

Robocode Maneuvers

In this exercise you will:

1. Learn about moving the robot.
2. Learn about detecting other robots (an event) and firing.

Robocode Setup

As stated in Lab 9, you will need to install the Robocode files on your lab computer for every Robocode lab session. Follow the instructions in Lab 9 under "Installing Robocode" and "Running Robocode".

Maneuvering a robot

Your robots must be told what to do. The robot from the previous lab was told to go ahead at a set distance of 100, rotate its gun 360 degrees, go back at a distance of 100, and rotate its gun 360 degrees again. The statements used were

```
ahead(100);
turnGunRight(360);
back(100);
turnGunRight(360);
```

`ahead`, `turnGunRight`, and `back` are methods that can be called to make the robot perform a maneuver. The `ahead` and `back` methods are passed a value that determines the distance to move. The `turnGunRight` method is passed a value that determines the degree of rotation of the gun.

Two other methods that can be used are `turnRight` and `turnLeft`. These methods are passed a value that determines the degree to turn with respect to the robot's current heading.

```
turnRight(90) // Turn the robot right 90° from its current heading
turnRight(30) // Turn the robot left 30° from its current heading
```

Experiment 3.12

Step 1. Run the robot J03E12 in a battle by selecting Battle > New. Add the lab3.J03E12 robot to the battle and click Start Battle. Observe the robot's behavior.

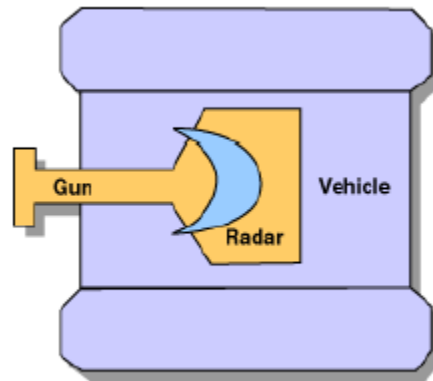
Step 2. Open the source code for the robot J03E12 by selecting Robot > Editor. In the robot editor, select File > Open. Double click on the lab3 directory and select and open J03E12.java. The behavior of the robot is determined by the statements in the `run` method.

The `faceNorth` method turns the robot to face north. The `ahead` and `turnRight` methods have been described above.

Step 3. Make the robot move in a square once by placing the last two statements (`ahead` and `turnRight`) in a loop. To simulate the robot moving in a square, these two statements should be executed four times. Any loop (`for`, `while`, `do-while`) will work, but a `for` loop would fit well here.

On the offensive

In addition to a gun, each robot is equipped with a radar. This radar continuously scans the battlefield for other robots.



Every robot that is detected by the radar will be reported to the robot. It is up to the programmer to determine what course of action to do if another robot is detected. When a robot is detected, statements contained in the `onScannedRobot` method will be executed.

Experiment 3.13

Step 1. Run the robot J03E13 robot in a battle with another robot, e.g. SittingDuck, Crazy, etc. Observe its behavior. In addition to moving in a square continuously, the robot fires when its radar detects other robots.

Step 2. To visually see the range of the radar, select Options > Preferences. Check the box beside “visible scan arcs” and click Finish. If you did this during a battle, when the battle resumes, you can see the range of the scans of the J03E13 robot’s radar. If not, start a new battle. Every time another robot enters into the range of the radar, the J03E13 robot will fire a shot.

Step 3. Open the J03E13 robot in the robot editor. The `while` loop

```
while (true)
{
    ahead(100);
    turnRight(90);
}
```

in the `run` method is used to determine what the robot will do on the robot’s turn. This is a special loop in Robocode, where all statements in this loop will be executed every time it is

the robot's turn. The loop will terminate only when the robot dies or the battle round is finished.

At the bottom of the file is an `onScannedRobot` method. As mentioned earlier, this method will automatically be called if another robot is detected by the radar. When this method is called, all statements in this method will be executed.

Step 4. Add your own method calls or change the values that are passed to the statements in the both the `while` loop and `onScannedRobot` methods mentioned in Step 3. Compile and run the updated robot in a battle.

FYI: Events

The detection of a robot by the radar is just one “event” in Robocode. The following are other “events” in Robocode. The methods related to the events are given in parentheses.

- When the robot is hit by a bullet (`onHitByBullet`)
- When the robot hits another robot (`onHitRobot`)
- When the robot hits the wall (`onHitWall`)
- When the robot wins the battle round (`onWin`)
- When the robot dies (`onDeath`)

Just like the `onScannedRobot` method is called when a robot is detected by the radar, the methods given in parentheses will be called when the “event” listed beside it occurs on the robot. If you would like the robot to perform a task when an “event” occurs, then you will need to add the corresponding method (and the statements in this method) in your robot's source code file, just as was seen in the previous experiment's robot.