Edison Robots Navigation Problems

CS 4 the WALC

# Introduction

In this activity you will rescue a robot stuck in a maze. You will rescue the robot by writing and running algorithms that move the robot out of the maze.

If you hit any cups or your robot doesn’t end up on the tape at the end, then you will have to move your robot back to its starting position and try again.

## Materials

Each group will need the following equipment:

* An Android tablet with Edware app installed
* An Edison robot
* An Edware cable for connecting the tablet and robot

## Moving the Edison Robot

There is information about how to write a program that makes the robot move and how to download the program to the robot on the "Edison Robot Quick Reference" handout.

An example program that moves the Edison robot forward for 3 seconds is shown below. Note that the *dual drive* block moves the robot forward and the *event wait* block lets the robot move for 3 seconds. To make the robot move you will always need 2 blocks, the motor drive block and the wait block.



# Maze Key Explanation

Each maze that we have to navigate the robot around will have a picture to show how it should be built, the robot’s original position and where we want the robot to end up.

Each square represents enough space for one cup to be placed. So each square is 1 cup x 1 cup in area.

The squares that are grey are where the cups should be placed.

The green square is the starting position of the robot, and the arrow is facing in the direction that the robot should be facing.

The square with the striped diagonal pattern is where we want the robot to move to.

If you complete all of the mazes before the end of the session split your group into 2. Then each group should take turns in drawing new mazes while the other group solves them.

In the image to the left below, the maze on the left has been set up using cups.

# Maze Images

## Maze #1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  | **↑** |  |  |
|  |  |  |  |  |

## Maze #2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  | **↑** |  |  |  |
|  |  |  |  |  |  |

## Maze #3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | **↑** |  |  |  |  |
|  |  |  |  |  |  |  |

## Maze #4

In this maze, instead of helping the robot escape, we want to move it around the square and back to its original position.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | **↑** |  |  |  |  |
|  |  |  |  |  |  |

Think about the steps that you need to take to move the robot around the maze. Is there any sequence of steps that you need to repeat more than once?

In programming, we can use loops to repeat the same steps multiple times. In Edware the loop blocks are in the Flow (Yellow) section. An example program that uses a loop is shown below. In this program the *dual drive* block is set to move in the *right spin* direction at a speed of 10. When the program runs, the loop makes the *dual drive* block run over and over, making the robot spin continuously. How could you use a loop to make the robot move around the square over and over?

