COSC343 Assignment 1

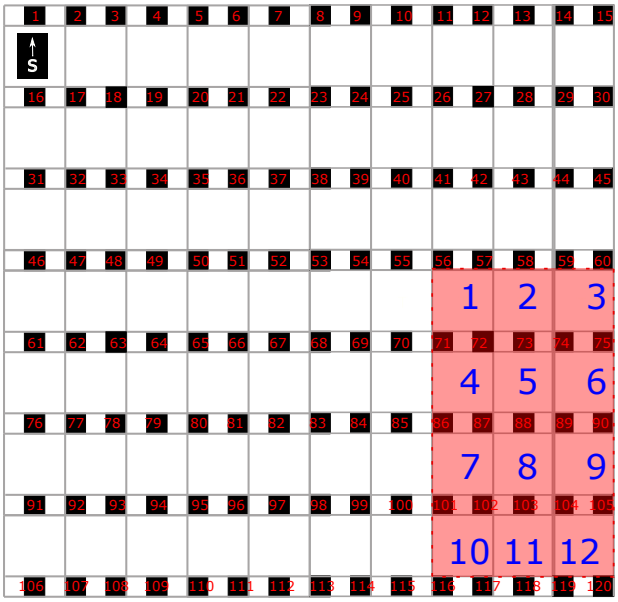
Robots on a chessboard

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 Lech (`lech.szymanski@otago.ac.nz`) to be assigned to an existing group.

Task 1 (10 marks)

Your group's first task is to write a Python program which will allow a LEGO robot to search for a target 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'), facing in the direction of the arrow; it has to reach a plastic tower placed on one of the 12 tiles in the red zone (shown as red overlay in the figure above) . The bot can travel to the tower following any path, but it must continuously report its position any time it passes over a black square (in order to assure the observer that it is not driven by dead reckoning and it knows where it is at any point). Black square are numbered in red in the image above – the robot must report the square number it thinks it's driving over. Note that this is not an exercise in avoiding black squares – the point is to go over them and keep reporting where the robot is. After finding the tower, the robot should play a sound indicating it has finished its task and

report the big tile number where the target has been found. Big tiles are numbered in blue in the image above. Note that the red and blue markings in the image above are just labels for the figure – there will be no red nor blue markings on the floor of the lobby.

Here are all the specifications for the task.

- When driving over the black tiles, the robot must explicitly report the tile number it passes.
- The robot should not avoid the black tiles – the point is to provide regular updates of robot's location
- You can use the sonar sensor to locate the tower, and the bump sensors to detect contact with the tower.
- The robot should not move the target tower off of its square.

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.

[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.

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.

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
Black tile reporting	3.5	3.5
Finding the tower	1.0	1.0
Reporting the correct location of 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** (**30th 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 booked and borrowed from the administrator at the CS department reception. 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). You can book your robot via `http://jet343.otago.ac.nz` (login with your cs username and password is your student number). The website is accessible from the campus network, so if you want to book from home, you'll need to setup an ssh tunnel via `hex.otago.ac.nz`. **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.

Academic Integrity and Academic Misconduct

Academic integrity means being honest in your studying and assessments. It is the basis for ethical decision-making and behaviour in an academic context. Academic integrity is informed by the values of honesty, trust, responsibility, fairness, respect and courage. Students are expected to be aware of, and act in accordance with, the University's Academic Integrity Policy.

Academic Misconduct, such as plagiarism or cheating, is a breach of Academic Integrity and is taken very seriously by the University. Types of misconduct include plagiarism, copying, unauthorised collaboration, taking unauthorised material into a test or exam, impersonation, and assisting someone else's misconduct. A more extensive list of the types of academic misconduct and associated processes and penalties is available in the University's Student Academic Misconduct Procedures.

It is your responsibility to be aware of and use acceptable academic practices when completing your assessments. To access the information in the Academic Integrity Policy and learn more, please visit the University's Academic Integrity website or ask at the Student Learning Centre or Library. If you have any questions, ask your lecturer.

- Academic Integrity Policy
- Student Academic Misconduct Procedures