

**Mission 6 Introduction**

Read the introduction and project goals. During this assignment, you will complete the first two goals.

**Mission 6 Objectives 4-6**

Start Objective 4.  Discuss several ways that <code>ls.check()</code> is different from <code>ls.read()</code> .  What is a tuple?	<p style="color: red;"> <code>ls.check()</code> has a second parameter <code>is_reflective</code>  <code>ls.check()</code> returns a tuple  <code>ls.check()</code> is screaming fast          The ADC value is different.           A tuple is a read-only form of list.       </p>
--	--

Try Your Skills: Debug Console  
 Use the console panel to experiment with a new **thresh** value. Use the Test Surfaces color blocks.  
 Type in the console: `>>> from botcore import *`  
 Then type: `>>> ls.check(0)` for each color block and record the tuple.

Black: something similar to (3964, 3964, 3964, 3964, 3964)	Gray: something similar to (268, 244, 259, 248, 274)	White: something similar to (221, 206, 211, 205, 220)
--	--	---

Complete Objective 4. Use a <b>thresh</b> value from the data above.  Set the value of <b>is_reflective</b> to detect the line you are using. It will be either True or False	<p style="color: red;"> <i>Do a lot of testing to make sure your program works before continuing to the next Objective. Move the 'bot under a black line and make sure the correct LEDs are lighting.</i> </p>
---	--

Complete Objective 5. After reading the information in the Objective Panel, use the instructions below to modify the code.

<p>A. Define the <b>SPEED</b> constant and enable the motors near the top of the code with your other variables.</p> <p>B. Define a function to move CodeBot. In previous programs, you called the function <code>go_straight()</code> or <code>go_forward()</code> and both wheels used the same speed. This <b>drive()</b> function will give the 'bot the ability to turn as well as drive forward. Use parameters for two speeds (left and right).</p> <p>C. In the Main Program, add a <b>while True:</b> loop to wait for a button press.</p> <p>D. Similar to CodeTrek, in the second <b>while True:</b> loop, add the <b>if</b> statement below the two lines already there. Call the <b>drive()</b> function for each branch. The arguments will be the two speeds for left and right. The final else needs to call <code>motors.run()</code> 2 times.</p> <p>Try doing these steps on your own. Code is included at the end of the assignment, if needed.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> SPEED defined</li> <li><input type="checkbox"/> Motors enabled</li> <li><input type="checkbox"/> <code>drive()</code> function defined</li> <li><input type="checkbox"/> 'Bot waits for button press</li> <li><input type="checkbox"/> Branching if statement added for line control</li> </ul>
---	---

<p>Experiment with your code.</p> <p>Use a variety of short courses. Some examples of courses are given →</p> <p>Try the courses at different speeds and record the results.</p>	<ul style="list-style-type: none"> <li>● Smooth slow curve</li> <li>● Sharper curve (left or right)</li> <li>● Two curves, left then right</li> <li>● Sharp curve, like a right angle</li> <li>● U-turn curve that turns the bot in the opposite direction</li> </ul>
--	---

Add more rows as needed.

Speed of CodeBot	Turn direction (L / R)	Type of curve	Result
			Results will vary, but students
			should see that the 'bot doesn't
			work well on sharp turns.

<p>Complete Objective 6.</p> <p>What happens if the 'bot overshoots the line?</p> <p>What is the fix for this?</p> <p>Modify the <b>else</b> to an <b>elif</b>, as directed in CodeTrek. Also, change the 2 <code>motors.run()</code> calls to calling the function instead. (see below)</p>	<p>It seems like the 'bot is failing to detect the line, but that is not the case.</p> <p>When the 'bot departs the line, it goes back to "go straight" so it stays off the line. Even the slightest overshoot sends the CodeBot off on a tangent and away from the line.</p> <p>Keep turning when the line is lost. Only go straight if a line is detected by the middle sensors.</p>
--	--

Now experiment with your code again. Try the courses with the modified code. Add more rows as needed.

Speed of CodeBot	Turn direction (L / R)	Type of curve	Result
			Results will vary. The 'bot should
			work much better on the turns.

After Objective 6, turn in this assignment.

Objective 5 Sample Code:

Use YOUR values for thresh and is\_reflective

```
1  from botcore import *
2
3  thresh = 1000 # Use a thresh value that works for your 'bot and line
4  is_reflective = True # Depends on your line
5
6  SPEED = 30
7  motors.enable(True)
8
9  def drive(left, right):
10 |     motors.run(LEFT, left)
11 |     motors.run(RIGHT, right)
12
13 # -- Main Program --
14 while True:
15 |     if buttons.was_pressed():
16 |         break
17
18 while True:
19 |     vals = ls.check(thresh, is_reflective)
20 |     leds.ls(vals)
21
22 |     if vals[0]:
23 |         drive(0, SPEED)
24 |     elif vals[4]:
25 |         drive(SPEED, 0)
26 |     else:
27 |         # To meet the validator, call motors.run()
28 |         motors.run(LEFT, SPEED)
29 |         motors.run(RIGHT, SPEED)
--
```

Objective 6 Sample Code:

Modify the if statement:

```
if vals[0]:
|     drive(0, SPEED)
elif vals[4]:
|     drive(SPEED, 0)
elif vals[1] or vals[2] or vals[3]:
|     drive(SPEED, SPEED)
```