



| MISSION 5: Fence Patrol<br>Lesson 2 (Objectives 3-5)  | Time Frame: 35-40 minutes   |                                      |   |  |   |                   |  |  |  |
|---|---|--------------------------------------|---|--|---|-------------------|--|--|--|
| <p><b>Project Goal:</b> Students will use functions to read data from CodeBot's line sensors and use the data to control the line sensor LEDs.</p> <p><b>Learning Targets</b></p> <ul style="list-style-type: none"> <li>I can assign a Boolean value to a variable using an if statement.</li> <li>I can use a Boolean variable to turn on/off an LED.</li> <li>I can define a function that reads a line sensor and uses the data to turn on/off an LED.</li> <li>I can define a function that uses a loop to call another function.</li> </ul>   | <p><b>Key Concepts</b></p> <ul style="list-style-type: none"> <li>Use threshold comparisons to make decisions with sensor data.</li> <li>A Boolean variable can get its value directly from an if statement.</li> <li>A function can use a parameter to indicate which line sensor to read and LED to turn on/off.</li> <li>A function can call another function as part of its code block.</li> <li>A while loop that repeats a specific number of times can be used to call a function multiple times.</li> </ul> |                                      |   |  |   |                   |  |  |  |
| <p><b>Assessment Opportunities</b></p> <ul style="list-style-type: none"> <li>Mission 5 Lesson 2 Log (digital)</li> <li>Quiz after Objective 5</li> <li>Submit completed program <b>LineSense</b></li> <li><a href="#">Mission 5 Obj. 3-5 Review Kahoot!</a></li> </ul>   | <p><b>Success Criteria</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Use an if statement to assign a Boolean value to a variable</li> <li><input type="checkbox"/> Use a Boolean variable to turn on/off an LED</li> <li><input type="checkbox"/> Define a function that reads a line sensor and then uses the data to turn on/off an LED</li> <li><input type="checkbox"/> Define a function with a while loop to call another function</li> </ul>                                       |                                      |   |  |   |                   |  |  |  |
| <p><b>Teacher Materials in Learning Portal</b></p> <ul style="list-style-type: none"> <li>Mission 5 Lesson 2 Slides</li> <li>Mission 5 Lesson 2 Log</li> <li>Mission 5 Lesson 2 Answer Key</li> </ul>   | <p><b>Additional Resources</b></p> <ul style="list-style-type: none"> <li><a href="#">Mission 5 Obj. 3-5 Review Kahoot!</a></li> <li>LineSense_obj5 sample code (learning portal)</li> <li>Testing Surfaces paper (learning portal)</li> </ul>  |                                      |   |  |   |                   |  |  |  |
| <p><b>Vocabulary</b></p> <ul style="list-style-type: none"> <li><b>DRY:</b> Don't Repeat Yourself – never write the same code twice.</li> <li><b>Function:</b> (Review) A named chunk of code you can run anytime just by calling its name; lets you reuse code without retyping it or copy/paste.</li> <li><b>Parameter:</b> (Review) A variable that gets its value when the function is called; part of the function definition.</li> <li><b>Argument:</b> (Review) A value that is passed to a function during a function call.</li> </ul>  |   |                                      |   |  |   |                   |  |  |  |
| <p><b>New Python Code</b></p> <table border="1"> <tbody> <tr> <td data-bbox="106 1425 726 1510">is_detected = ls.read(0) &gt; threshold</td><td data-bbox="726 1425 1537 1510">Assign a Boolean value to a variable using an if statement.</td></tr> <tr> <td data-bbox="106 1510 726 1636">while true:<br/>    is_detected=ls.read(0)&gt;threshold<br/>    leds.ls_num(n, is_detected)</td><td data-bbox="726 1510 1537 1636">Use a Boolean variable to turn on/off an LED.</td></tr> <tr> <td data-bbox="106 1636 726 1721">Surface detection</td><td data-bbox="726 1636 1537 1721">Dark line on light surface – use val &gt; threshold<br/>Light line on dark surface – use val &lt; threshold</td></tr> <tr> <td data-bbox="106 1721 726 1883">def detect_line(n):<br/>    is_detected=ls.read(n)&gt;threshold<br/>    leds.ls_num(n, is_detected)</td><td data-bbox="726 1721 1537 1883">Define a function with a parameter for detecting a line.</td></tr> </tbody> </table> |   | is_detected = ls.read(0) > threshold | Assign a Boolean value to a variable using an if statement. | while true:<br>is_detected=ls.read(0)>threshold<br>leds.ls_num(n, is_detected) | Use a Boolean variable to turn on/off an LED. | Surface detection | Dark line on light surface – use val > threshold<br>Light line on dark surface – use val < threshold | def detect_line(n):<br>is_detected=ls.read(n)>threshold<br>leds.ls_num(n, is_detected) | Define a function with a parameter for detecting a line. |
| is_detected = ls.read(0) > threshold  | Assign a Boolean value to a variable using an if statement.   |                                      |   |  |   |                   |  |  |  |
| while true:<br>is_detected=ls.read(0)>threshold<br>leds.ls_num(n, is_detected)  | Use a Boolean variable to turn on/off an LED.   |                                      |   |  |   |                   |  |  |  |
| Surface detection   | Dark line on light surface – use val > threshold<br>Light line on dark surface – use val < threshold  |                                      |   |  |   |                   |  |  |  |
| def detect_line(n):<br>is_detected=ls.read(n)>threshold<br>leds.ls_num(n, is_detected)  | Define a function with a parameter for detecting a line.  |                                      |   |  |   |                   |  |  |  |



