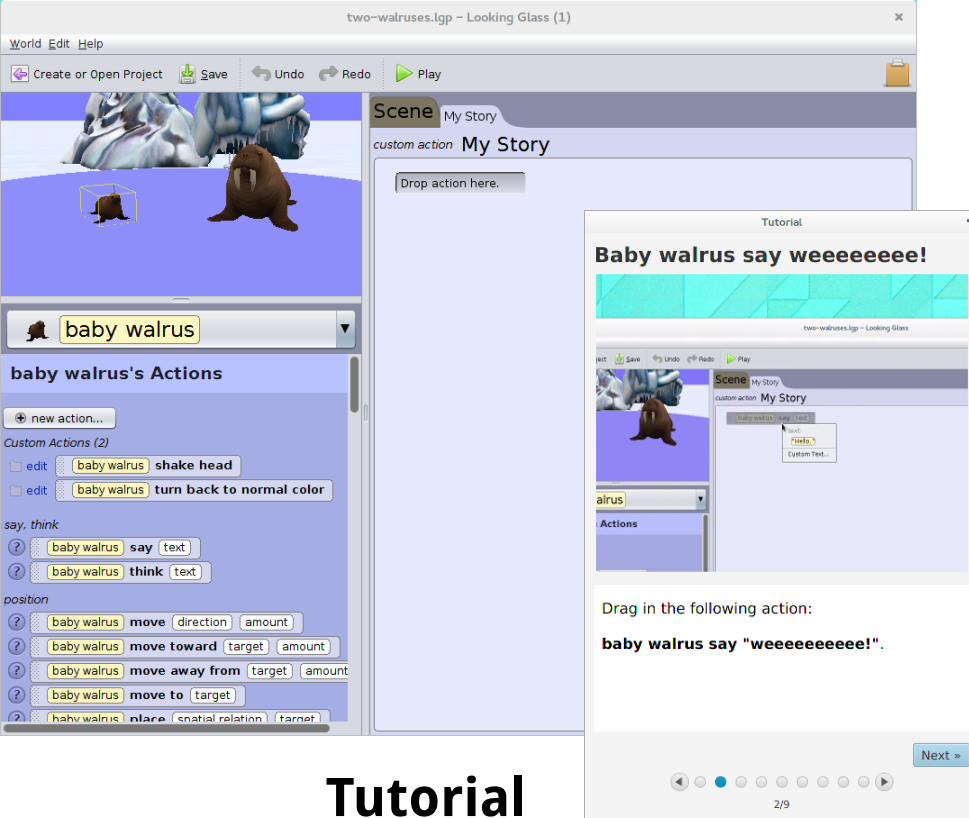
**5. Parallel {  
Repeated  
}**



**6. Repeated {  
Parallel  
}**



# Training Task



two-walruses.lgp - Looking Glass (1)

World Edit Help

Create or Open Project Save Undo Redo Play

Scene My Story

custom action My Story

Drop action here.

baby walrus

baby walrus's Actions

+ new action...

Custom Actions (2)

edit baby walrus shake head

edit baby walrus turn back to normal color

say, think

baby walrus say text

baby walrus think text

position

baby walrus move direction amount

baby walrus move toward target amount

baby walrus move away from target amount

baby walrus move to target

baby walrus place spatial relation target

Tutorial

Baby walrus say weeeeeeeee!

two-walruses.lgp - Looking Glass

Scene My Story

custom action My Story

Drop action here.

Drag in the following action:

baby walrus say "weeeeeeeeeee!".

Next »

2/9

**Tutorial**



\*two-walruses.lgp - Looking Glass (0)

custom action My Story

Undo Redo Clear Done X

Drop action here.

Play Correct Play Mine

Use all of these actions to put the animation back in the correct order.

baby walrus set color green

baby walrus say "uh oh"

baby walrus turn RIGHT 8.0 rotations duration

mommy walrus shake head

baby walrus turn back to normal color

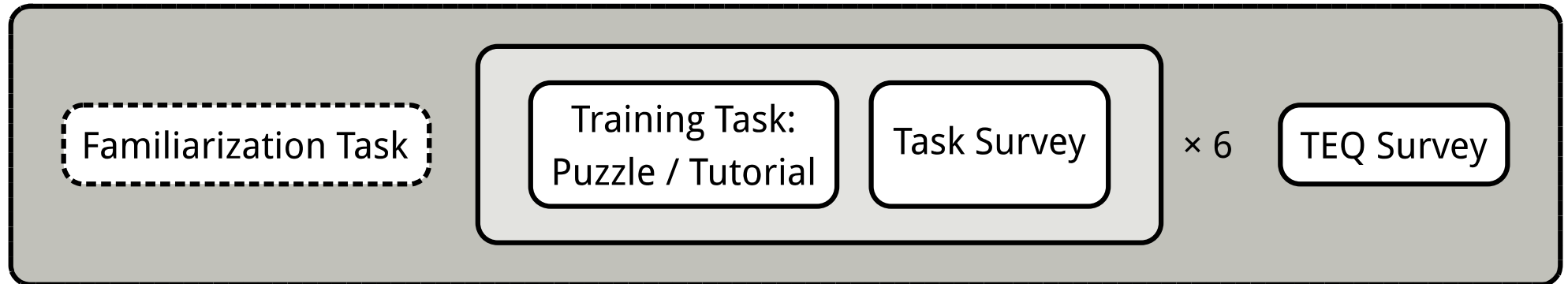
baby walrus say "weeeeeeeeeee!!"

