COMP101 Lab 4: Second Assessed Coursework
Robot Movement

worth 12% of the final mark, assignment two of seven

Failure to submit this assignment or submission of work that is deemed not to be a reasonable attempt will result in the failure of the module. The completion of the implementation and report as described below will obtain 90% of the marks. To obtain the final 10% students should also complete the Extended Requirements. This may involve doing some additional reading beyond what has been presented in lectures or more complex programming. Only attempt the extended requirements if you are confident with programming.

Learning Outcomes. This assignment addresses the following learning outcomes:

- to be able to implement, compile, test and run Java programmes, comprising more than one class, to address a particular software problem;
- to understand how to include arithmetic operators and constants in a Java program;
- to be able to make use of members of classes found in the Java API (such as the Math class).

Requirements. A robot moves at an *angle* between 0-90° from the vertical in a clockwise direction. (See Fig. 1.) The robot travels at a fixed *speed* between 1.0-5.0m/sec for some *time*. Assume that the *angle*, *speed*, and *time* are input by a user, and the angle has values between 0-90 (degrees), the speed is a real number between 1.0 and 5.0 (m/sec), inclusive, and the time travelled is greater than or equal to zero (seconds). To move, the robot uses more power than when it is idle. An estimate of the battery used, in terms of battery time required for an idle robot, can be calculated by the time spent moving multiplied the square of the speed, multiplied by 3.7.

Design, implement, and test a Java program to calculate and print out

- the distance traveled by the robot;
- the horizontal distance from its starting point;
- the vertical distance from its starting point;
- the estimated battery usage (in terms of idle time).

The following equations can be used to calculate each of the above

$$\begin{aligned} distance &= speed \times time \\ horizontal &= distance \times (sin\ angle) \\ vertical &= distance \times (cos\ angle) \\ batteryEstimate &= time \times (speed)^2 \times 3.7 \end{aligned}$$

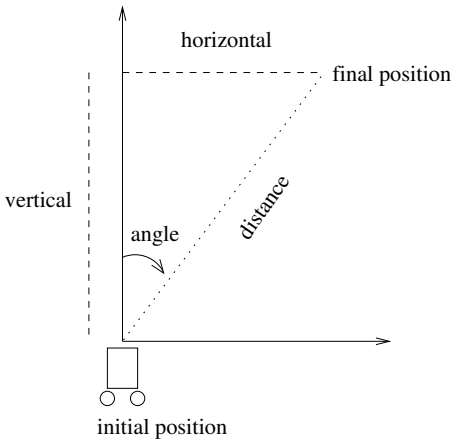

Figure 1: Robot movement diagram

where the second and third equations use the trigonometric functions sine and cosine. Assume that a user will not input impossible values, i.e.

- the input angle will be greater than or equal to 0.0 degrees and less than or equal to 90.0 degrees;
- the speed will be between 1.0 and 5.0 (inclusive); and
- the time will be a positive or zero (real) number.

You don't have to test for values outside these ranges. The angles should be input in degrees (not radians).

Hints. The calculations of sine and cosine are from the Java Math class. Note that sine and cosine require their input in radians (another way of measuring angles) rather than degrees, so you will have to convert the degrees to radians using methods from the Math class.

Example. angle = 30 degrees, speed = 1.5 m/sec, time = 10 seconds
Calculate distance:

$$\begin{aligned} distance &= speed \times time \\ &= 1.5 \times 10 = 15 \end{aligned}$$

Calculate horizontal distance:

$$\begin{aligned} horizontal &= distance \times sin\ angle \\ &= 15 \times sin\ 30^\circ \\ &= 15 \times 0.5 \\ &= 7.5 \end{aligned}$$

Calculate the vertical distance:

$$\begin{aligned} vertical &= distance \times cos\ angle \\ &= 15 \times cos\ 30^\circ \\ &= 15 \times 0.8660254 \\ &= 12.990 \end{aligned}$$