|   |  |
|---|--|
| detect_line(0)  | Call a function that has a parameter for detecting a line.   |
| <pre>n = 0 while n &lt; 5:     detect_line(n)     n = n + 1</pre>                                   | While loop that repeats 5 times.<br>This loop uses n as the control variable, which is initialized outside the loop and incremented inside the loop. It is used to determine which line sensor to read and which LED to turn on/off. |
| <pre>def scan_lines():     n = 0     while n &lt; 5:         detect_line(n)         n = n + 1</pre> | A function that calls another function.  |

## Real World Applications

Functions are a form of abstraction, which is used all the time in real-world applications. Abstraction enables you to simplify systems to focus on essential features while hiding the details.

- You can be given a list of chores to complete without step-by-step directions for each chore.
- You don't need to know how an engine works to drive a car.
- You can follow simple directions to get to a destination without needing all the details.
- The chorus of a song is like a function that is called multiple times.

|  |   |
|--|---|
| <b>Teacher Notes:</b> <ul style="list-style-type: none"><li>• The slides should replace the instructions in CodeSpace. Code will be similar to CodeTrek, but a little different. All goals will be met.</li><li>• This lesson has students define and use functions. Do you need to review this process?</li><li>• You may want to review incrementing a variable and how a counter can be used to control a while loop for repeating a specific number of times.</li><li>• You may want to review using a variable for lighting LEDs.</li></ul> | <b>Extensions / Cross-Curricular:</b> <ul style="list-style-type: none"><li>• Change the while loop to start at the left LED and sweep right, instead of right to left. This means initializing the control variable to 5 and counting down.</li><li>• Play a tone when an LED turns on, or when an LED turns off.</li><li>• <b>MATH:</b> This program uses a function. Discuss what a function is in math. Compare and contrast.</li><li>• <b>LANGUAGE ARTS:</b> Have students write about an abstraction used in their daily lives.</li><li>• Supports <b>language arts</b> through reading instructions, guided notes, and reflection writing.</li></ul> |
|--|---|

## Preparing for the lesson:

- Look through the slides and workbook. Decide what materials you want to use for presenting the lesson. The slides can be converted to Google Slides. They can be projected on a large screen. The workbook (if used) can be printed or remain digital through your LMS and given to students.
- Be familiar with the mission log assignment and the questions they will answer. Prepare the assignment to give through your LMS.
- Have the Test Surfaces paper available for each student or programming pair.
- This lesson isn't really very long. You should have time to do some review for the concepts used in earlier missions.
- If you have a word wall, or another form of vocabulary presentation, prepare the new terms.



## Lesson Tips and Tricks:

### Teaching tip:

You can use a variety of discussion strategies to get the most engagement from your students. For example, you can have students write their answers before asking anyone for an answer. You can use one of many think-pair-share methods.