mommy walrus turn to face baby walrus

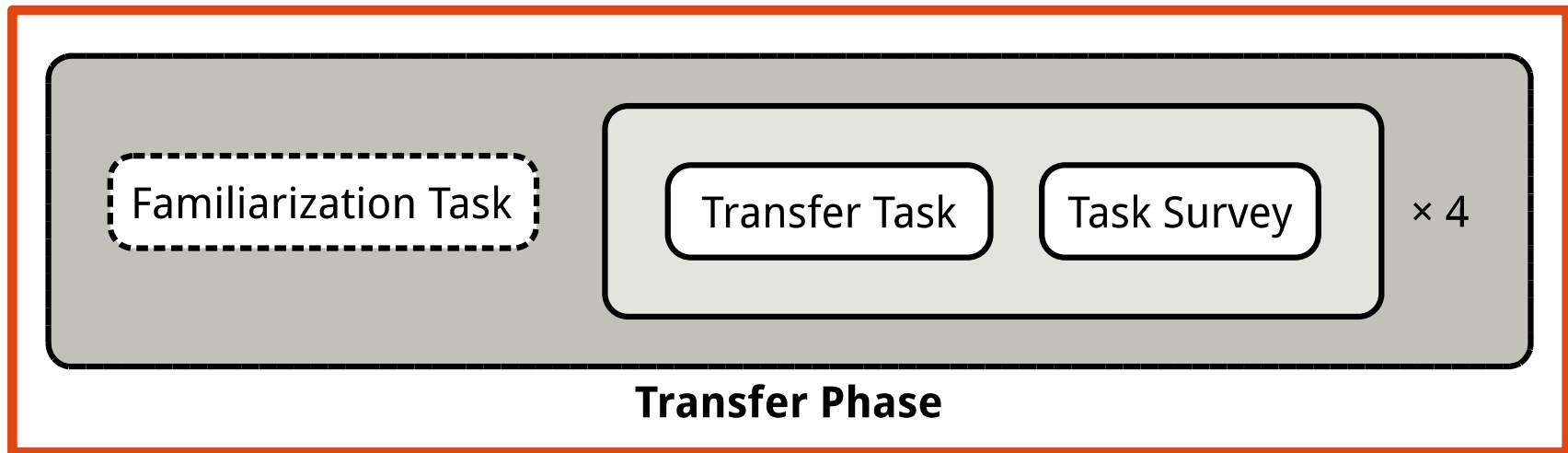
**Puzzle**



# Study Design



**Training Phase**



**Transfer Phase**



# Transfer Phase



**Repeated**



**Parallel**



**Parallel {  
Repeated  
}**



**Repeated {  
Parallel  
}**



# Transfer Task

tropical-food-hunt.lgp - Looking Glass (0)

World Edit Help

Create or Open Project Save Undo Redo Play

Scene My Story

custom action My Story

Instructions

- All of these actions are already in the correct order.
- You may only add up to 3 additional actions.
- Before making any changes, first play the animation.

Instructions

- The monkey should jump forward twice.

monkey jump forward

monkey say "yummy! yummy!" more..

Instructions

- The monkey should look up and climb twice.
- Then the monkey should shake the tree and the tree should drop a coconut, twice.

monkey look up

monkey climb

monkey shake thing : palm tree

palm tree drop coconut

monkey move DOWN 2.0 meters duration 0.5 seconds more..

other

monkey straighten out joints

Action Ordering Boxes

Do actions in order, one at a time

Do in order

Do actions at the same time

Do together

Repeat actions multiple times

Repeat number times loop


Repeat while condition loop

Do one action, otherwise do another action

Task Instructions

Try your best to make your animation match the correct animation below.

- You may only add **up to 3** additional actions or action ordering boxes.
- All of the actions that are currently there are in the **correct** order already.



Play

Correct Animation



# Completed Transfer Task

**Instructions**

- All of these actions are already in the correct order.
- You may only add up to 3 additional actions or action ordering boxes.
- Before making any changes, first play the animation.

**Instructions**

- The monkey should jump forward twice.

monkey jump forward

monkey say "yummy! yummy!" more...

**Instructions**

- The monkey should look up and climb twice.
- Then the monkey should shake the tree and afterwards the palm tree should drop a coconut, twice.

monkey look up

monkey climb

monkey shake thing : palm tree

palm tree drop coconut

monkey move DOWN 2.0 meters duration 0.5 seconds more...

**Initial**

**Instructions**

- All of these actions are already in the correct order.
- You may only add up to 3 additional actions or action ordering boxes.
- Before making any changes, first play the animation.

**Instructions**

- The monkey should jump forward twice.

Repeat 2 times

monkey jump forward

monkey say "yummy! yummy!" more...

loop

**Instructions**

- The monkey should look up and climb twice.
- Then the monkey should shake the tree and afterwards the palm tree should drop a coconut, twice.

Repeat 2 times

monkey look up

monkey climb

loop

Repeat 2 times

monkey shake thing : palm tree

palm tree drop coconut


loop

monkey move DOWN 2.0 meters duration 0.5 seconds more...

**Completed**



# Research Questions:

- 
- 1) Do puzzles require a different time and mental investment compared to tutorials?
  - 2) Do puzzle users show more evidence of learning compared to tutorial users?
  - 3) Are puzzles more motivating than tutorials?

