

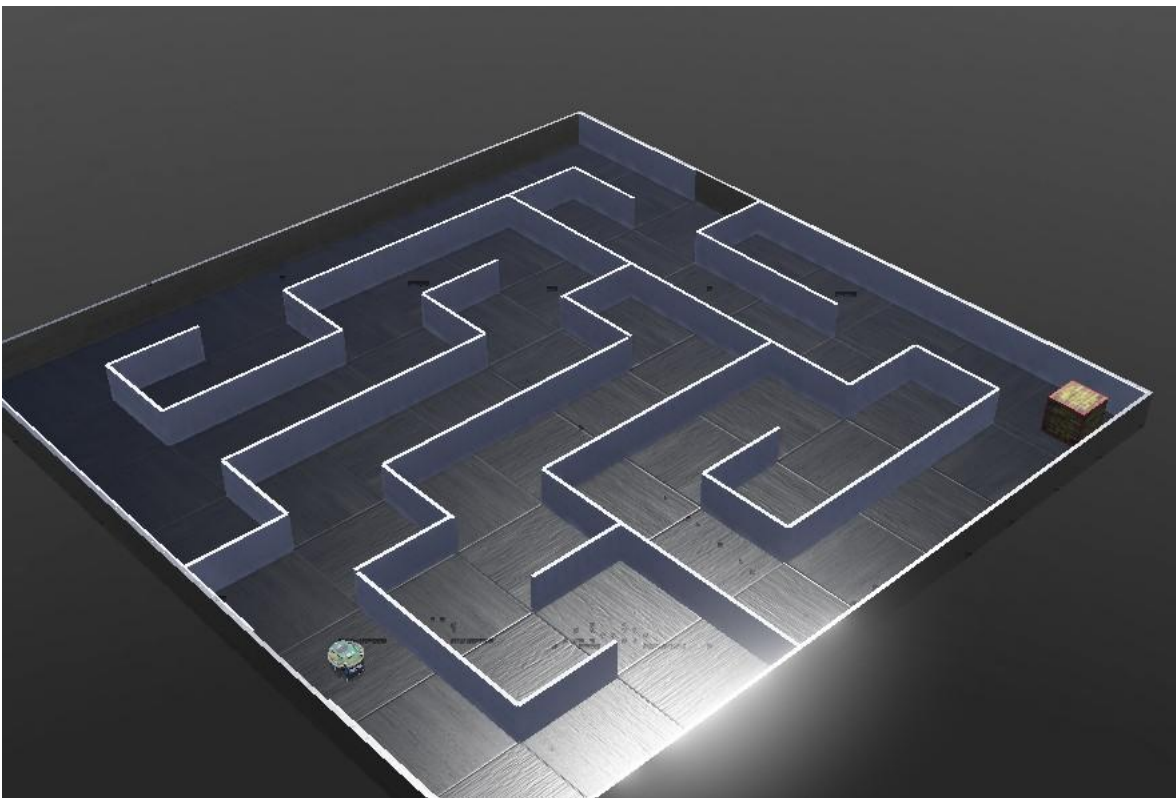


Robotics and Python. Maze Traveller.

Sila, Eaaa. DMU. November 2024.

Maze traveller.

In this exercise we are again looking at an e-puck working its way around a maze.



Again. A number of algorithms might be useful here.

E.g

Hand On Wall Rule, also known as either the left-hand rule or the right-hand rule. If the maze is simply connected, that is, all its walls are connected together or to the maze's outer boundary, then by keeping one hand in contact with one wall of the maze the solver is guaranteed not to get lost and will reach a different exit if there is one; otherwise, the algorithm will return to the entrance having traversed every corridor next to that connected section of walls at least once.

For more, see

https://en.wikipedia.org/wiki/Maze-solving_algorithm

Template code (E-puck, Webot):

Maze_traveller.py (Canvas).

Exercise 1.

In groups of 2-3.

Given the script `maze_traveller.py` the robot will (from its starting position) eventually reach the box. But when the robot has reached its target (the box), the next target is to find its way back home, to the starting position. In the most direct way possible.

In order to find its way back home, the robot must have mapped the environment, along with an understanding of where it is. And where it has been.

So, on reaching the box target, the robot must then use its map of the world to find its way home. As directly as possible. Without using the wall-following algorithm, it used on its way out to the box target.