### Pre-Mission Warm-up: -- slide 2

Students can write in their log first and then share, or discuss first and then write in their log. The warm-up questions review concepts from earlier lessons. Students can share their answers, or compare with each other.

- Question: What code reads a line sensor?
- Question: What do you remember about functions?

### Mission 5 Lesson 2 Activities:

The Chrome browser works best, but other browsers also support CodeSpace. Each student will complete a Mission Log. Students could work in pairs through the lesson, or they can work individually.

#### Teaching tip: Mission Introduction -- slides 3-5

This mission is divided up into four lessons. The second lesson focuses only on the first and second goals.

#### Teaching tip: Objective #3 -- slides 6-9

This objective uses a Boolean variable. Do you need to review Boolean? It shows a new way to assign a Boolean value to a variable. You could show the two examples on the board and talk about how the new way works.

#### Teaching tip: Objective #3 Activity -- slide 10-11

The slides give the instructions and show the code for Objective 3. Students will need their Test Surfaces paper for this. You can also have a small black paper to use as the line to detect, as shown in the image on the slides.

#### Teaching tip: Objective #3 Activity -- slide 12

This is an optional extension that you can have students do if you have time. It reverses the if statement for white on black. If students do this extension, they should change their code back to the original before proceeding.

#### Teaching tip: Objective #4 -- slides 13-16

This objective discusses the use of functions to avoid code repetition. Review functions as necessary: how to define and call them. The concept was first introduced in Mission 4 Lesson 3.

#### Teaching tip: Objective #4 Activity -- slides 17-19

Students do not start a new program. They use the same LineSense program from Lesson 1.

When defining the function, students can cut their code from the while loop and paste it above the while loop in the function. If so, make sure they change the literal value 0 for the variable n.

The program should work exactly the same as Objective 3, just with a function. Test the program the same way.

#### Teaching tip: Objective #5 -- slides 20-21

This objective discusses using another function with a loop to call the first function. You might need to discuss the fact that functions can call each other, in case this is confusing to your students.

#### Teaching tip: Objective #5 Activity -- slides 22-25

Students define another function to their program. Functions usually are grouped together near the top of a program. It can go either above or below the current function. The slide shows it below. Then students modify their main program while loop.

The program can be tested the same way as Objective 4, but the black paper can go under each of the line sensors, and not just line sensor 0.



### **Teaching tip: Quiz -- slide 26**

Students take a  short quiz over objectives 1-5. The quiz questions are shown below.

You might want to review the data from Lesson 1, since there will be questions on the reflectivity values of different surfaces.

At the end of this objective, students should have a working program to turn in.

### **Teaching tip: Extension**

Two suggestions are given for an extension. These are completely optional, but if you have students who always work more quickly than other students, these are a good way to challenge them.

If any extension is completed, students should do a File–Save As and give the extension program a new name. The LineSense program will be continued (and completed) in the next lesson.

**Optional:**  Mission 5 Obj 3-5 Kahoot! Review.

A review Kahoot! is available for the three objectives.

### **Post-Mission Reflection:**

The post-mission reflection asks students to think about functions and their use in programming. It is an opinion question, so all answers should be valid. This is a good topic for class discussion.

You can use an extension or cross-curricular activity as post-mission activity.

End by collecting the Mission 5 Lesson 2 Log.

### **SUCCESS CRITERIA:**

- Use an if statement to assign a Boolean value to a variable
- Use a Boolean variable to turn on/off an LED
- Define a function that reads a line sensor and then uses the data to turn on/off an LED
- Define a function with a while loop to call another function

### **QUIZ Questions on next page**



## ?

 Checkpoint

Using **more reflective** objects, or moving them **nearer** to the sensor makes **+5 XP**  
the `ls.read(0)` values:

- Increase
- Stay the same
- Decrease

How could you make your program detect a **dark** line against a **light** **+5 XP**  
background?

- Set "threshold" to a higher value
- Use a different LED function.
- Use ">" instead of "<" in the comparison.

What does the *acronym* **DRY** stand for? **+5 XP**

- Design Reference Year
- Dont Repeat Yourself
- Defensive Rushing Yards