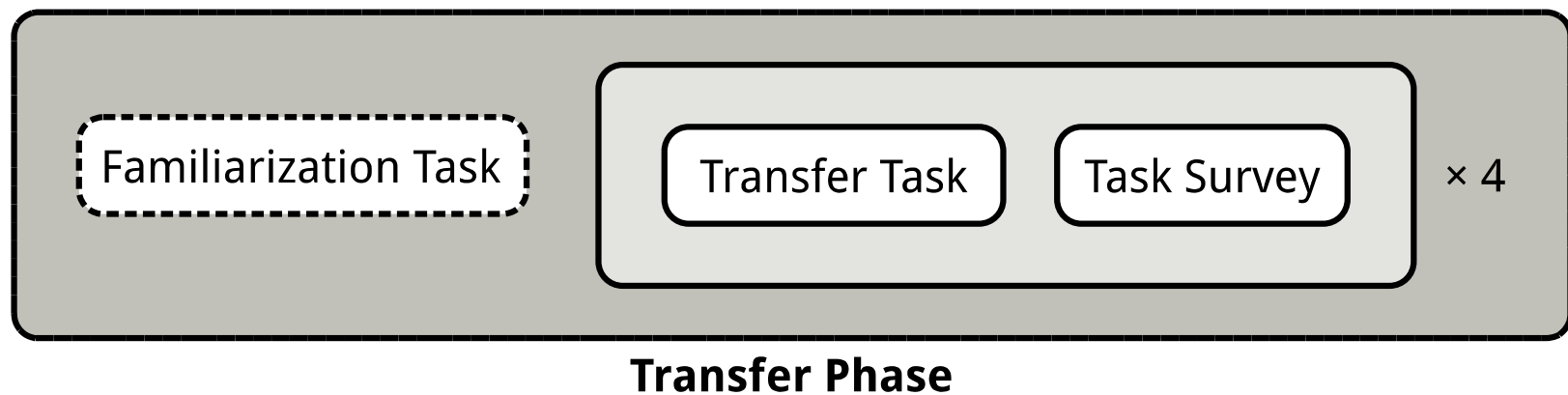
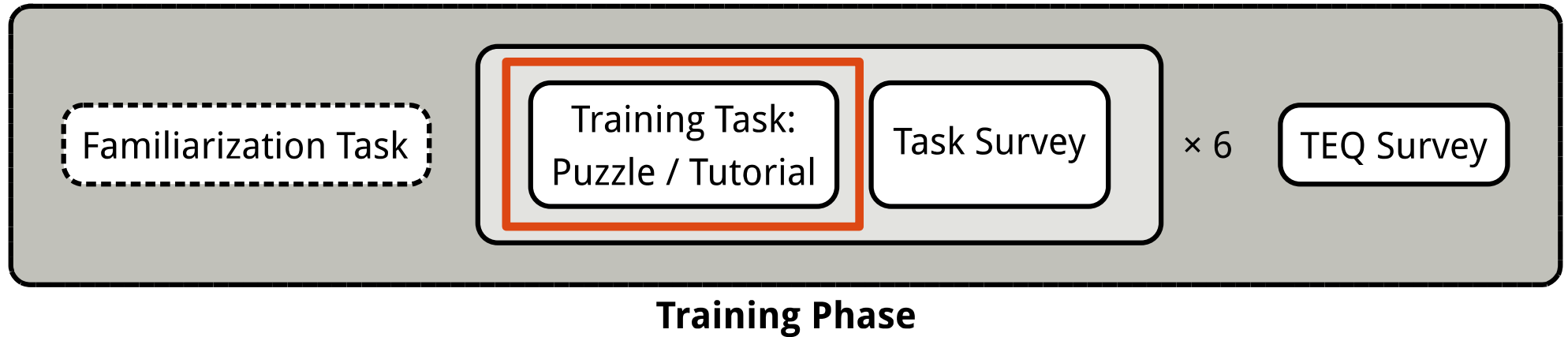




