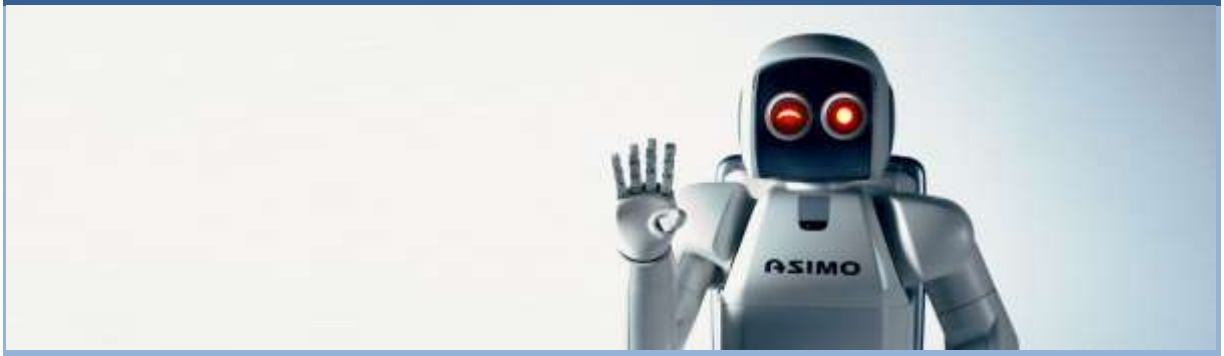
RoboMind Challenges

Getting Started

Learn about the RoboMind programming environment



Difficulty: ★☆☆ (Easy), Expected duration: an afternoon

Description

This activity uses RoboMind, a robot simulation environment, to present the fundamental concepts of robotics. No prior experience with Robotics is necessary! There is some simple programming involved, but don't be afraid to give it a try even if you have never programmed before. The activity instructions will show you everything you need to know!

In this activity, you will:

- Become familiar with RoboMind, a robot simulation environment
- Command a robot to move where you want it to go
- Teach the robot to sense its surroundings, so it doesn't crash
- Make the robot perform a simple task for you (pick up and move an object)

Make sure you have the latest software

You need the RoboMind software to complete this activity. If you have done other RoboMind activities, you should already have the software installed on your system. If not, download and install the software that is available on www.robomind.net.

Introduction

This exercise introduces you to the fundamental concepts of robotics, and prepares you to do other activities with robotics. You will learn how to program a robot using the RoboMind robot simulator environment. Don't worry if you don't have any programming experience, RoboMind is very friendly and we will walk you through it!

Copyright © Research Kitchen, Freshbrain, Hill Air Force Base

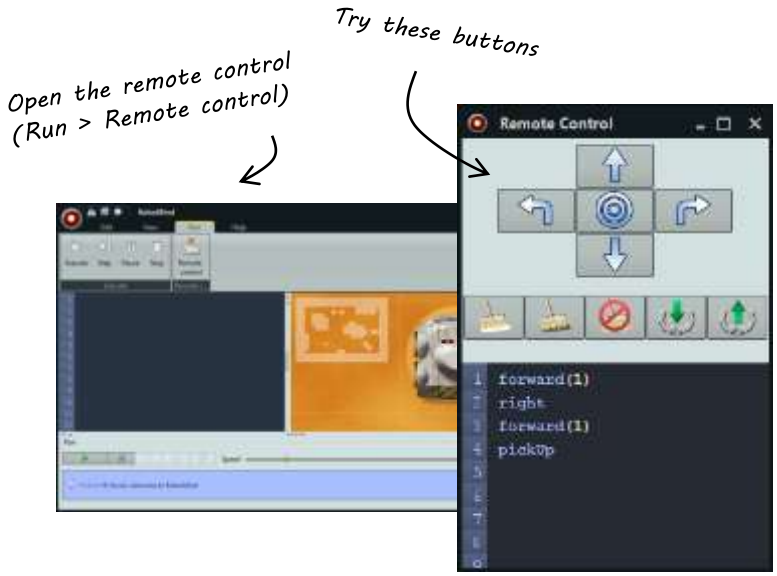
By completing this exercise you will learn:

- Movement. Robots aren't very useful if they can't move! You will learn how to control the direction and motion of your RoboMind Robot.
- Sensing. Robots need to sense their surroundings, so they don't bump into things (or walk off of a cliff!)
- Interacting with the Environment. Robots are often used to move items from one place to another. Robots are used in office buildings to deliver mail to offices. Robots are also used in warehouses to gather a collection of items for a shipment.

