

# ROBO Exercise Set 1

Arvid Halma  
www.robomind.net

June 8, 2011

In this set of exercises you'll find puzzles of varying degrees of difficulty. Some of them are fairly easy, others require experience in solving problems of this kind. Its purpose is to inspire teachers for creating their own exercises.

## 1 Move beacon

### 1.1 Using the remote control

Use the remote control (Run ▷ Remote Control) to move the beacon located south east of the robot to the south west corner.



### 1.2 Write the script yourself

Try to do the same, but then by typing the command's yourself and running the script. In contrast to the code the remote control generates,

```
1 ...  
2 forward(1)  
3 forward(1)  
4 forward(1)  
5 ...
```

you can write

```
1 ...  
2 forward(3)  
3 ...
```

## 2 Drawing stairs

### 2.1 Using the remote control

Use the remote control to draw stairs like the one below:

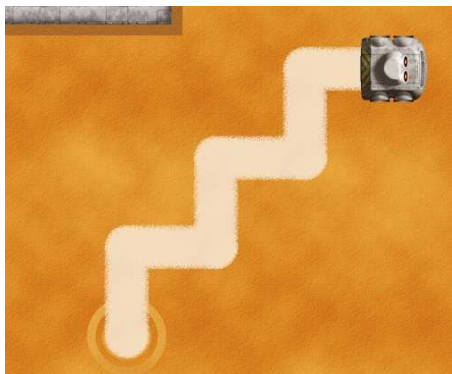


Figure 1: Stairs with three steps.

### 2.2 Using a loop

Use a `repeat` loop to draw the three steps. Hint:

```
1 # Start to paint  
2 repeat(3)  
3 {  
4     # Draw a single step  
5 }  
6 # Stop painting
```

### 2.3 Using a procedure

Create a procedure `drawStairs()` that draws the three steps like the previous exercise. Call this procedure to make sure it actually starts drawing.

### 2.4 Generalize this procedure

Make this procedure more widely applicable by changing it to draw stairs with an arbitrary number of steps. This procedure takes an argument that you may call

`number`, so you can define the number of steps to be drawn in the procedure call.

Hint:

```
1 drawStairs(2)
2
3 procedure drawStairs(number)
4 {
5     # draw stairs with 'number' steps.
6 }
```

draws stairs with two steps. By changing line 1 to `drawStairs(5)` it makes the robot draw five steps.

### 3 Follow the white marks

In the map *goRightAtWhite1.map* you'll find a number of white painting marks. Those can be followed up by moving forward until you see one right in front of you and then by going right at this spot. You'll find another beacon right in front of you, until you hit a beacon. your goal is to get this beacon. Write a single script that works in the maps: *goRightAtWhite1.map*, *goRightAtWhite2.map*, *goRightAtWhite3.map*.



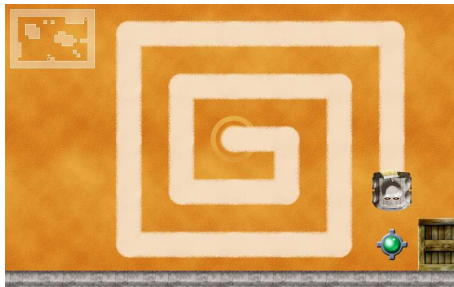
### 4 Draw a Spiral

#### 4.1 First version

Write a script that draws a spiral until you can't go any further because there is an obstacle in front of you.

#### 4.2 Second version

Create a script that lets the robot perform the same behavior, but now by using recursion to draw the arms of the spiral. Hint: first make a procedure that lets the robot move along one arm of the spiral.



## 5 Rewriting repetitions

The following script makes the robot move forward until it encounters an obstacle.

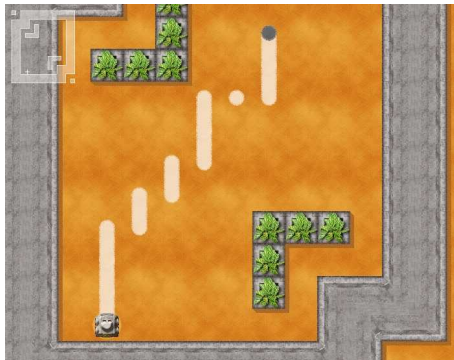
```

1  repeatWhile(frontIsClear())
2  {
3      forward(1)
4  }
```

rewrite this code in such a way that it doesn't use `repeatWhile(...)` any more, but uses `repeat()` and `if(...)` instead.

## 6 Jump over from line to line

Create a program that makes the robot move to the grey spot in the map *verticalLines.map*.

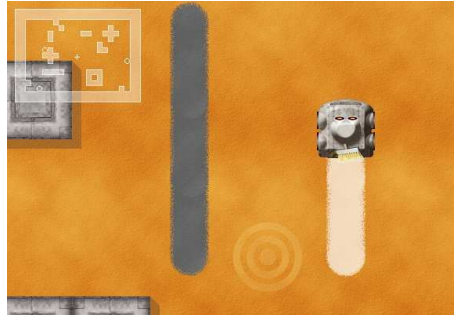


## 7 Copy a line

### 7.1 Make an equally long line

In the map *copyLine1.map* there is a black line at the left of the robot. Your goal is to draw a parallel line of equal length at the right of the robot. Do this in such a way

you don't have to know in advance how long this line is, by doing it step by step.



## 7.2 Make a twice as long line

Use the same map to draw a twice as long line. This is far more complex, since you don't know which part you already did while copying. You'll have to find a trick to overcome this. Hint: don't only draw the copy, but paint marks to keep track of the progress. Again, make a script that works for lines of any length.