COSC343 Assignment 1

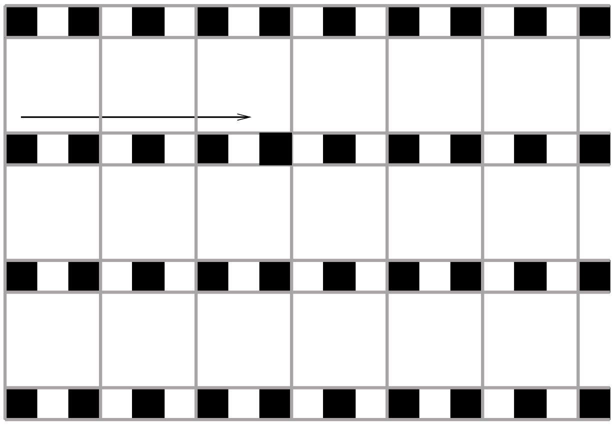
Robots on a chessboard

Weight:20% Lecturer: Lech Szymanski

For this assignment, you will be working in the group that you were assigned to during Tutorial 1, when you first used the LEGO robots. Any students who weren't at that tutorial should email Craig (`atkcr398@student.otago.ac.nz`), and they'll be assigned to an existing group. If you have any other questions about group allocation, you should also email Craig. If you have problems with your group, you should again contact Craig as early as you can.

Task 1 (10 marks)

Your group's first task is to write a Python program which will allow a LEGO robot to travel between two points in the lobby of the Owheo building. The lobby is laid out with black and white tiles, in a pattern that looks like this:



Your robot will start on a tile marked 'S' (for 'start'), in the direction of the arrow; it has to reach a plastic tower placed on a tile marked 'F' (for 'finish'), which is positioned on a diagonal from the start tile. The robot must travel to the tower in two stages. In the first stage, it must travel in the direction shown by the grey arrow in the figure, a distance of 15 black tiles. In the second stage, it must turn 90 degrees to the right, and travel to the plastic tower placed at the finish square, a distance of seven large tiles. It must make contact with the tower, and push it off the square it's sitting on, then make another sound to indicate it has finished.

Here are some additional specifications for the task.

- During the first stage, your robot must explicitly count the black tiles it passes, making a distinctive sound when it encounters each one.

- During the second stage, there is no requirement to count tiles: the robot can move by ‘dead reckoning’, turning a prespecified angle and driving forward a prespecified distance. But counting tiles may allow more accuracy.
- You can use the sonar sensor to locate the tower, and the bump sensors to detect contact with the tower.

One important feature of the lobby environment is that the light is very variable: it changes throughout the day, and also from point to point in the corridor. To sense different surfaces on the floor, your robot may need to adjust for ambient light levels. Note also that the big white tiles and the small ones are not exactly the same colour. (In fact the big tiles are grey, not white, but the light sensor can’t distinguish the grey and white accurately.)

Note also, your code should work on *any* robot, at *any* level of battery power. The power of the robot’s motors depends on the battery levels, and also varies from robot to robot, so it’s safer to give commands using rotations as units, rather than time.

We will test your algorithm in tutorial time. The idea is that it is completely autonomous during the test: you’ll lose marks for any interventions. The robot must complete the task successfully in two attempts (to demonstrate that its algorithm is robust).

Task 2: Group report (8 marks)

Your group should also write a report about your robot solves Task 1. The report will comment on:

- The algorithms that your robot uses: how they work (in reasonable detail[1]) and why you chose them;
- The problems that you overcame (or not) while coding and testing your robot.

Task 3: Individual report (2 marks)

Each group member must also write an individual report about how the group worked on the assignment. You should say how your group decided to divide the work up, and what your own contribution was. Feel free to discuss any lessons you learned about working in groups! This report can be very brief—it doesn’t need to be longer than a page.

[1] A description of an algorithm is not the same as description of the code. Pasting the code into the report and explaining it line by line is akin to commenting, which may not necessarily explain the algorithm. Think of your reader as a programmer who is not familiar with the Python syntax. Your explanation should be such that this reader could re-implement your algorithm in some other language.

Marking scheme (Total: 20 marks)

Marks will be allocated as follows:

- Task 1: **10 marks**. This task will be assessed by a demo of your robot during your tutorial in Week 5. You will get two attempts, both of them marked. You will be awarded points based on how well the robot performs on different stages of its journey. The exact mark breakdown for each stage is as follow:

Stage	Attempt 1	Attempt 2
Counting	3	3
Finding the tower	1.5	1.5
Pushing the tower	0.5	0.5

  You will lose points for any 'hands of God' (human intervention while the robot is running). Note that for full marks, the robot must perform the task perfectly on both attempts. During the demo, each member of the group should be prepared to answer questions about the Python code the robot is running.

- Task 2: **8 marks**. Marks will be awarded for clarity of the report, your code, and for addressing the topics you need to discuss.

- Task 3: **2 marks**. You'll get full marks if you make it clear how the work was divided up, and what contribution you made.

Naturally, if you didn't make a contribution to your group, you won't share the marks.

Submission

The assignment is due at **4pm on Tuesday of Week 5** (**26th March**). Each group should submit electronically on Blackboard (Assignment 1 Group) a zip file containing the code for Task 1 and ReportGroup.pdf for Task 2. For Task 3, each individual should submit electronically on Blackbaord (Assignment 1 Individual) their individual report.

The demo for Task 1 will be done in your tutorial in Owheo later that week.

For each task, you will lose 10% of available marks for each day late.

Borrowing robots

Starting from Week 2, the robots can be borrowed from Kaye Saunders at the CS department office. Students can book the robots out in three shifts per day:

- Shift 1: 9:00am - 12:00 noon
- Shift 2: 1:00pm - 4:00pm
- Shift 3: 4:30pm - 9:00am the following day.

A team can also check out a robot for the weekend 4:30pm on Friday to 9am the following Monday). See Kaye to book in times to borrow your robot. **If you return your robot late, there is a penalty in the assignment:** you will lose 10% of available marks for each session (day/night) you are late.

There's a lot of demand for robots leading up to the assignment deadline. If you're smart, you will plan on getting the assignment done well before the deadline.