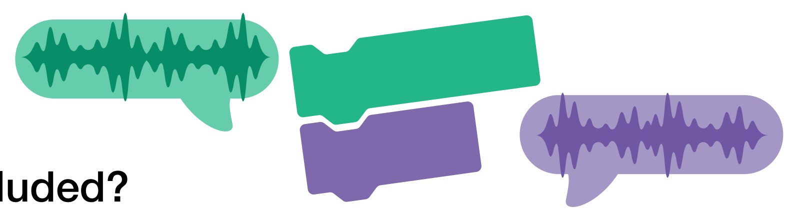
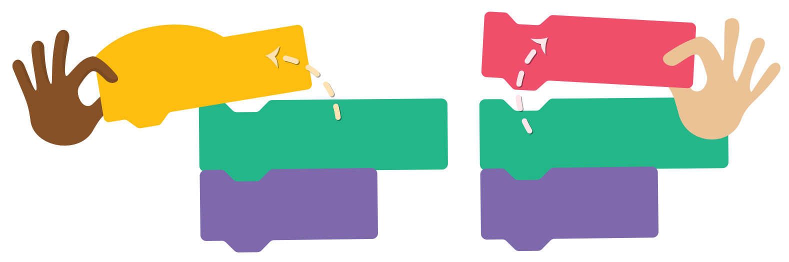
SCRATCH™ Debugging Strategies

Read Aloud



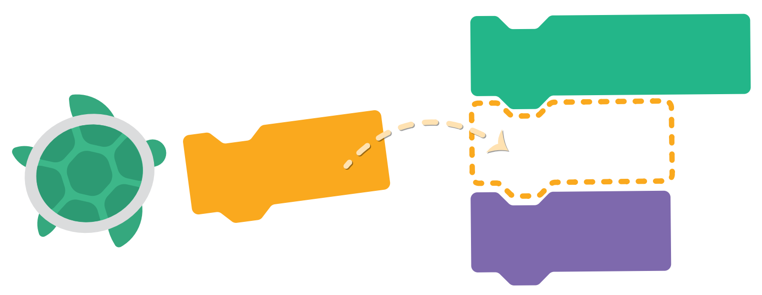
As you read your code aloud, think from the computer's perspective. Are you including steps that aren't there? Are your instructions clear? If something needs to be reset each time the program has run, are those instructions included?

Break It Down



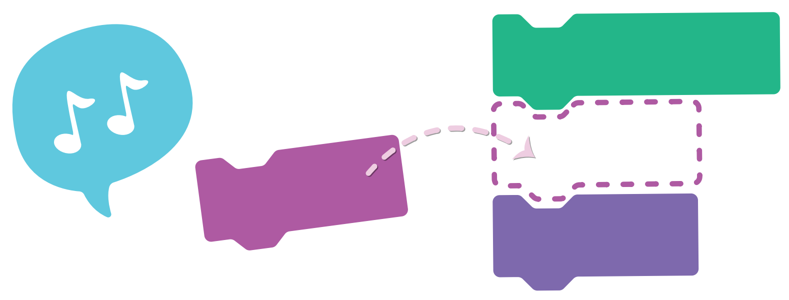
Separate the blocks into smaller chunks (or sequences), and click to see what each sequence does. Once the smaller sequences work as you expect, add them back into the main program. The process is called decomposition.

Slow It Down



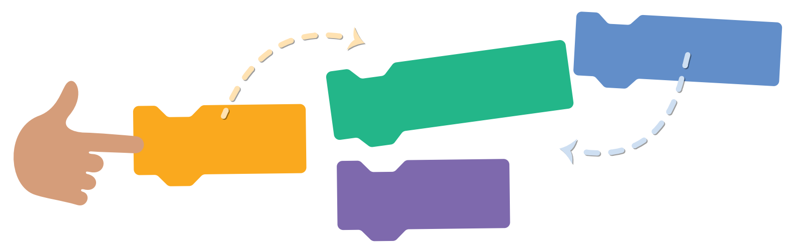
The computer runs your program so quickly it can be hard to follow with your eyes. Add temporary "wait" or "wait until" blocks to slow down the sequence. This gives you time to process if a piece worked or not. Remove these wait blocks once your code works.