Getting Started

You must have RoboMind installed to do this exercise. If you haven't yet installed RoboMind, visit the RoboMind installation page and follow the instructions to install and start RoboMind.

Go to the Run menu, and select Remote control. This will pop up a small window with controls for moving the robot (forward/back), turning the robot (left/right), start painting a white or black line, stop painting, pick up a beacon, and put down a beacon. There is also a control to reset. Note that as you move around, commands are printed in the Remote Control window. This is an easy way for you to learn some of the basic programming language of RoboMind. Experiment a bit, drive the robot around, paint some lines on the floor, find the beacon, pick it up, and drive around and put it down somewhere else. When you feel comfortable with these the basic commands, click the reset button [] on the Remote Control to clear your map and put your robot back at the starting point. Close the Remote Control window.



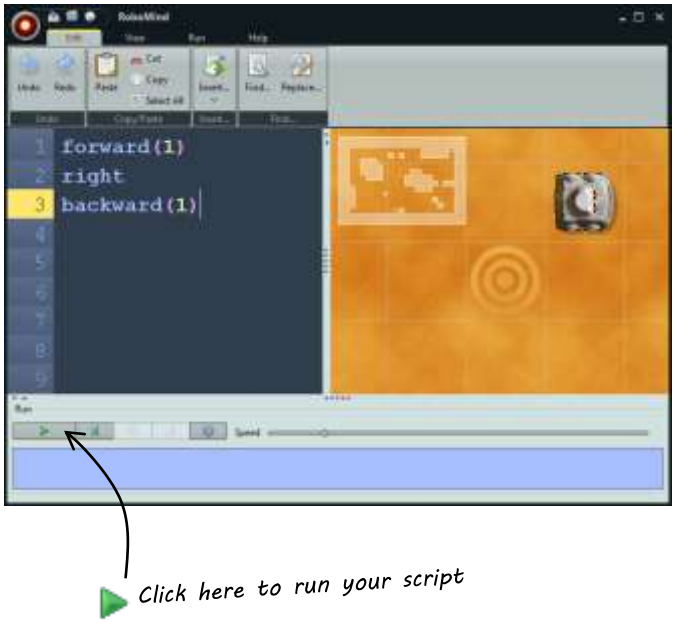
Movement

Now you will write some of the code yourself. In the main RoboMind window, click in the dark blue area; this is the text editor window, where you write and edit your code. Type in some of the commands that you learned when you were using the Remote Control. Note that the robot does not move while you type the commands.

In the main window, creating the commands and running the commands are separate steps. To run your code, click on the green run button in the Run area below the text editor. One of two things should happen: your

Copyright © Research Kitchen, Freshbrain, Hill Air Force Base

robot executes your commands, or an error message is displayed in the area below the run button. If you have errors, correct them, and click the Run button again.



If you have trouble remembering some of the commands, you can either consult the documentation, or look under the Edit menu, at the Insert ... item. This allows you to insert code into your script without typing it, and helps avoid errors. Don't worry if you don't understand all of the options under Insert ... at this time. If you want to reset the map to start over, go to the File menu, select Open map, and in the map file browser that pops up, select default.map.

Sensing

You may have noticed that if there is an obstacle (such as a wall) in front of your robot, and you try to move forward, your robot will bump into the obstacle and bounce off of it. If this were a real robot, such a collision could cause damage to the environment (in this case, the wall), the robot, or both. Robots often have built in sensors to determine when things are in their way, or to keep track of where they are. In RoboMind, the robot has functions that allow it to "see" its surroundings.

These functions can be found in the documentation, or under Edit > Insert... > See. The robot has the ability to see obstacles in front of or beside it, and also white or black lines or spots on the floor. In order to use these sensing functions, you will need to learn to use the `if()` condition of the RoboMind programming language. Select Edit > Insert... > Conditions > if() { }, this will insert an `if()` statement into your text editor. The `if()` statement evaluates the code inside of the () (the "condition"), and if it is true, it runs the code that you put inside of the {}. So, for example, if you wrote:

```
if (frontIsClear)
{
	forward(1)
}
```

your robot would make sure there is no obstacle in front of it, and if so, move forward 1 square. Try this now; create a script that has 10 lines, each line has the `if(frontIsClear){ forward(1)}` statement. Run

Copyright © Research Kitchen, Freshbrain, Hill Air Force Base

the code. You will see that the robot walks forward, and when it gets to a wall, it stops without bumping into it.

