UNIT 4 ACTIVITY

DEBUG IT!

SUGGESTED TIME
15–30 MINUTES

OBJECTIVES

By completing this activity, students will:

- investigate the problem and find a solution to five debugging challenges
- explore a range of concepts (conditionals, operators, and data) through the practices of testing and debugging

ACTIVITY DESCRIPTION

- ❑ Optionally, have the Unit 4 Debug It! handout available to guide students during the activity.
- ❑ Help students open the Debug It! programs from the Unit 4 Debug It! studio or by following the project links listed on the Unit 4 Debug It! handout. Encourage students to click on the "Look Inside" button to investigate the buggy program, tinker with problematic code, and test possible solutions.
- ❑ Give students time to test and debug each Debug It! challenge. Optionally, have students use the remix function in Scratch to fix the bugs and save corrected programs.
- ❑ Ask students to reflect back on their testing and debugging experiences by responding to the reflection prompts in their design journals or in a group discussion.
- ❑ Create a class list of debugging strategies by collecting students' problem finding and problem solving approaches.

RESOURCES

- ❑ Unit 4 Debug It! handout
- ❑ Unit 4 Debug It! studio
  http://scratch.mit.edu/studios/475634

REFLECTION PROMPTS

- What was the problem?
- How did you identify the problem?
- How did you fix the problem?
- Did others have alternative approaches to fixing the problem?

REVIEWING STUDENT WORK

- Were students able to solve all five bugs? If not, how might you clarify the concepts expressed in the unsolved programs?
- What different testing and debugging strategies did students employ?

NOTES

- This activity provides an opportunity to check in with students who might need some additional attention or support, particularly around the concepts of conditionals (e.g., if), operators (e.g., arithmetic, logical), and data (e.g., variables, lists).

NOTES TO SELF

❑ ________________________

❑ ________________________

❑ ________________________

❑ ________________________

DEBUG IT!

HELP! CAN YOU DEBUG THESE FIVE SCRATCH PROGRAMS?

In this activity, you will investigate what is going awry and find a solution for each of the five Debug It! challenges.

START HERE

- □ Go to the Unit 4 Debug It! Studio: http://scratch.mit.edu/studios/475634/
- □ Test and debug each of the five debugging challenges in the studio.
- □ Write down your solution or remix the buggy program with your solution.

□ **DEBUG IT! 4.1** http://scratch.mit.edu/projects/24271192

In this project, the "Inventory" list should be updated every time Scratch Cat picks up a new item. But Scratch Cat can only pick up the laptop. How do we fix the program?

□ **DEBUG IT! 4.2** http://scratch.mit.edu/projects/24271303

In this project, Scratch Cat gets 10 points for collecting Yellow Gobos and loses 10 points for colliding with Pink Gobos. But something isn't working. How do we fix the program?

□ **DEBUG IT! 4.3** http://scratch.mit.edu/projects/24271446

In this project, Scratch Cat is thinking of a number between 1 and 10. But something is wrong with the guess checking -- it doesn't work consistently. How do we fix the program?

□ **DEBUG IT! 4.4** http://scratch.mit.edu/projects/24271475

In this project, the "# of hits" display should increase by 1 every time the Scratch Cat is hit by a tennis ball. But the "# of hits" increases by more than 1 when Scratch Cat is hit. How do we fix the program?

□ **DEBUG IT! 4.5** http://scratch.mit.edu/projects/24271560

In this project, Scratch Cat is navigating a maze to get to the yellow rectangle. But Scratch Cat can walk through walls. How do we fix the program?

FEELING STUCK?

THAT'S OKAY! TRY THESE THINGS...

- □ Make a list of possible bugs in the program.
- □ Keep track of your work! This can be a useful reminder of what you have already tried and point you toward what to try next.
- □ Share and compare your problem finding and problem solving approaches with a neighbor until you find something that works for you!

FINISHED?

- + Add code commentary by right clicking on blocks in your scripts. This can help others understand different parts of your program!
- + Discuss your testing and debugging practices with a partner. Make note of the similarities and differences in your strategies.
- + Help a neighbor!

DEBUG IT! REFLECTIONS

NAME: ______________________________

RESPOND TO THE FOLLOWING REFLECTION PROMPTS USING THE SPACE PROVIDED BELOW OR IN YOUR DESIGN JOURNAL.

+ What was the problem?

+ How did you identify the problem?

+ How did you fix the problem?

+ Did others have alternative approaches to fixing the problem?