Add Sound Checkpoints



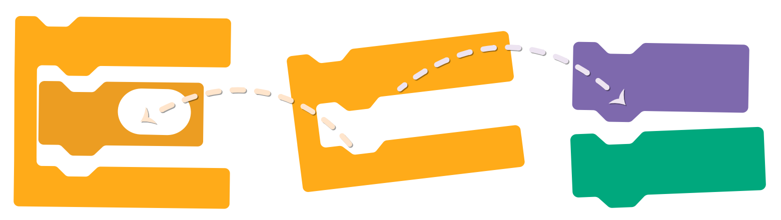
Similar to the Slow It Down strategy, you can add different sounds with the "play until done" block at key points to test your sequence. If a sound doesn't play, your bug may be before this block. If the sound plays, the bug is probably after this block. Remove the sounds once your code works.

Tinker with Block Order



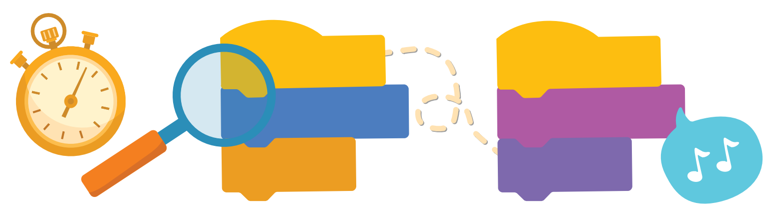
Try adjusting the order/sequence of the blocks. What needs to happen first? What happens second? Do values or sprites need to reset before the next piece of code runs? Try using blocks inside a loop or conditional statement, versus outside a loop or conditional statement.

To Loop or Not to Loop



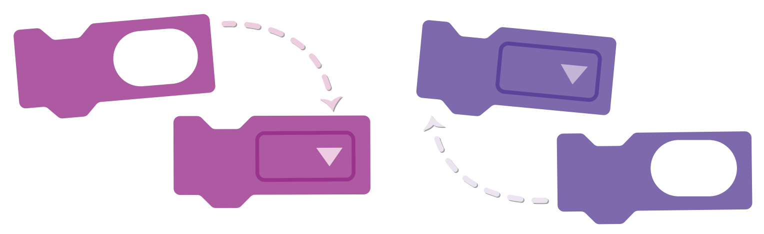
If using Control blocks like "forever" and "repeat", check that all blocks inside a loop should be there, or if a block (like "wait") is missing to reset the action or adjust the timing. Do you want your loop to run forever or for a certain number of times? Should something stop the looping? Perhaps you aren't using a loop when you should be? For instance, if you are using a conditional statement block like "if then," does the program only need to check if it is true or false once? Or does it need to check continuously, in which case, you would want to place your conditional statement inside a forever loop?

Think About Timing & Parallelism



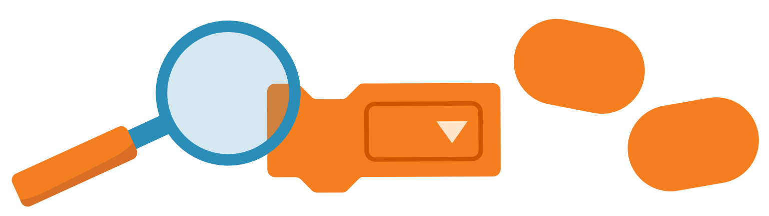
Do you have multiple events trying to run at the same time? If two sequences are programmed to start at the same time, you can get unpredictable behavior. Add small waits, broadcasts, or user interaction (like clicking or pressing a key) to see if this affects the result.

Think About Block Options



Is there a similar but different block you can use? Some blocks look similar but can behave differently, such as "set" vs "change" or "play until done" vs "start." Try using a similar block in place of what you have, and see the effect.

Check the Values



If you are using variables or reporter blocks, check the value at the moment the code sequence is run. Do/should all the sprites control a variable, or should only one sprite have control? Where is the value reset? Where is it changed?

Check Code Sequence

Check that your code sequence is attached to the correct sprite or the backdrop, if appropriate. If you need to move your code to another sprite, click and drag it until you are hovering over the correct sprite. Release it once the sprite wiggles. You can also use your Backpack (bottom of screen) to store and move your code or assets.

Comment Your Code

Adding comments to your code can help others looking at your code to understand it. It can also help you remember how your code works when you come back to it later. Right click on script area to "Add Comment." Use everyday language to explain what a block, or small sequence of blocks, does.

Take a Break, Step Away • • • Ask for Help

scratch.mit.edu

SCRATCH™ FOUNDATION