

Pathfinder!

Solve Problem

1. Understand the problem

- a) Describe the problem
- b) Abstract the problem
- c) Disassemble the problem

2. Solve the problem

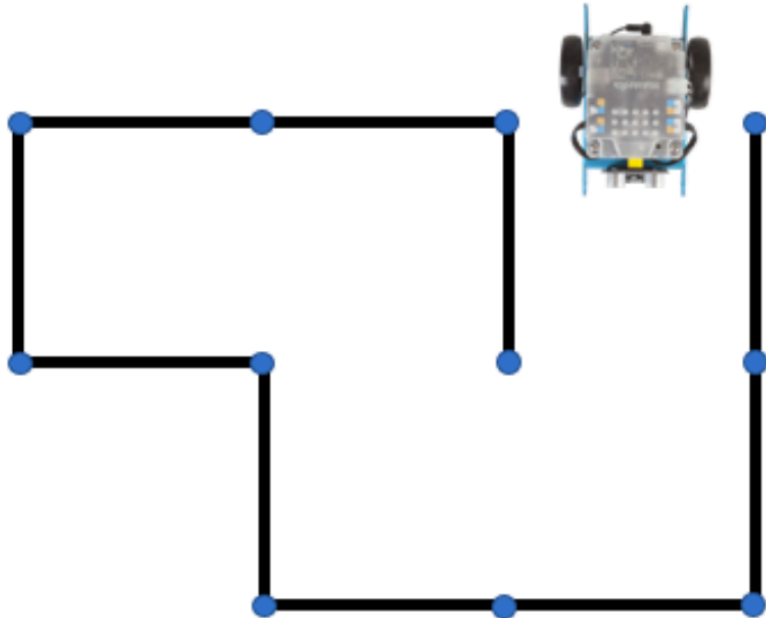
- a) How can (partial) problems be solved?
- b) Implementation in Scratch

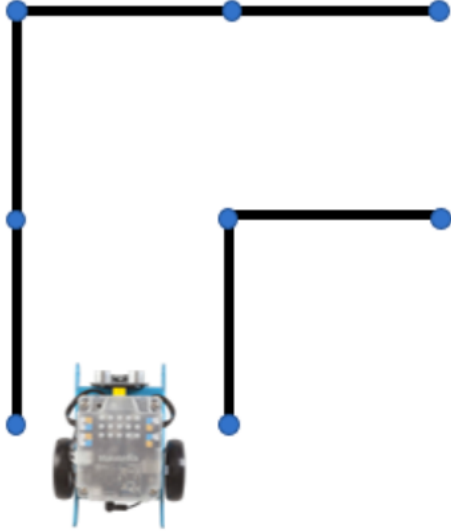
3. Analyze the problem

- a) Test the program
- b) Debug the program
- c) Transferred to other mazes

Problem: The mBot is to find its way out of a maze independently and as quickly as possible!

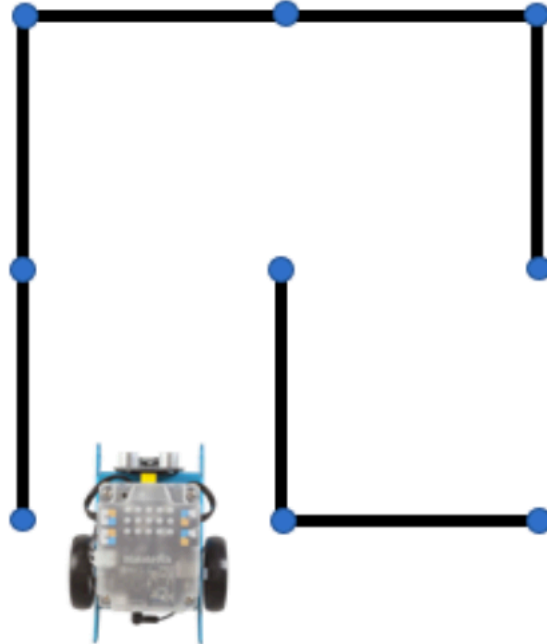
Program the mBot in such a way that it drives out of an unknown maze independently and as quickly as possible!



