COM160 Spring Semester and COM162 2015-6

Assignment

Dr Siobhán North

Department of Computer Science, University of Sheffield

Introduction

The purpose of this assignment it to give you an opportunity to demonstrate that you can write well structured, reusable java code. It will also give you a chance to experience the difficulties of working on the interaction between hardware and software. Most of the marks will be for the quality of your code rather than a working program or the use of the hardware, Lego® robots.

The Programming Task

This task is based on my (limited) understanding of the game of Rugby. As I understand it, it is necessary to pick up an elliptical ball, run with it and put it down the other side of a line. You are to program your robot to go to a round ball, pick it up or otherwise capture it, transport it slowly and carefully to a line and put it down on the other side.

The Lego EVA robots consists of a small microcomputer which can be slotted on top of a chassis. The chassis has two motors connected to its two wheels. These motors must be connected to ports on the microcomputer and the microcomputer can tell them whether to go forward or backwards, change speed or stop. You tell the microcomputer what commands to issue through a Bluetooth connection from a program running on a desktop computer.

The robot comes with a box of components, one of which is a third motor and this can be used to move whatever you create to capture the ball. The manual that comes with the robots has a design for one sort of device but you can use anything your like. You can only components in your box, not anyone else's, so you can only use one extra motor.

You should instruct you robot what to do using a Graphical User Interface on the desktop computer. Your GUI will probably need buttons for "Pick up the ball" and "Put down the ball". The robots are not mechanically perfect so something on your user interface that allows you to steer the robot may be useful even if it is just "Left a bit", "Right a bit" buttons.

Robot/Program Communication

On the course web pages there is information about fitting your robot onto its chassis, connecting your computer and your robot, the various libraries you will need to download to make this communication work, documentation for the robots and a sample program to practice with. This is at

https://staffwww.dcs.shef.ac.uk/people/S.North/campus_only/com162/robots/index.html

It also tells you how to compile robot programs without a robot so you can write your GUI at home before the practical if you want to.

The program will have to be submitted so ensure your program is in a new Netbeans project all by itself before you start programming because that will make the submission a lot easier. Your program will need to be readable when printed out so make sure the lines are not too long. You will get marks for style as well as a working program (see below for full details) so remember to use meaningful variable names which follow the usual Java naming conventions and split your work into sensible classes.

Using the Robots

There are almost enough robots to ensure you each have your own robot to work on. It won't be exclusively yours because other groups of students may use them occasionally but mostly it will be yours. I said there are almost enough robots to go around but in fact there are 4 robots too few. I don't want to allocate a robot to a student who won't bother to use it so I am going allocate robots to students only at the Friday practical sessions; this ensures the most conscientious students get first go and students who never attend anything run the risk of being one of the last few students who have to share a robot.

You will be not be able to use the robots until after the Easter vacation but then you can use them either during the Friday practical sessions or on Wednesday afternoons. On Wednesdays, if your robot cabinet has not been unlocked and none of the purple shirted Diamond technicians are around you will need to go to Engineering reception on the 3rd floor of the Diamond and ask someone to unlock your robot cabinet for you. It may be possible to arrange extra sessions near the deadline too.

Assessment & Submission

The deadline for submission is **3pm on 10th May**. You will not only have to submit your work on paper and electronically (see below) you will also have to demonstrate your program working on either the 10th or the 11th. The timetable for individual students to demonstrate their work will be released later and is open to a certain amount of negotiation. If you have good reason, so is the date, **but attendance at a demonstration is compulsory** and students who fail demonstrate their robots doing something (simply moving a bit and then falling over is sufficient), will not get their work marked.

You must submit your work by 3pm on Tuesday 10th May in two forms - both on paper and electronically. Only work submitted in **both** forms will be marked.

The paper submission should be put in the drop box in Computer Science reception (1st floor, Regent Court) with the usual Faculty of Engineering cover sheet (from https://foe-coversheet.group.shef.ac.uk/) stapled on top. It must consist of readable printouts of all your **`.java`** files and a screen dump of your user interface. The electronic submission should just be all the files in your Netbeans **`src`** directory for the program as a zipped folder.

The usual University rules about late submission apply and so do the usual rules about the use of unfair means. This is an individual assignment and your work will be put through a plagiarism checker.

The Marking Scheme

A working program.

A program which does everything required and has a practical, usable user interface will get full marks but marks will be awarded for whatever the robot can do up to a maximum under each heading as follows:

	Upto
Getting from the start to the ball	9
Making an effort to capture the ball	9
Making an effort to convey the ball across the arena to the line	9
Attempting to put down the ball once the line has been crossed	9
Succeeding in capturing the ball at the start and putting it down at its destination without losing it on the way	4
The functions of the GUI	5

Program readability and Style.

Your program should be readable both on screen and when printed out. Avoid very long lines and print your work in landscape format if the lines still wrap around when printed. Your program should be well commented to describe what is going on. All variables and methods should have meaningful names. The indention should make it clear which brackets match. The layout should enhance readability with blank lines to separate blocks of code. A program which is easily understandable both on the screen and on paper will get full marks.

You should adhere to all the java naming conventions (see page 10 of the slides from lecture 3 for a summary of these). You should use programming structures such as loops and conditional statements appropriately. You should use the correct types for all variables and use constants where appropriate.

	Upto
Readability	5
Correct identifiers	5
Correct use of types and constants	10

Use of Classes and Methods.

The program should be structured in a sensible way where different sorts of code is grouped into different classes. You should make extensive use of methods. All the methods should be short with meaningful names and sensible parameters. Long methods or, worse still, repeated code will lose you marks.

	Upto
Use of classes	15
Use of methods	20