**1) Do puzzles require a different time and mental investment compared to tutorials?**

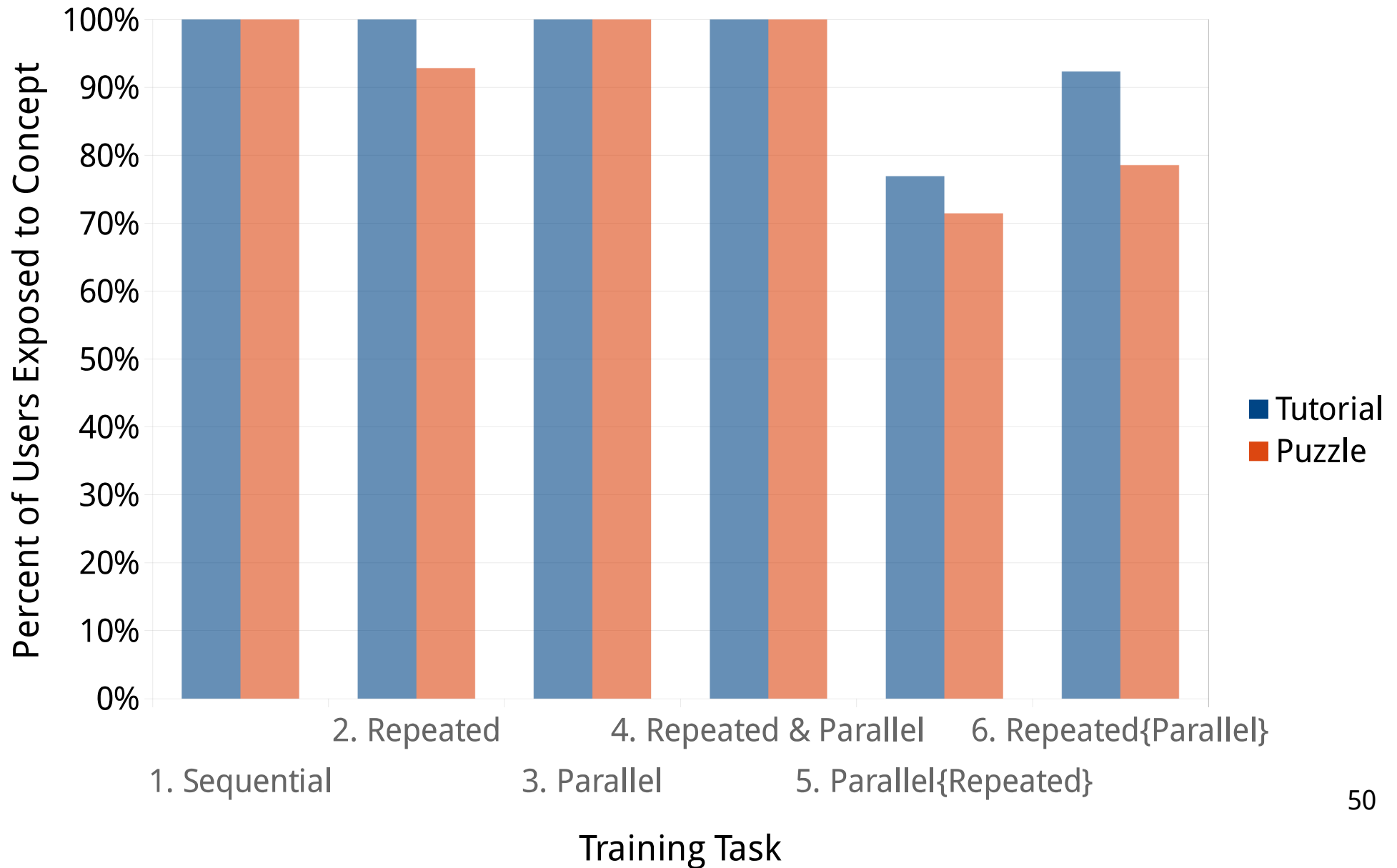


# Study Design





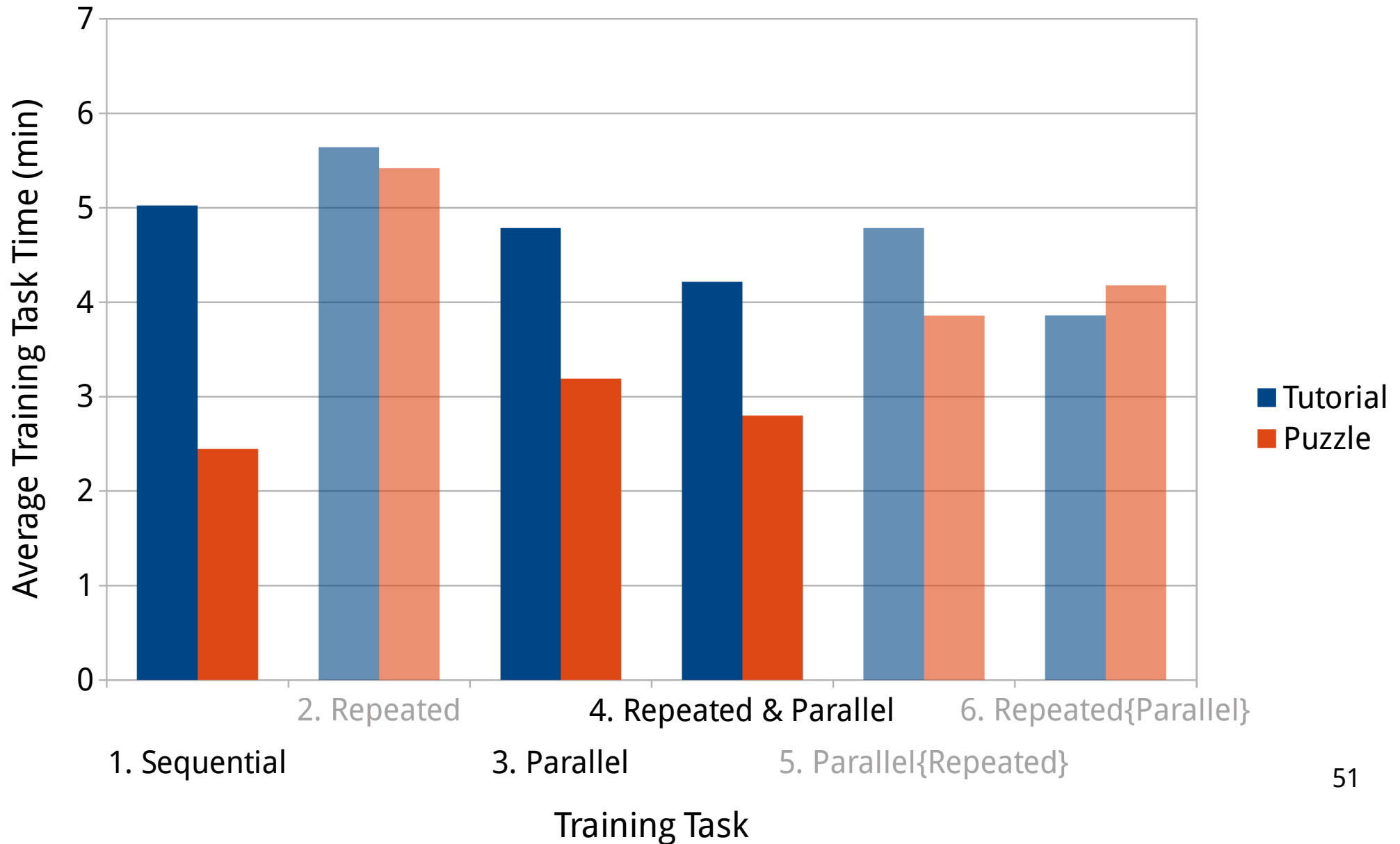
# Percent of Users Exposed to Programming Concept





