ASSIGNMENT 5

GENE-121-2017-3F-ASSN-05-Description

Fall 2017

CONTENTS

INTRODUCTION

This assignment is to be completed in pairs.

This assignment is due on Monday Oct 16 at 8:20 am in the physical dropbox. **However, you are strongly encouraged to complete the assignment as soon as possible so that you can concentrate on midterms. Assignments submitted before Friday Oct 13 at 8:00pm will receive a 10 mark bonus on the assignment.**

In this assignment, you will be implementing a small RobotC program on Lego EV3. The first problem is expected to take each group approximately 30 minutes to complete. The second problem is a C++ exercise. Both of these problems are exercises in implementing and calling functions.

At the start of the lab session please sit at the robot kit with the number you have been assigned from the partner list. Please do not alter the physical construction of the robot. When you are finished, do not dismantle the robot. Simply return the robot in its original configuration to a TA.

You can re-use functions from your Tron Days code so long as you acknowledge the authors.

QUESTION 1 – TILES OF HYRULE

Link has installed a new tile floor at his home and needs your assistance checking the tile dimensions. The tiles are all blue or red. Link is only interested in measuring the red tiles.

Implement a RobotC program on EV3 to drive the robot over the tiled floor and measure the red tiles. The robot will drive in a straight line over alternating blue and red tiles, starting on a blue tile.

RED TILE FUNCTION

Implement and use a well-named function that drives the robot over a red tile, stops when it reaches a blue tile, and returns the width of the red tile in millimeters (integer value). Ignore any readings that are not red or blue. You can assume that the color sensor is on the leading edge of the red tile when the function is called.

BLUE TILE FUNCTION

Implement and use a well-named function that drives the robot over a blue tile and stops when it reaches a red tile. Ignore any readings that are not red or blue. You can assume that the color sensor is on the blue tile when the function is called.

MAXIMUM FUNCTION

Implement and use a well-named function that returns the maximum value of three integers that are passed to it.

MAIN FUNCTION

Implement the main function such that the robot measures three red tiles and stops. After stopping, the robot should display the maximum tile width for the three red tiles.

DEMO

Demonstrate the operation of your robot to a Teaching Assistant, and submit your code.

QUESTION 2 – RABBIT SEASON (C++)

Prof. Hulls is tired of rabbits eating her lettuce plants in her backyard. She has deployed a 360-degree ballistic Nerf dart rabbit deterrent system, which was placed in the center of her back lawn. Over a 24-hour period, the system has recorded the heading and distance to target for each launch. The recordings have been saved to a file called wascally_wabbits.txt.

Each line in the file contains the following information, in the given order:

- rabbit number
- time of launch, recorded as a six-digit integer of the form hhmmss
- heading to rabbit (in degrees, between 0 and 360)
- distance to rabbit (in meters)

The system tracks and launches Nerf darts at a single rabbit until that rabbit leaves the yard (i.e. no more recordings for that rabbit), which now happens in less than a few hours. The darts are so effective that rabbits that leave never return. This means that all recordings for a single rabbit will appear in a contiguous block in the file. For example, all the readings for rabbit 8 appear in consecutive lines, then all the readings for rabbit 12 appear on the next block of lines, etc.

A heading of 0-degrees corresponds to the direction of the unit vector (1, 0).

For each line in the input file, output the following to a file:

- rabbit number
- time in hh:mm:ss format (Hint: Find out how to use `setfill('0') << setw(2)`)
- (x,y) coordinates of rabbit position
- duration since last launch for a given rabbit, in seconds – if it is the first launch for a rabbit, then the duration is 0.0

You must implement and use the following functions in your program.

```
int hours(int time_hhmmss);
```

Returns the hours portion of the input time. The input time is in hhmmss.

```
int minutes(int time_hhmmss);
```

Returns the minutes portion of the input time. The input time is in hhmmss.

```
int seconds(int time_hhmmss);
```

Returns the seconds portion of the input time. The input time is in hhmmss.

```
double x_coordinate(double heading, double distance);
```

Returns the Cartesian x-coordinate of the position when converted from radial coordinates (angle and radius). You may re-use (with acknowledgement) your code from Tron Days.

```
double y_coordinate(double heading, double distance);
```

Returns the Cartesian y-coordinate of the position when converted from radial coordinates (angle and radius). You may re-use (with acknowledgement) your code from Tron Days.

```
int get_duration(int start_hours, int start_minutes, int start_seconds,
        int end_hours, int end_minutes, int end_seconds);
```

Returns the number of seconds that have passed between the two times. Note that this function needs to correctly account for crossing over midnight.

SAMPLE OUTPUT

```
 8    0:12:18    (  7.53,  1.01)       0
 8    0:12:33    (  9.61,  2.36)      15
 8    0:12:59    ( 10.44, -6.12)      26
 8    0:13:05    ( 10.87, -6.38)       6
12    1:36:18    ( -0.72, -0.96)       0
12    1:37:14    (  0.99,  0.48)      56
...
...
Make sure file goes over midnight
```

SUBMIT

Submit code with output.

EXTRA PRACTICE FOR MIDTERM

As a good practice exercise:

1. Write a function

```
double get_distance_travelled(double last_x_position,
        double last_y_position,
        double current_x_position,
        double current_y_position);
```

Returns the distance travelled between the two points (last_x_position, last_y_position) and (current_x_position, current_y_position).

2. Modify your program to output the distance that a rabbit travels between darts.

3. Modify your main program to find the rabbit that travelled the furthest and also the rabbit that travelled the fastest.

This extra practice does not need to be submitted, will not be marked, and is not worth any grades.