Lesson 1.6: In Search of Bugs - Part 1

Objectives

In this lesson, students will:

- ❖ Learn the concept of initialization code and debugging
- ❖ Investigate and learn strategies to debug and fix short programs
- ❖ Practice debugging and fixing code

Agenda

1. In Search of Bugs	5 mins
2. Learn through Debugging	15 mins
3. Student Activity: “Please help me debug my scripts!”	15 mins
4. Wrap Up and Reflections (Discuss Solutions)	15 mins

Preparation

- ❏ Create Scratch projects with the code for the debug 1 and debug 2 class exercises (optionally enter the code while instructing).
- ❏ Become familiar with the student activity debug exercises

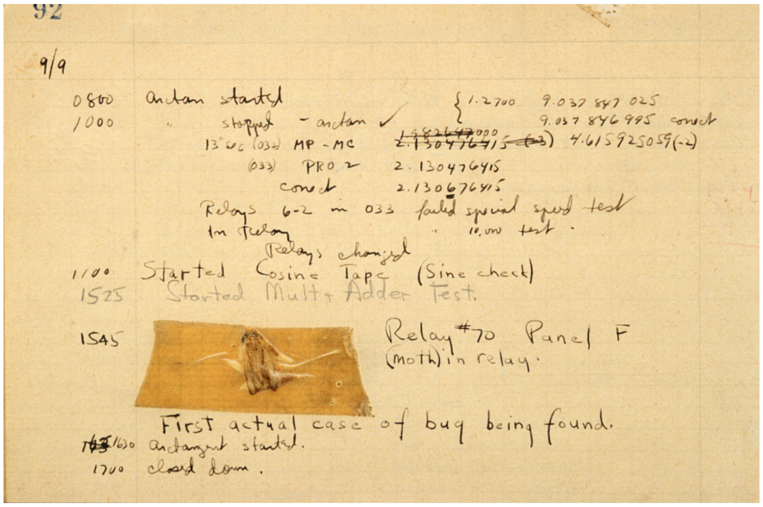
Resources & Links

- ❏ None

1. In Search of Bugs

It is very common for programs and code not to work the first time you try it. Programmers spend a great deal of time testing and debugging code. That is why it is important to develop good debugging strategies.

When there is a problem in the code, programmers refer to it as a bug. Program errors are called bugs because many years ago a very large computer called the Mark II malfunctioned because they found a moth stuck inside the computer. Grace Hopper, who was in charge of troubleshooting the computer, found the moth and taped it in the log book with the note "First actual case of bug being found"



92

9/9

0800 Arctan started

1000 " stopped - arctan ✓ {1.2700 9.037 847 025

9.037 846 995 correct

13" sec (032) MP - MC ~~1.982147000~~ ~~2.130476415~~ (-3) 4.615925059(-2)

(033) PRO 2 2.130476415

correct 2.130676415

Relays 6-2 in 033 failed special speed test

In relay " 10,000 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545 Relay #70 Panel F

(moth) in relay.

First actual case of bug being found.

~~143~~1630 Arctangent started.

1700 closed down.

Image from https://americanhistory.si.edu/collections/search/object/nmah_334663, *Log Book With Computer Bug*, National Museum of American History.

Since then programmers use the term debugging to remove bugs from programs.

2. Learn through Debugging

Activity Description:

Each of the following sections of code has a bug in it. Troubleshoot and fix each as an interactive class exercise. The fix is an opportunity to learn the programming concept of **initialization** code within a program and its importance. A program cannot make assumptions as to the initial values and locations of any sprites and should consequently explicitly set each initial or starting value at the beginning of the program. This section of code is called "**Initialization Code**".

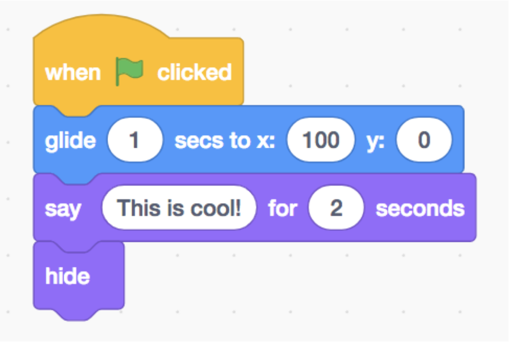
Directions:

For each debug exercise, project your screen for students to see.

