COM1003 Java Programming - Autumn Semester 2017-8

Assessed Assignment 3

Dr Siobhán North

Department of Computer Science, University of Sheffield

Learning outcomes

This assignment will assess your ability to:

- Understand how software objects can be used to represent real-world objects;
- Have gained experience of problem solving and algorithm design;
- Understand how use the API documentation to explore what methods are available;
- Have gained experience of working in groups to solve a problem;
- Appreciate the difficulties involved in developing robust systems that involve the interaction of hardware and software, and operate in a real-world environment.
- Practiced adapting a class for reuse.

This assignment is worth 20% of your mark for the first semester of the module so 10% of your overall mark. It must be submitted by 19 January 2018. You will find information about the exact deadline, the marking scheme and how you must submit your work at the end of this document.

The Task

Your task is to make your robot pretend to be a bee and collect nectar. The nectar will take the form of a ping pong ball.

The robot will start in one of the corners of the arena and should set out from there following a black line. It should do this autonomously, without help from you. When it gets to the centre there will be a coloured area with a lot of orange ping pong balls. Your robot should pick up one of these and get it back to your original corner (not necessarily by following the line) and put it down there. Once it has put down the ball it should perform some sort of dance to indicate success but it is not necessary to try to duplicate a real Waggle dance.

There are several complications to this task. Whilst the robot should get to the centre by itself it will not be able to do the rest of the task without direction. Your robot will need some sort of ping pong ball catcher, and it will need to know when to deploy it. The robot has sensors but they will not work on ping pong balls (although they will be fine for following the line) so you will need to give the robot a command when you want it to reach for the ball or put it down. In order to do this you should adapt the program you used to play robot football in the first week of term by modifying the Kick option to tell it to pick up and put down a ball.

There is no time limit on this task and if it drops the ball at the last minute it can go back to the centre and try again but picking up ping pong balls from outside the coloured area is forbidden.

Working in Teams

You must work in teams of four for this assignment. Faculty rules are that, for groups of more than two I have to dictate who you work with. The team list is on the course web page at https://staffwww.dcs.shef.ac.uk/people/S.North/campus_only/com1003/assignments/teams.html which is linked from the usual assignment web page.

Teams can decide how to divide responsibilities between themselves. However, all group members must contribute to the development of the software and the contribution of each member must be accounted for (see below). If there is evidence that a member of a group has not contributed equally to the project, that student's mark will be adjusted accordingly.

If you have difficulty contacting one of your fellow team members let me know.

Using the Robots

Groups of students will be allocated a specific robot. The name of the robot will appear on the web page with the list of team members.

You may access your robot during the two practical sessions and on Wednesday afternoons and whenever there is a Diamond technician in Computer room 3 to unlock the cabinets. During the vacation you will not be allowed to use the robots so there is no advantage to students staying in Sheffield. I hope, but cannot yet promise, that they will be accessible for much of the week starting 16 January.

Please do not take the robot out of the Diamond computer room 3 under any circumstances.

The Program Code

The starting point of your program should be the **`main`** method in a class called **`Assignment3`**. It should not be part of a package.

Your program should be well written and commented to show how you designed your algorithms. Put the comments that indicate the algorithm in lines by themselves not at the end of a line so that a printout of the program is also readable. All public methods should also be commented using the **`javadoc`** facility (which will not be described until next week) and the class(es) should include a properly used **`@author`** tag.

Read your feedback from Assignment 2 as soon as you get it (I will try to mark this assignment before the Christmas vacation) to make sure you learn from it and don't make the same mistakes again. You could also reread the feedback from Assignment 1.

Demonstrating your Code

As well as handing in your code you will need to demonstrate it working at some time on the 18th or 19th of January. I cannot be precise as to when yet (because I have to negotiate with Exams Office for the use of the room and they may not know when they need the room until the final exam timetable is settled) but I will confirm as soon as possible. The demonstrations will not be allowed to run into the following week because that will clash with the Global Engineering Challenge but if you want to demonstrate your working program early it might be possible. Let me know if you do.

This is an examination and the attendance of **all four** students is mandatory. All students may be asked questions about their software and how it works and missing students will lose marks.

If you are unable to attend for exceptional reasons you should fill in an Extenuating Circumstance Form (see the Undergraduate Handbook).

Submission and deadline

You should hand in your solution using the standard departmental hand in procedures (see the Undergraduate Handbook) so it will be both on paper and electronic. Only one copy of the paper and electronic submission is required per team.

The electronic hand in should be uploaded to MOLE (Click on the Assignment 3 icon where you downloaded this piece of paper from and attach your file) and consist of the file **`Assignment3.java`** with a main method to provide the starting point for the program. If you use any other classes you should upload their **`.java`** files as well. The upload must consist of separate **`.java`** files, do not upload a directory in any form. The deadline for the electronic hand in is 3pm on 19 January 2017. You do not need to upload anything except the program code. It does not matter which member of the team uploads the code as long as one of you does it.

The paper hand in should be stapled together and consist of

- The usual bar coded cover sheet **on top** of everything else;
- A title page with the names of your team members on it;
- Printouts of any code you upload in a readable form (so if the lines are too long to printout legibly then print it in landscape format);
- A statement containing a paragraph from each team member, explaining his/her contribution to the project effort. This statement should be **signed by all team members** to indicate that it is a fair and accurate record.
- The deadline for the paper hand in is also 3pm on 19 January 2017.

You have to submit your work in both forms to get a mark and the paper and electronic versions of the .java files must be identical.

Late work will be penalised using the standard University scale (see Undergraduate Handbook). Your work will be considered to have been handed in on the day the second of the two forms (paper or electronic) is handed in.

Assessed Assignment 3

Marking Scheme

The mark for this assignment is worth 20% of your mark for the first semester of the module so 10% of your overall mark.

The marks will be allocated as follows

Programming Style

Program structure (good use of classes and methods)	15 marks
Correct use of variables, types and constants	15 marks
Use of appropriate selection and repetition statements	5 marks
Readability (so it must be easy to understand)	10 marks

Matches Specification	5 marks

Working Demonstration

Robot can follow a black line	7 marks
Robot can respond to a ping pong ball by trying to catch it	7 marks
Robot can successfully pick up or control in some way a ping pong ball	8 marks
Robot can transport the ball	8 marks
To the correct destination	7 marks
Put it down	8 marks
And dance	5 marks