# Mars Rover Coding Challenge – ThoughtWorks Graduate Scheme

Understanding of requirement and assumptions made:

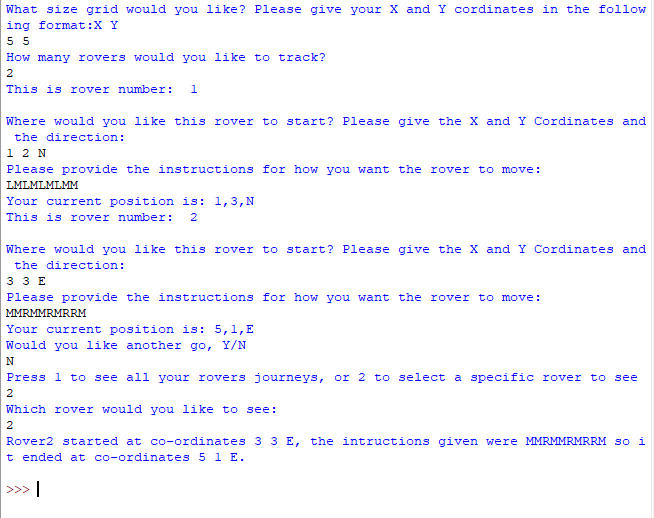
* This program should be able to track ‘rovers’ on a plateau on mars.
* The plateau can be represented by a grid structure.
* The bottom left of the grid will be represented by the coordinates – X = 0, Y = 0 (or 0,0)
* In a 5 x 5 grid, the top right of the grid will be represented by the coordinates = X = 5, Y = 5 (or 5,5)
* The rover can be facing N, E, S, W. I can move one 90 degree turn at a time, either to the left (-90) or to the right (+90)
* The rover can move one space forward in the direction it is currently facing
* The instructions for moving are ‘M’, ‘L’ or ‘R’
* A start position will be provided to the rover, followed by a string of instructions (e.g. MMRMMLMM)
* The end position must then be output to the user
* The next rover can then begin their journey, providing a start position and instructions have been provided

Design:

|  |
| --- |
| **Rover Class** |
| **Initial attributes:**  startX = 0  startY = 0  startDirection = “ “  defaults to these values, setStart handles setting up each rover |
| **Methods:**  setStart – set up start location from user input  execute – takes the user input instructions and calls the appropriate method for each sub string (MLMMR)  turnLeft – takes the current direction updates it (N 🡪W)  turrnRight – takes the current direction updates it (N🡪E)  move – moves the current X, Y coordinates dependent on current direction  returnCurrentPosition – formats string to output the current position |

|  |
| --- |
| **Grid Class** |
| **Initial attributes:**  startX = 0  startY = 0  endX = 0  endY = 0  defaults to these values |
| **Methods:**  setUpGrid – sets the grid end Co-ordinates, default start x & y to 00  upDateGrid – change the end x/y co-ordinates |

|  |
| --- |
| **RoverList Class** |
| **Initial attributes:**  roverList = [] empty list to start with  minSize = 0  maxSize = 0  currentRover = 0  roverRemaining = maxSize - minSize  roversDone = length (roverList) |
| **Methods:**  displayRoverJourneys – handles seeing all completed rover journeys in order. Iterates through list, providing rover num/ start position/ instructions given / end position  displayIndividualRover – allows the user to choose one rover and returns the same information as above but for just one rover |



|  |
| --- |
| **Main program** |
| Create loop to allow for additional testing  User to input grid size  Run grid set up ()  User input the num of rovers  Update max size of list  Iterate through (for loop using 1,max size of list -1)  Output current rover ID  User input the start position  User input the instructions  Update rover start ()  Execute rover commands ()  Return current position ()  NEXT  Ask user if they want to go again, Y/ N  If yes, repeat  If No, ask user if they would like to view all rovers or view one rover  All = rover display all journeys ()  1 = rover display (User given) rover ()  END |