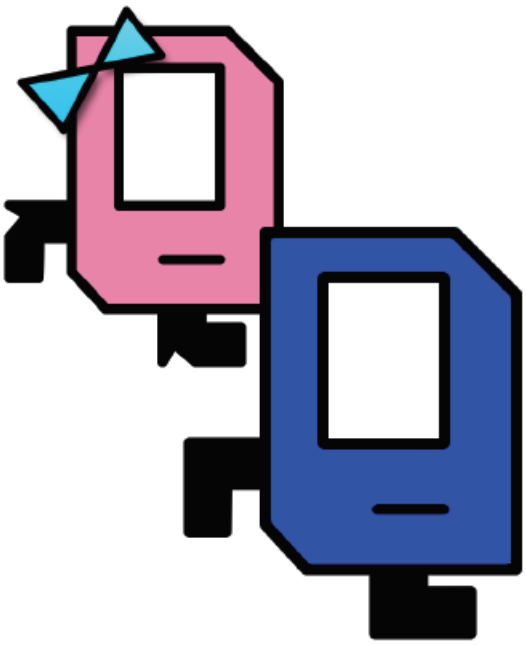
www.karelthe robot.ch

Karel & Karin The Robots

USER MANUAL

An educational tool for teaching programming



Ing. Inf. Dipl. ETHZ Philip Hubert
Dott. Prof. Eliana Imperatore
Edizione agosto 2020

Statements

move()

Move Karel one box in the direction in which he is oriented. By convention we will use as directions the cardinal points (N,S,E,W) to define the direction in which Karel is oriented. Be careful if Karel hits a wall or the edge of the layer the instruction generates an error.

turnLeft()

Make Karel rotate 90 degrees to the left (counterclockwise).

turnRight()

Make Karel rotate 90 degrees to the right (clockwise).

turnAround()

It performs a 180-degree Karel rotation, a U-turn.

If (<condition>) { <if true> }
If (<condition>) { <if true> } else { <if false> }

This instruction allows Karel to execute a certain block of instructions if a certain condition is true. The following instruction moves Karel only if it has no wall or level edge in front of it.

if (frontIsClear()) { move(); }

putBeeper()

Karel is capable of depositing beepers in his path. You don't have to collect any beforehand. The putBeeper() instruction is able to create beepers every time it is invoked. Each box can accept more than one beeper.

pickBeeper()

Karel picks up the beeper underneath him. Be careful if there are no beepers at Karel's position the instruction generates an error.

while (<condition>) {}

This instruction allows Karel to repeat a set of instructions until a certain condition is true. For example, you can write to orient Karel to the north:

while (notFacingNorth()) { turnLeft(); }

repeat (<n>) {}
repeat (<full number>) { < instruction block> }
This instruction allows Karel to repeat a set of instructions a defined number of times. If we wanted to move Karel by making him run a square counterclockwise with the measurement of the side equal to 3 squares the program would be:

repeat(4) { move();move();move();turnLeft(); }

Note that this program could be rewritten as:

repeat(4) { repeat(3) move();turnLeft(); }

Also note that if the instruction block consists of only one instruction, the curly brackets can be omitted. We could have written anyway:

repeat(4) { repeat(3) { move(); } turnLeft(); }

Conditions

Conditions are special instructions that return a true or false state. They are crucial for Karel to make choices along the way.

beepersPresent()
Returns true if there is at least one beeper in the position where Karel is, otherwise it returns false.

noBeepersPresent()
Returns true if there are no beepers in the position where Karel is, otherwise returns false.

facingNorth(), facingSouth(), facingEast(), facingWest()
Returns true if Karel is oriented in the respective direction otherwise returns false.

notFacingNorth(), notFacingSouth(), notFacingEast(), notFacingWest()
Returns true if Karel is not oriented in the respective direction otherwise returns false.

frontIsBlocked()
Returns true if the box in front of Karel is a wall or if it has reached the edge of the level, false otherwise.

frontIsClear()
Returns true if the box in front of Karel is free (not a wall or level edge), false otherwise.