

# Coding Challenge

## The Pawn Simulator

### Description

- The application is a simulation of a Pawn moving on a chess board, of dimensions 8 squares x 8 squares.
- There are no other obstructions/pieces on the chess board.
- The pawn is free to roam around the surface of the table following the rules below, but must be prevented from falling off. Any movement that would result in the pawn falling from the table must be prevented, however further valid movement commands must still be allowed.
- The pawn can move only to adjacent squares but not diagonally. The first time that the pawn moves it can move 1 or 2 squares.
- Create an application that can read in commands of the following form:
  - PLACE X,Y,F,C
  - MOVE X
  - LEFT
  - RIGHT
  - REPORT
- PLACE will put the pawn on the board in position X, Y, facing NORTH, SOUTH, EAST or WEST and Colour(White or Black)
- The origin (0,0) can be considered to be the SOUTH WEST most corner.
- The first valid command to the pawn is a PLACE command, after that, any sequence of commands may be issued, in any order, including another PLACE command. The application should discard all commands in the sequence until a valid PLACE command has been executed.
- MOVE will move the pawn one unit forward in the direction it is currently facing.
- LEFT and RIGHT will rotate the pawn 90 degrees in the specified direction without changing the position of the pawn.
- REPORT will announce the X,Y,F and C of the pawn. This can be in any form, but standard output is sufficient.
- If the pawn is not on the board it needs to ignore the MOVE, LEFT, RIGHT and REPORT commands.
- Input can be from a file, from standard input or through a UI with four buttons and scrolling log of the result, as the developer chooses. (This might depend on the dev stack you use)
- Provide test data to exercise the application.

## Constraints

- The pawn must not fall off the table during movement. This also includes the initial placement of the pawn.
- Any move that would cause the pawn to fall must be ignored.
  - Example Input and Output:
    - i. PLACE 0,0,NORTH,WHITE  
MOVE 1  
REPORT  
Output: 0,1,NORTH,WHITE
    - ii. PLACE 0,0,NORTH,BLACK  
LEFT  
REPORT  
Output: 0,0,WEST,BLACK
    - iii. PLACE 1,2,EAST,BLACK  
MOVE 2  
MOVE 1  
LEFT  
MOVE  
REPORT  
Output: 4,3,NORTH,BLACK

## Deliverables

- The code source files, the test data and any test code.
- All files should be managed under a local GIT repository.
- Your knowledge of good coding practices.
- Your knowledge of software design.

## Time

- This task should take you no more than 4-6 hours to complete.

## NOTE:

- This task should be completed in the tech stack you are applying for.