Great, but in general, how do we know how far the robot will have to walk before it comes to a wall? How do we know how many copies of the if-statement to put into the script? Well, we don't, but this is where a programming concept called looping can be used. With the if-statement, the condition is tested once, and if true, the body of the if-statement (the code inside the {}) is executed once. With a loop, the body will be executed repeatedly, as long as the condition is true.

Delete all of the `if`-statements from your script, and replace them with the single line:

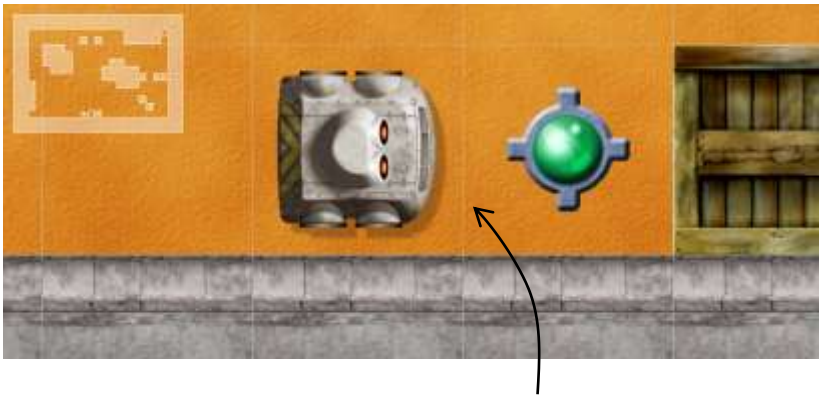
```
repeatWhile(frontIsClear)
{
        forward(1)
}
```

The `repeatWhile` command can be found under Edit > Insert... > Loops. This code checks to see if front is clear, and if so, moves forward 1; then it does it again, and again, until `frontIsClear` is false (that is, an obstacle is in front of the robot). The nice thing about this is that the robot can go in any direction, and it will keep walking forward until it reaches an obstacle. You don't have to worry about making sure you add enough copies of the `if`-statement.

Interacting with the Environment

At this point, your robot knows how to wander around without hurting itself or anything in its path. Next, we will teach it to interact with objects in its environment.

The RoboMind system provides objects called beacons that the robot may sense, pick up, and put down. There are functions that allow the robot to see the beacons (`frontIsBeacon, leftIsBeacon, rightIsBeacon` and can be found in the Edit > Insert... > See menu and are mentioned in the documentation), and also to pick them up (`pickUp`) and put them down (`putDown`). On the default map, you will see a blue beacon two squares below and three squares to the right of the robot's starting position. Let's write some code to pick up the beacon and move it.



The robot is in position to pick up a beacon

Copyright © Research Kitchen, Freshbrain, Hill Air Force Base

First, we need to get to the same row as the beacon. We can easily just write `backward(2)`, and that will work, but it will only work for this starting position. Let's be more clever, and write code that will work if the robot's starting position were moved.

We know the beacon is below the robot. RoboMind provides a function, `south`, that allows you to make the robot head south, regardless of what direction it is currently headed. The `south` needs to be passed a number, telling the robot how many squares it should move once it is heading south. So we could do `south(2)`, but again, this only works as long as the beacon is always 2 squares south of the starting position. If you do `south(0)`, the robot will turn and head south, but won't move. At this point, we have him heading towards the wall. Now, use some of what you learned above to make the robot walk up to the wall, and stop before it bumps into the wall — remember your loops here!

Now we know the beacon is to the east. Turn the robot east (`east(0)`), and then walk until the beacon is right in front of the robot — remember the `frontIsBeacon` function. Once the robot is right next to the beacon and facing it, you can use the pickUp() code to pick up the beacon. The robot will reach out with its grabber and pick up the beacon. Now you can carry it wherever you want! Write some code to move somewhere else in the room and put the beacon down, again, being careful not to bump into anything.

Congratulations!

You now have experience with the basics of robotics: movement, sensing, and interacting with the environment. And if you haven't done any programming before, you've gained some programming skills as well!

The last thing to do is to save your script. Go to the File menu, and select Save. Pay attention to where it is saving the script, as you will need to find it later to submit it as your solution! Give the script a descriptive name, like MoveTheBeacon or MyFirstRobot or whatever will help you remember what the script does.

Copyright © Research Kitchen, Freshbrain, Hill Air Force Base