- Start by stating the desired behavior of the sprite.
- Explain each line of code as you enter it in your project.
- Run the script pointing out what is supposed to happen. The expected behavior occurs the first time the script runs; however, when you run it a second time, it no longer runs as expected. In both debugging examples it is due to missing initialization code.
- Prompt students for the bug and the solution. Discuss the solution as a class exercise.

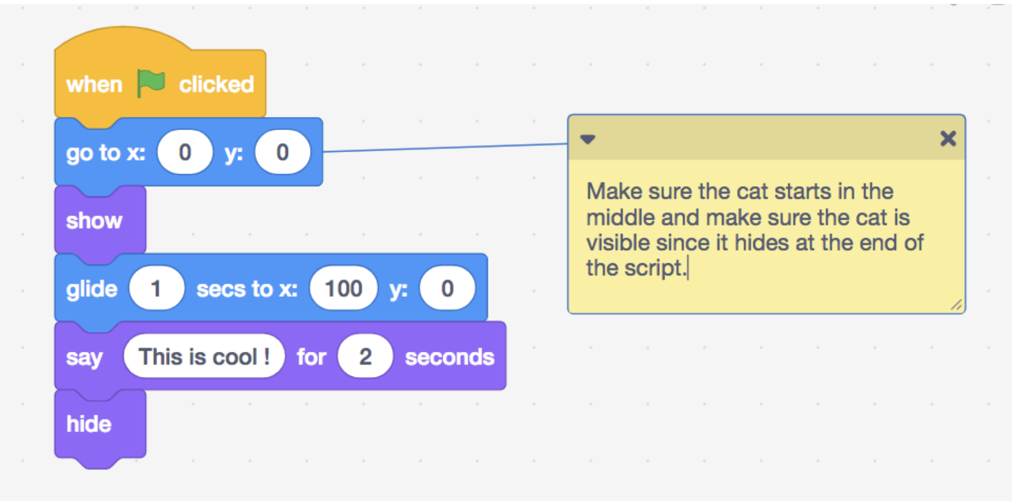
Debugging Demonstration 1:

In this section of code, the cat sprite should glide 100 steps to the right, say “This is cool!” and then hide. Run it a second time to see what happens.



```
when ⚑ clicked
glide (1) secs to x: (100) y: (0)
say (This is cool!) for (2) seconds
hide
```

Solution:



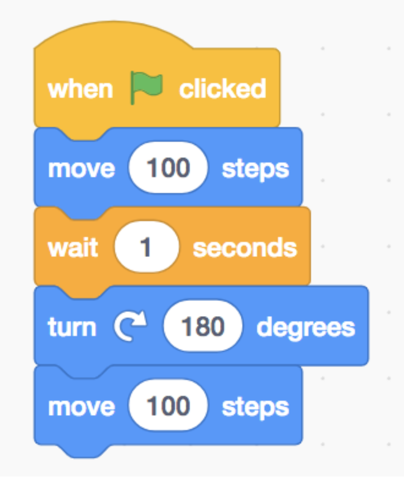
```
when ⚑ clicked
go to x: (0) y: (0)
show
glide (1) secs to x: (100) y: (0)
say (This is cool !) for (2) seconds
hide
```

Make sure the cat starts in the middle and make sure the cat is visible since it hides at the end of the script.

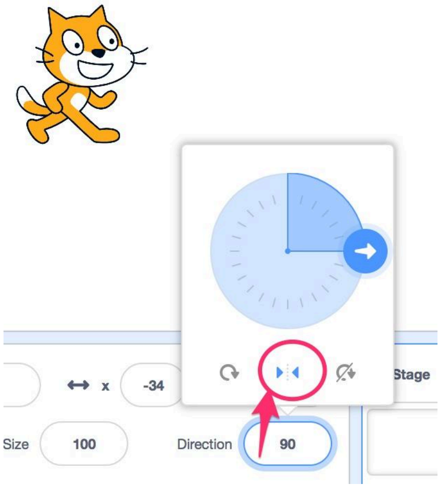
Debugging Demonstration 2:

In this section of code, the cat sprite should move 100 steps to the right, wait 1 second and then move 100 steps to the left by changing its direction.