# Average Training Task Time

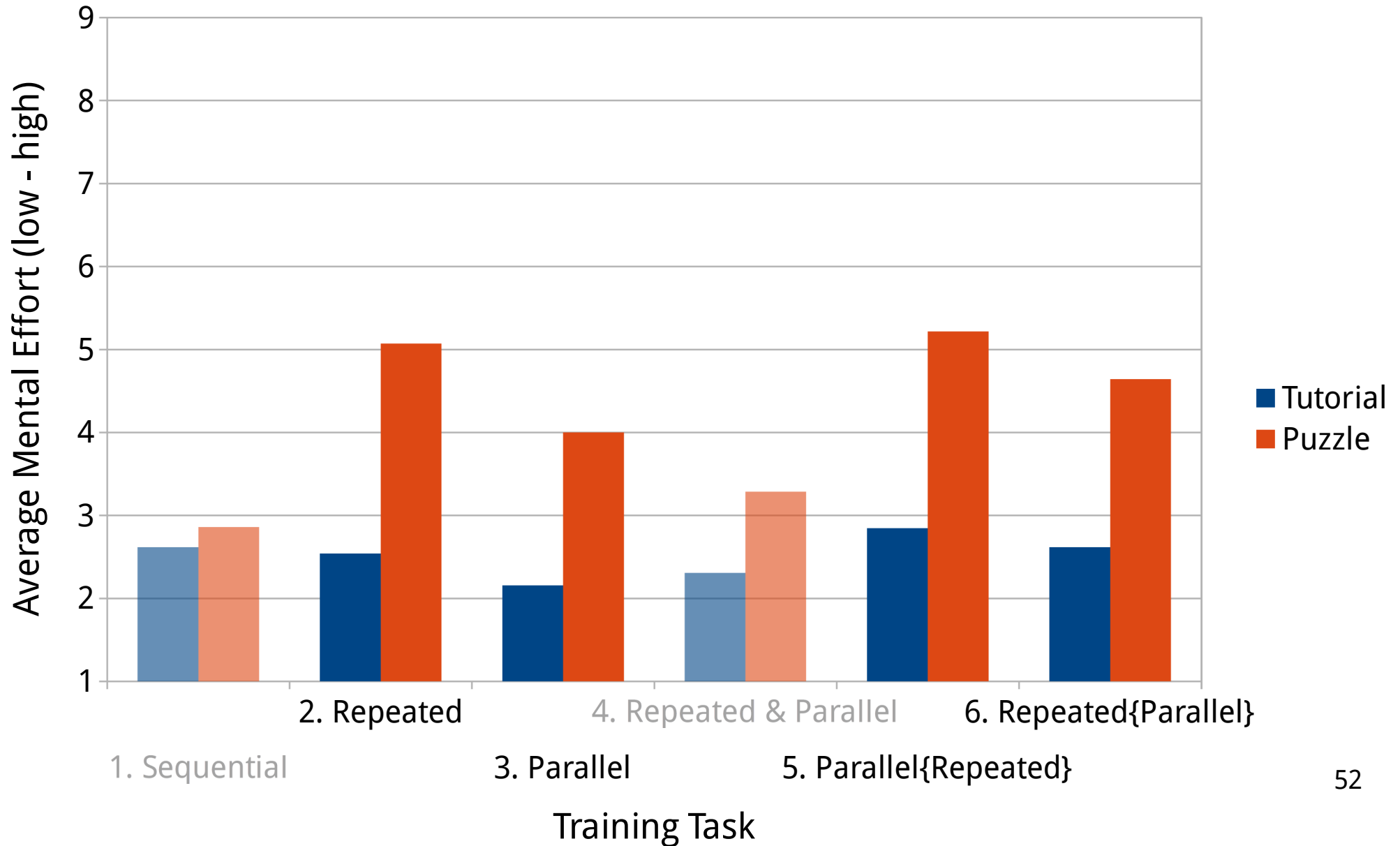
$p < .001$






# Average Training Task