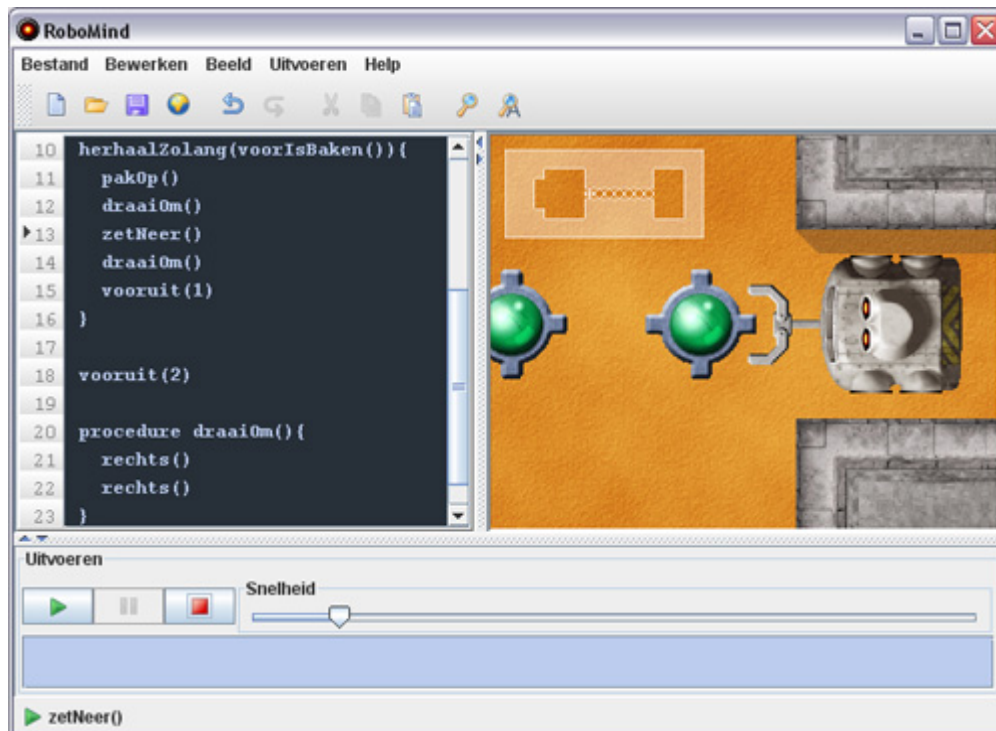


## A first impression of programming with RoboMind



- Machines and Instructions
- Writing programs



# Instructions

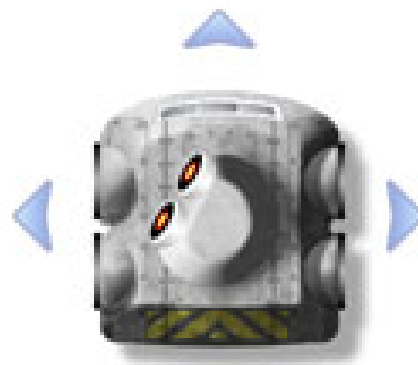
- If you want to make a machine, computer or robot work, you have to give it *instructions*
- Instructions differ from machine to machine and can be anything. For example:
  - Video recorder: *record, play, pause*
  - Web browser: *go to site, go back, print page*
- Question: What could be instructions for an automatic pilot?

# Basic instructions

- Every machine has a set of basic instructions: actions it can perform directly.
- The robot we're going to program has basic instructions to:



Move



See



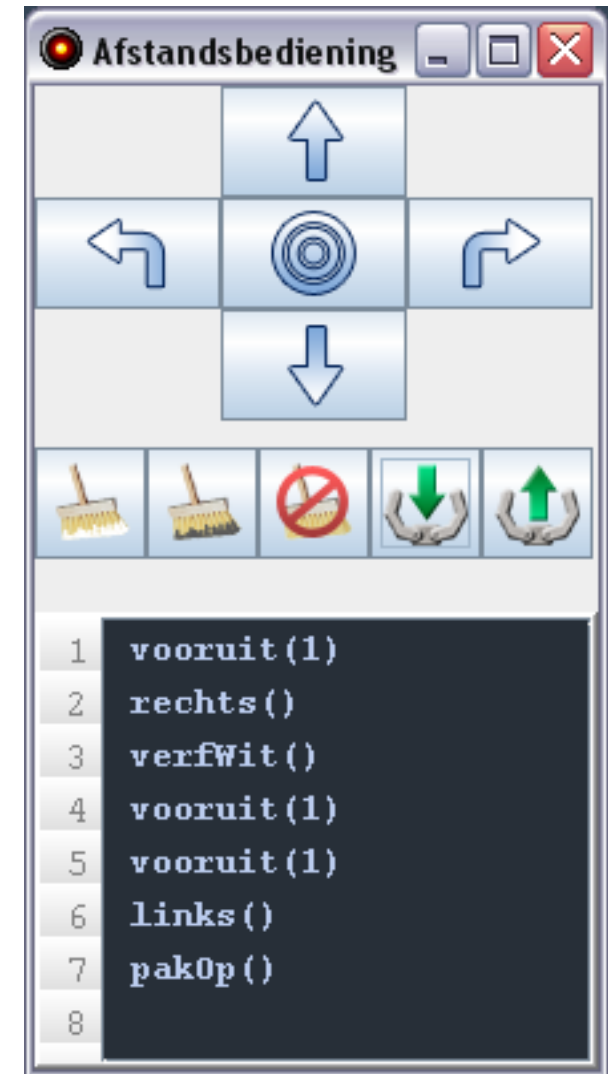
Paint



Grab

- Instructions can be given by
  - Pressing a button
  - Dragging your mouse
  - Writing commands
  - ...

- Open the remote control in RoboMind  
(Execute > Remote Control)
- Click the buttons to make the robot act
- Exercise: Try to make the robot drive a small square



# Writing instructions

- You can also write instructions
- Moving can be done with the commands:  
`forward(1)`, `backward(1)`, `left()`, `right()`

The screenshot shows the RoboMind application window. On the left, a code editor displays the following code:

```

10 herhaalZolang(voorIsBaken()) {
11     pakOp()
12     draaiOm()
13     zetNeer()
14     draaiOm()
15     uit(1)
16 }
17
18 vooruit(2)
19
20 procedure draaiOm(){
21     rechts()
22     rechts()
23 }
    
```

On the right, a 3D simulation shows a robot on a platform with two green sensors. A callout bubble points to the robot with the text "3. See the result".

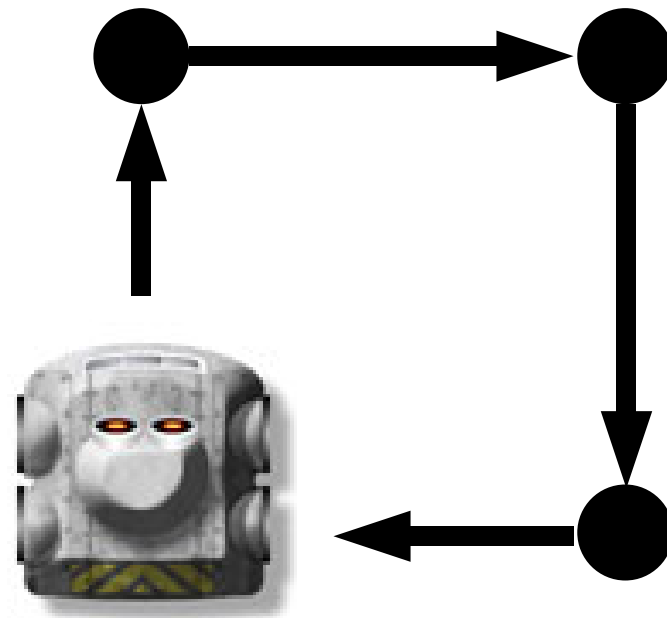
At the bottom, the "Uitvoeren" (Execute) panel contains a play button, a stop button, a speed slider labeled "Snelheid", and a status bar showing "zetNeer()". A callout bubble points to the play button with the text "2. Click execute".

Another callout bubble on the left points to the code editor with the text "1. Write instructions".

# Example: Square

- Write the following movement instructions:

```
forward(1)
right()
forward(1)
right()
forward(1)
right()
forward(1)
right()
```



Script 1



# Writing seems slower

- Writing instructions seems to be more work than using the remote control
- However, it is more convenient if you want to automate the process of giving instructions
  - You can execute the instructions more than once, without writing the script again or clicking the buttons on the remote control
  - Execution goes faster. You don't have to find the buttons for giving instructions.

# Extra instructions

- Using extra instructions results in less typing
- Use for instance :

`repeat (number) {instructions}`

```
forward(1)
right()
forward(1)
right()
forward(1)
right()
forward(1)
right()
```

Script 1

```
repeat(4)
{
  forward(1)
  right()
}
```

Script 2

Same square,  
less work

# Question: Two squares

- How can you make the robot drive two squares?

```
repeat (4)
{
  forward (1)
  right ()
}
```

Script 2

# Answer: Two Squares

- It can be done in several ways:

```
repeat (8)
{
  forward (1)
  right ()
}
```

Script 3

```
repeat (2)
{
  repeat (4)
  {
    forward (1)
    right ()
  }
}
```

Script 4

Less boring than 16 mouseclicks in the remote control!

- How do you know what instructions you're allowed to use?
  - like: `forward(1)`, `left()`, `repeat`, ...
- You don't know in advance. It depends on the rules of the programming language.
- You'll have to read the documentation to find this out. There are many different programming languages.

- This is the end of this presentation
- Open the example scripts to explore the possibilities.
- Exercise: program a "dance" for the robot with repeat-loops.

**Tip:** Use `leftIsClear()` and `rightIsClear()` to move the head of the robot.