So that the cat points in the correct position, click on **Direction** in the lower right area under the stage and then select >< for Left/Right rotation style as shown in the image below. Run the code a second time to see what happens. (the cat moves in the opposite direction the second time and then every other time)

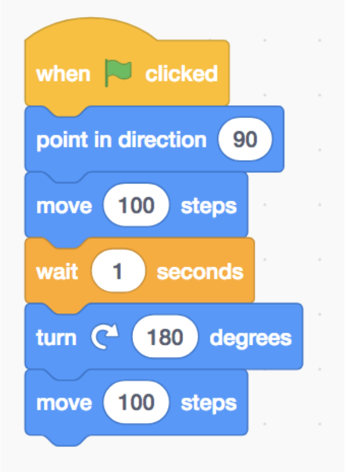


```
when ⚑ clicked
move (100) steps
wait (1) seconds
turn ↻ (180) degrees
move (100) steps
```



Solution:

Add the ‘**point in direction 90**’ instruction at the beginning of the program:



```
when ⚑ clicked
point in direction (90)
move (100) steps
wait (1) seconds
turn ↻ (180) degrees
move (100) steps
```

3. Student Activity: “Please help me debug my scripts!”

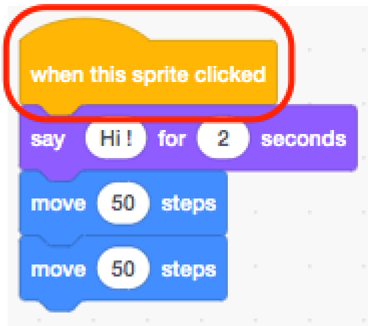
In this activity students will debug 2 scripts on paper. Explain the activity and distribute the activity worksheet.

4. Wrap Up and Reflections (Discuss Solutions)

Discuss the activity solution as a class exercise followed by reflections.

<ins>**Script 1:**</ins>
The **bug** is that the script starts with the “when green flag clicked”. The desired action is to click on the sprite.

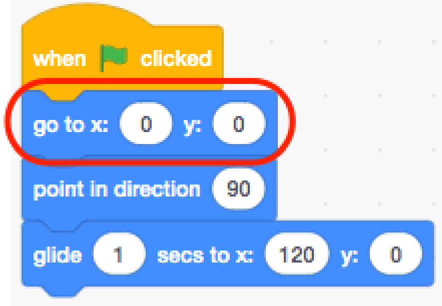
The **fix** is to replace the green flag block with the “when this sprite clicked” block.



```
when this sprite clicked
say Hi ! for 2 seconds
move 50 steps
move 50 steps
```

<ins>**Script 2:**</ins>
The **bug** is that the script is missing the initialization code to place the sprite in the middle of the stage.

The **fix** is to add the **go to x:0 y:0** block at the beginning.

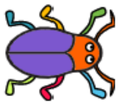


```
when 🏴 clicked
go to x: 0 y: 0
point in direction 90
glide 1 secs to x: 120 y: 0
```

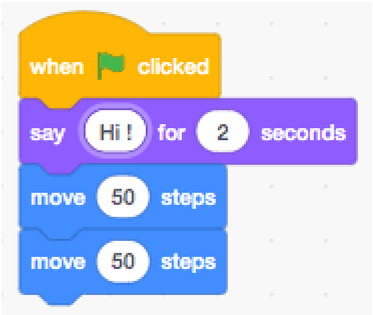
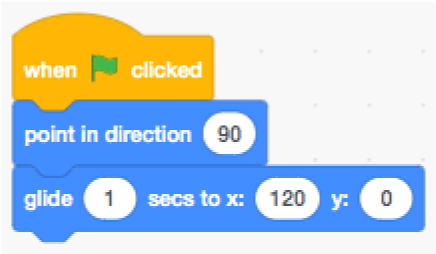
Reflection Points

Write down
- Date
- What is a bug?
- Why it’s important to initialize your sprite.
- What you liked about finding bugs.

Student Activity: “Please help me debug my scripts!”



This is Peter. I wrote some scripts for my beetle, but they are not working! Please help me debug them!

Script:	What is the Bug? What is the fix?
Script 1 I want the beetle sprite to say “Hi” and then move 100 steps when you click on it. But nothing happens !  when ⚑ clicked say Hi ! for 2 seconds move 50 steps move 50 steps	Bug: ______________________________ ______________________________ ______________________________ Fix: ______________________________ ______________________________
Script 2 I want the beetle sprite to start in the middle of the stage, and then glide to the right 120 steps. It works the first time, but not when I run it again.  when ⚑ clicked point in direction 90 glide 1 secs to x: 120 y: 0	Bug: ______________________________ ______________________________ ______________________________ Fix: ______________________________ ______________________________