Mental Effort

$p < .05$



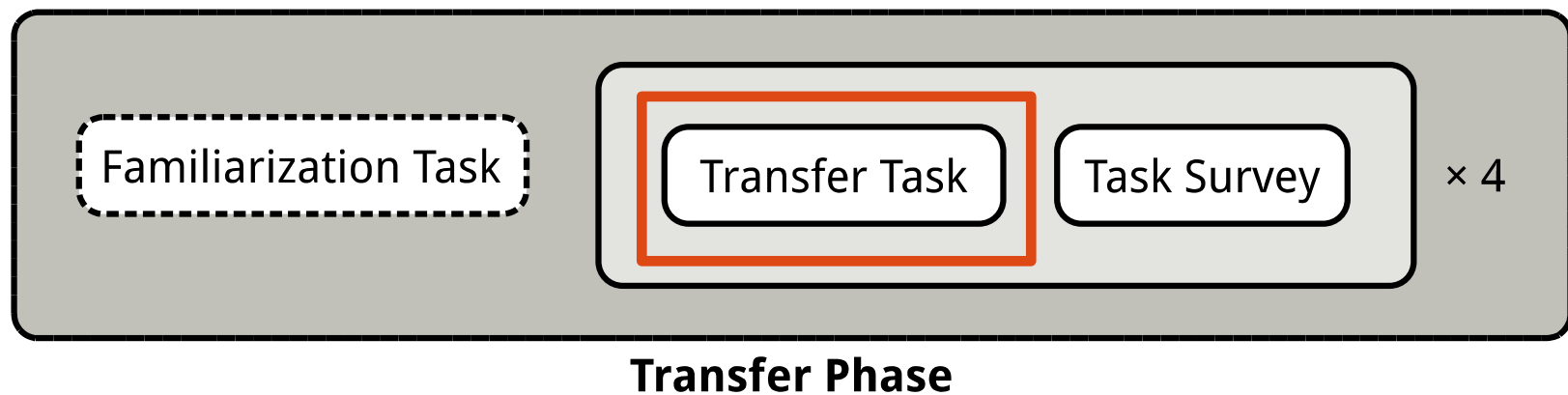
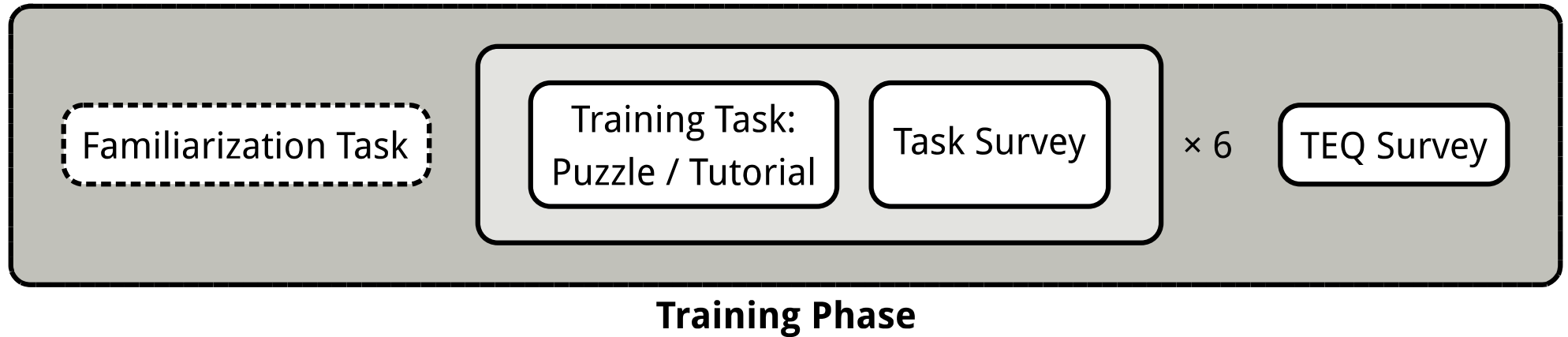




