15-RobotMotionWithSensors

Read this entire document, with help from your instructor, before doing any of the assignment.

You are given the following:

- A `main` function that calls some testing functions.
- Some testing functions that begin empty. (You will write testing code in them.)
- A **`Motor`** class, as described below.
- A **`ColorSensor`** class, as described below.

Your task is: Implement a *`SimpleRoseBot`* class as described on the next page, along with code for testing its methods.

Here are the **`Motor`** and **`ColorSensor`** classes that you are *given* (that is, they are already implemented for you). They have methods as indicated below:

Motor

Methods:

`__init__`(*port*)

`turn_on`(*speed*)

`turn_off`()

`get_position`()

The *port* must be either **`"B"`** or **`"C"`**, for the left/right wheels' motors, respectively.

The *speed* must be an integer between -100 (full speed backward) and 100 (full speed forward).

The units for the wheel's position are "tachometer counts" that are internal to the wheel. You are given constants that help you convert to inches:

`Motor.TachoCountsPerRotation` (happens to be XXX)

`Motor.WheelDiameter` (happens to be XXX)

ColorSensor

Methods:

`__init__`(*port*)

`get_reflected_light_intensity()`

The *port* must be 3 (since the color sensor should be plugged into port 3 on the brick).

The physical color sensor is on the bottom of the robot. It shines light down (in this case, red light).

The color sensor measures the intensity of the reflected light. In our classroom, you will typically see numbers that range from roughly XXX (not much reflected) to roughly YYY (lots reflected).

SimpleRoseBot

Methods:

```
__init__()

go(left wheel’s speed, right wheel’s speed)

stop()

go_straight_for_seconds(seconds, speed)

go_straight_for_inches(inches, speed)

go_straight_until_black(speed)
```

where `__init__` must construct and store:

- ○ A `Motor` for the left wheel (in port “B”).
- ○ A `Motor` for the right wheel (in port “C”).
- ○ A `ColorSensor.`

Each speed is between -100 and 100. The methods do what their names suggest. For pedagogical reasons, **you may NOT use** *`time.sleep`* anywhere in this session’s exercises.

Use the following ***iterative enhancement plan***.

1. With your instructor, implement and test the ***measuring_time*** function, so that you understand how to measure elapsed time (which you will need in some of the following steps).
2. Implement and test `__init__`. Put the testing code in *`run_test_init`* (and similarly for the other methods as you implement them). For the test, just construct a `SimpleRoseBot` and make sure that the program does not crash (i.e., throw an Exception) when it runs.
3. Implement and test **go** and `stop` (together). For the test, make the robot go, then have the program wait for (say) two seconds, and then make the robot stop. Test with various speeds for the wheels.
4. Implement and test `go_straight_for_seconds`. It will be remarkably similar to the testing code that you wrote in the previous step of this plan.
5. Implement and test `go_straight_for_inches`. Use a Motor’s `get_position` method (left motor or right motor, your choice). Convert from the motor’s internal tachometer units to inches by using appropriate arithmetic applied to the `Motor.TachoCountsPerRotation` and `Motor.WheelDiameter` constants.
6. Implement and test `go_straight_until black`. Use the `ColorSensor` to judge when the robot is on top of a “black” surface; experiment to find a reasonable number to use as the threshold.