Part One: Modeling a Turn: Point

There are instance where turning in place, or a point turn is preferable. Can you list some of these situations?

_________________________________________________________________________________________________________
_________________________________________________________________________________________________________
_________________________________________________________________________________________________________

A point turn happens when both the left and right wheels move at the same power level, but in different directions. If the wheels below perform a point turn, what path will they take? Please draw in the path:

![Diagram of wheels performing a point turn]

Hopefully you saw that with a point turn, the robot turns in a circle with a diameter equal to the wheelbase of the robot (illustrated below).

What is the radius of this circle? (answer relative to the parts of the robot creating the circle)

What is the circumference of this circle?
Given the circumference found above, how many rotations and in what direction must the wheels travel in order to turn 90-degrees to the left?

Right_____
Left_____

What would happen if we increased the size of the wheels?

_____________________________________________________________________________________________________

_____________________________________________________________________________________________________

How would the addition of gears change your results?

_____________________________________________________________________________________________________

_____________________________________________________________________________________________________

Check your Understanding:
Which turning option is best for navigating a closed maze with a narrow wall width? Why?
Part Two: Displaying a Value

In part three we are going to write a program to test our calculations from part one. In order to display the number of rotations completed by each motor, it is important that the rotation sensor is zeroed before a reading is collected. Where in the program do you think the rotation sensor should be reset?

a. At the very beginning of the program
b. Just before the turn
c. Just after the turn
d. Just before displaying the value to the screen.

Answer: ________

If your answer is ‘b’ then you're right! The number of times the motor turns is constantly being tracked, so it is important that the rotation sensor is reset just before the turn if we are interested in knowing the number of rotations traveled during the turn.

After resetting the rotation sensor and completing the turn, we want to stop all motors and display the rotation count on the screen. The display icon can be tricky, and will only display values stored as text. Since the rotation sensor gives us numbers, NXT-G gives us a nifty way to convert numbers into text. The following code reads the value of motor a’s rotation sensor and prints it to the screen:

Check your Understanding:

Write a program that displays the name and age of each member of your group on a separate line on the LCD screen. The final product should look like something this (but obviously with your group’s information):

John is 15 years old
Jeff is 14 years old
Sara is 15 years old
Part Three: Testing your Turns

Now it’s time to combine the rotation display code from part two with the turns discussed in part one. A basic example of this is shown below (please note that this code does not perform all of the functions required for this step, but is meant to given a basic example of how to print rotation sensor values to the screen):

You should now program your robot to make a **90-degree pivot turn**. Your program should constantly print the number of rotations traveled by each motor on the LCD screen (this should update constantly while the program is running). Once you have completed the program, run it twice and record your results below in the columns designated ‘actual’. For comparison, fill in the predicted rotation value from part one in the table. Prior to writing this program in NXT-G, please explain in words (or pseudocode) how your program will work:

<table>
<thead>
<tr>
<th>Pivot Turn Table:</th>
<th>Predicted</th>
<th>Actual from Test 2</th>
<th>Actual from Test 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Pseudocode for Pivot Turn:**
Now, write a program to make your robot complete a 90-degree point turn. Again, the number of rotations traveled by the motors should be displayed to the LCD screen, and constantly updated as the program is running. Prior to writing your program, please outline your ideas in the pseudocode box below.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual from Test 2</th>
<th>Actual from Test 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pseudocode for Point Turn:

Are your predicted and actual values the same for both your point and pivot turns? Are you surprised by the outcome?