**2) Do puzzle users show more evidence of learning compared to tutorial users?**



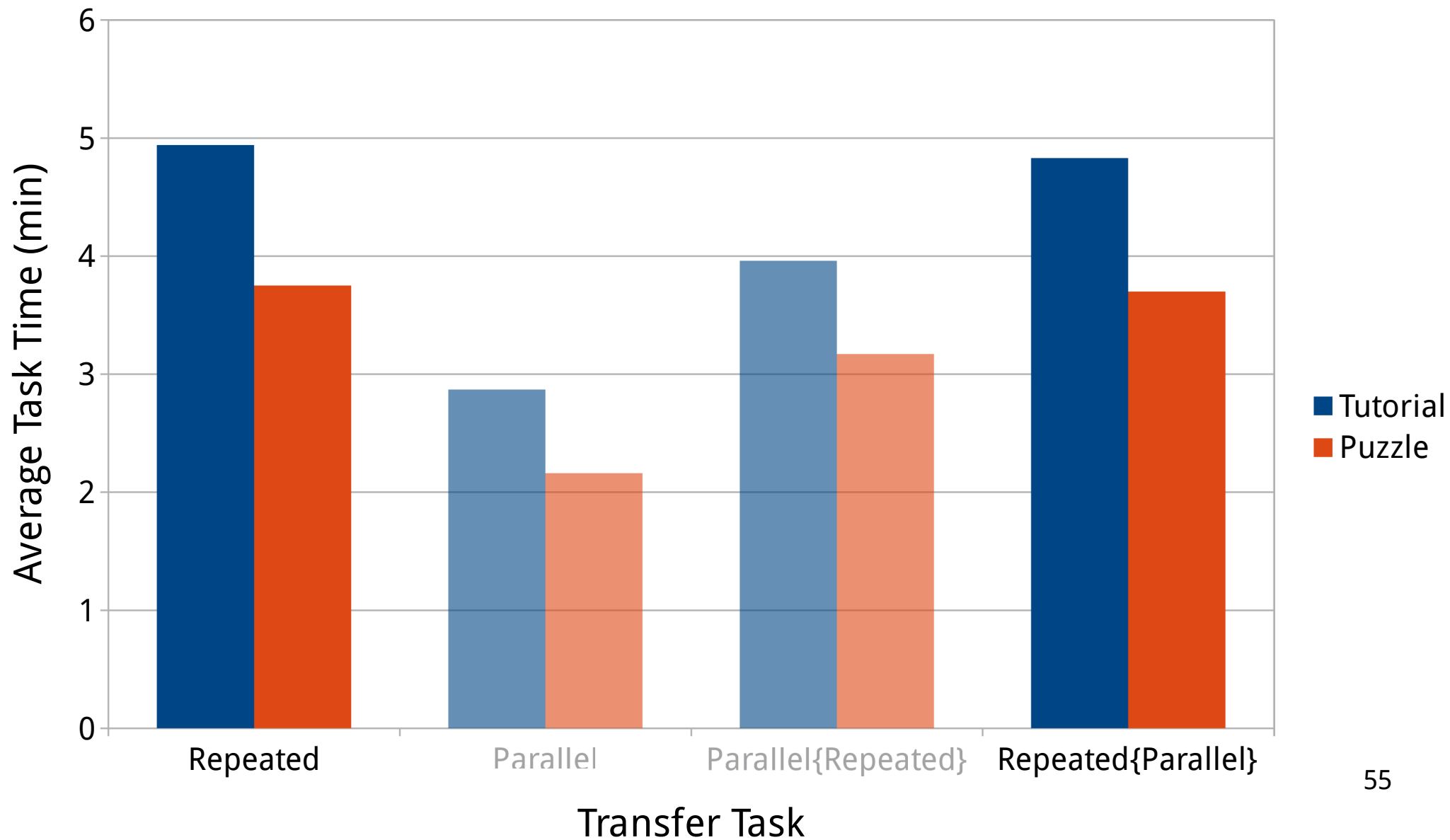
# Study Design





# Average Transfer Task Time

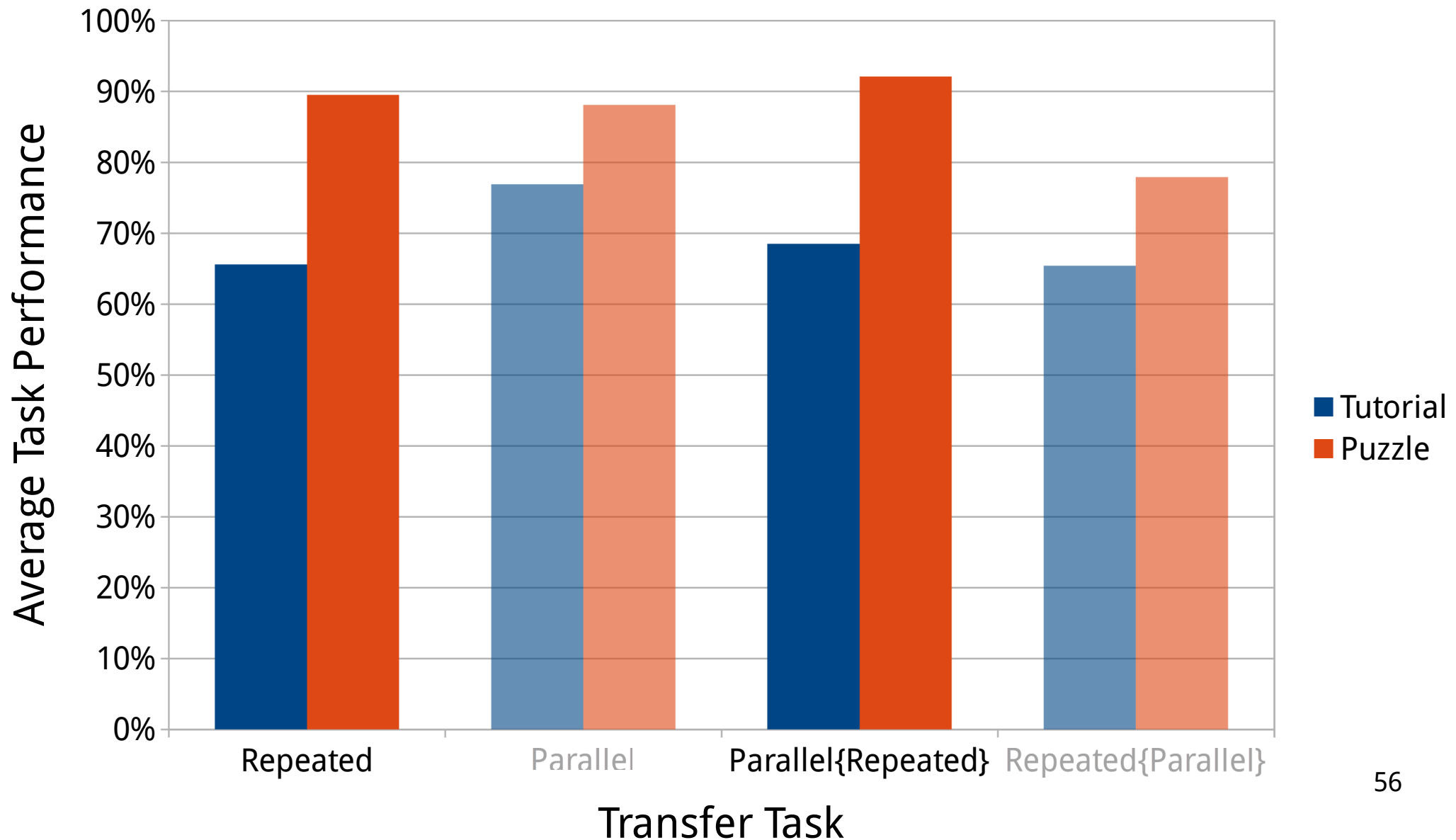
$p = .06$





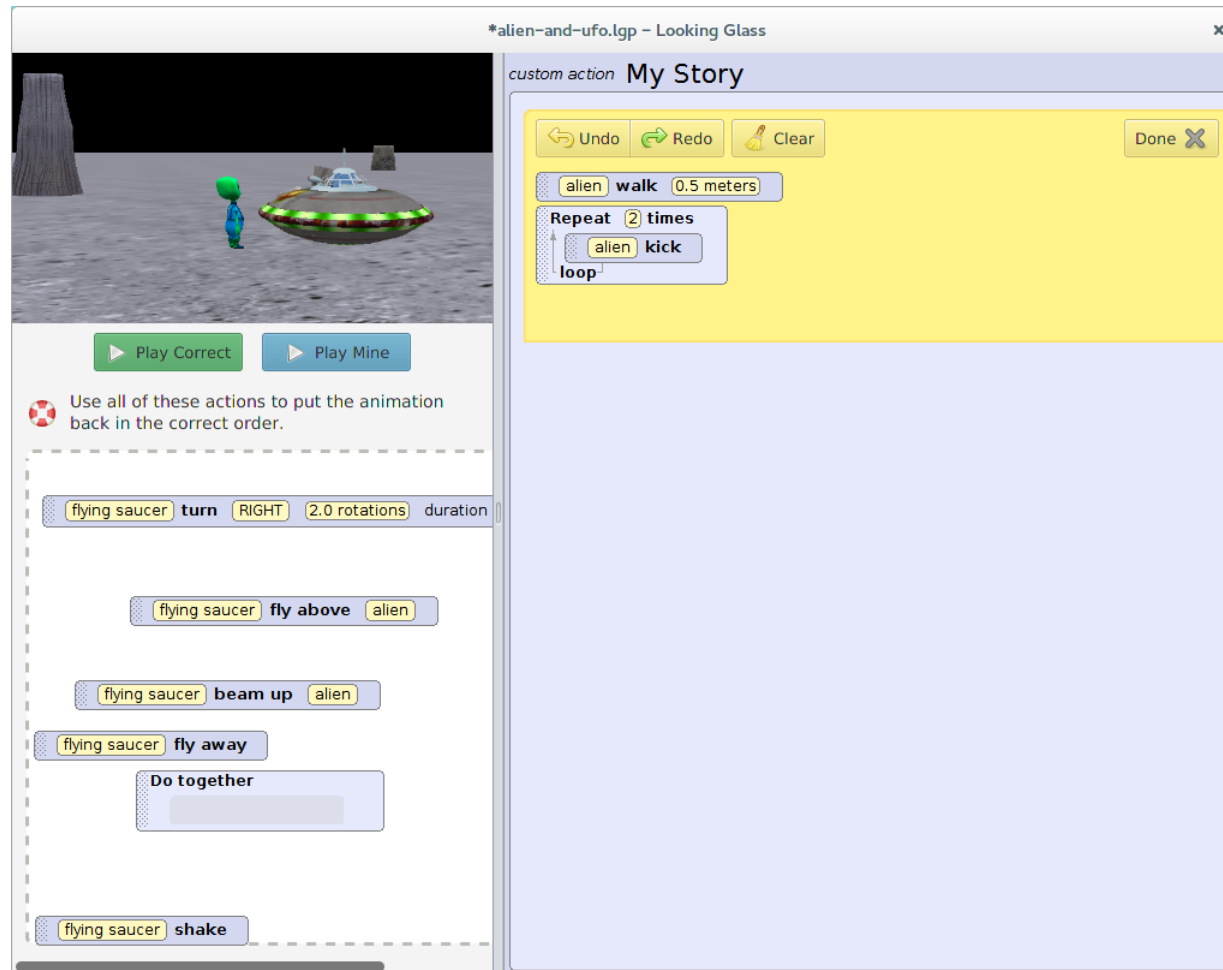
# Average Transfer Task Performance

$p < .05$





# Summary



Puzzle users performed 26% better on transfer tasks while requiring 23% less training time. 57





# Future Work

- Completion Problems
  - Paired with worked examples
- Distractors
  - Common in puzzle-like systems
  - Impact on completion problem effect



# Questions

**Kyle J. Harms**  
**Washington University in St. Louis**  
**kyle.harms@wustl.edu**



**Lookingclass**

<https://lookingglass.wustl.edu>



Washington University in St. Louis