Problem 1: Turn right!	
<p>Program the mBot so that it turns straight right at a wall and continues driving (see illustration on the right).</p>	
1. Understand problem	
<p>a) Describe the problem briefly, in general terms, in your own words - without thinking about the specific mBlock programme.</p> <p>b+c) Abstract and decompose the problem by considering what information you need during the journey so that the robot turns right in this course.</p>	
2. Solve problem:	
<p>a) Describe how to solve the problem (e.g. required sensor, required programme components...).</p>	
<p>b) Implement your solution in mBlock.</p>	
3. Analyze problem	
<p>Test your programme using the mBot and improve both the mBlock programme and your solution on the worksheet if necessary.</p>	

Problem 2: Turn right and left!

Program the mBot so that it turns either left or right when it sees a wall ahead - depending on how the maze is constructed!



1. Understand problem:

- a) **Describe** the problem briefly, in general terms, in your own words - without thinking about the specific mBlock programme.
- b+c) Abstract and decompose the problem by considering what information you need during the journey so that the robot can turn right and left.

2. Solve problem

b) **Implement your solution in mBlock.**

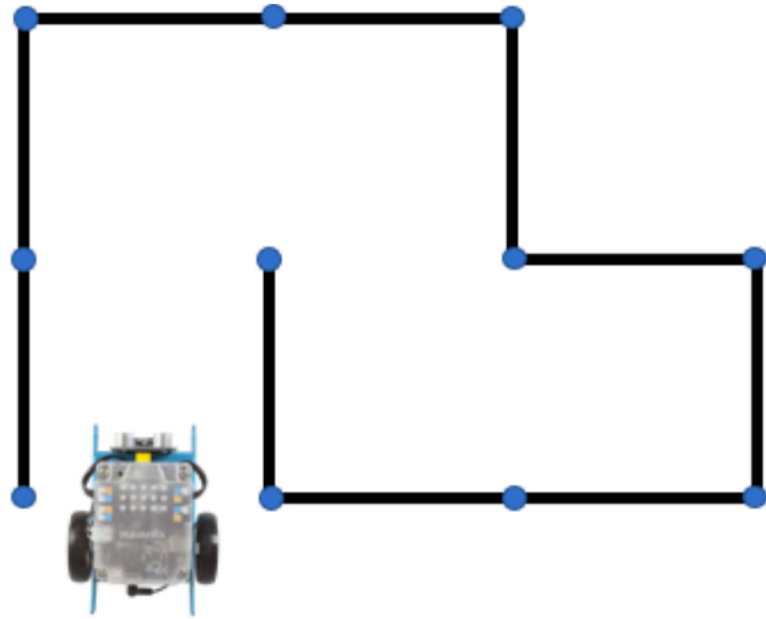
3. Analyze problem

Test your programme using the mBot and improve both the mBlock programme and your solution on the worksheet if necessary.

Problem 3: Dead end? No problem!

Program the mBot to find its way out of the maze even if there is a dead end!

The **maze competition** is as follows: The fastest robot wins. If the robot has to be reset manually, 10s are added to the final time!



1. Understand problem:

- a) **Describe** the problem briefly and in general in your own words - without thinking about the concrete mBlock programme.
- b+c) Abstract and decompose the problem by considering what information you need during the journey so that the robot moves through the maze as quickly and completely independently as possible!

2. Solve problem

- a) **Describe** how to solve the problem (e.g. required sensors, required program modules...).

b) Implement your solution in mBlock.

3. Analyze problem

Test your programme with the mBot and improve both the mBlock programme and your solution on the worksheet if necessary.

If you still have time, improve your programme by having the mBot recognise white dots on the track, for example, and show this on the LED display! Then you get 20s deducted! Or think of a suitable addition yourself - there are no limits to your creativity!