Calculate estimated battery usage (in seconds of idle-time battery usage):

$$\begin{aligned} estimatedBatteryUsed &= time \times (speed)^2 \times 3.7 \\ &= 10 \times (1.5)^2 \times 3.7 \\ &= 83.25 \end{aligned}$$

The output from this example is given below. Your output should be similar in nature (obviously the numbers will be different with different input.)

```
$ java RobotMovementUser
Input the angle, measured clockwise from the vertical a real number between 0-90: 30
Input the speed, as a real number between 1.0 - 5.0: 1.5
Input the time travelled a positive or zero real number: 10

A robot moving with angle 30.0, at a speed of 1.5, for 10.0 seconds has:
 the distance travelled: 15.00
 horizontal movement: 7.50
 vertical movement: 12.99
 the estimated battery usage (in idle-battery seconds): 83.25
```

Extended Requirements. Develop an application that uses the target class and allows the robot to move three times with the second and third movements being from the position the robot reached previously. You will need to allow the input of three *angles*, three *speeds*, and three *times*. The program should calculate and display the total distance traveled, the total time traveled, the horizontal and vertical distances of the final position from the initial position (see Figure 2), and the total estimated battery usage. If you attempt the extended requirements, make sure that you also submit a program that satisfies the standard requirements.

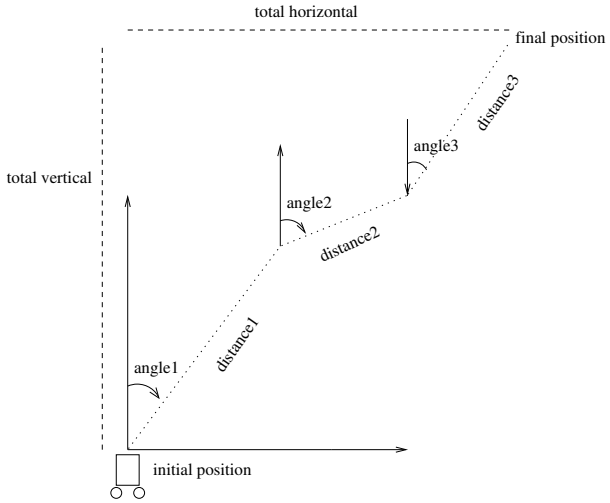

Figure 2: Robot Movement Extended Requirements

Submission Instructions. Your submission should consist of a report (as a PDF file) and implementation files.

- The report should consist of

  Requirements: Summary of the above requirements statement.

  Analysis and Design: A short (one paragraph) description of your analysis of the problem including a Class Diagram outlining the class structure for your proposed solution, and pseudocode for methods. Note that your solution should comprise (at least) two classes: an "application class" and a "target class".

  Important: Make certain that your application class has the name "RobotMovementUser". If you do the extended requirements, make certain your application class has the name "RobotMovementUserExt".

  Testing: A set of proposed test cases presented in tabular format, including the expected output, and evidence of the results of testing (the simplest way of doing this is to cut and paste the result of running your test cases into your report).

- The implementation should consist of

  Your Java source files, i.e. the relevant .java files, not the class (.class) files.

Upload your files to

https://sam.csc.liv.ac.uk/COMP/Submissions.pl

(you will need to log in using your Computer Science username and password).

Submission Deadline. Monday 31st October, 5pm.

Mark Scheme. Marks will be awarded for

- Analysis and Design 15%
- Implementation 60%
- Testing 15%
- Extended requirements 10%

Please see the module web page for the feedback form.

Note.

- Because submission is handled electronically, ANY FILE submitted past the deadline will be considered (at least) ONE DAY late. Late submissions are subject to the University Policy (see www.csc.liv.ac.uk/department/regulations/practical.html).
- Please make sure your Java classes successfully compile and run on ANY departmental computer system, both Windows and Unix. For this reason the use of powerful IDEs like NetBeans is discouraged.
- Note this is an individual piece of work. Please note the University Guidelines on Academic Integrity (see https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix_L_cop_assess.pdf).