

# Robocode/My First Robot

## From RoboWiki

< Robocode

This is the classic **My First Robot Tutorial** that tells how to create your first robot.

### Contents

- 1 Creating a Robot
- 2 My First Robot
  - 2.1 The Robot Editor
  - 2.2 A New Robot
  - 2.3 Let's move somewhere
  - 2.4 Fire at Will!
  - 2.5 Compile your robot
  - 2.6 What's next?
  - 2.7 See also
    - 2.7.1 Robot API
    - 2.7.2 Tutorials
    - 2.7.3 News and Releases
    - 2.7.4 Home pages

## Creating a Robot

Here's the fun stuff. This is what Robocode is all about!

Creating a robot can be easy. Making your robot a winner is not. You can spend only a few minutes on it, or you can spend months and months. I'll warn you that writing a robot can be addictive! Once you get going, you'll watch your creation as it goes through growing pains, making mistakes and missing critical shots. But as you learn, you'll be able to teach your robot how to act and what to do, where to go, who to avoid, and where to fire. Should it hide in a corner, or jump into the fray?



# My First Robot

Ready to create your first robot? I hope you'll find that it's easy, straightforward, fun, and addictive!

Robocode ships with a number of sample robots that you can look at for ideas, and to see how things work. You can use the Robot Editor to look at all of them.

In this section, we'll use the Robot Editor to create your very own, brand new robot.

## The Robot Editor

The first step is to open up the Robot Editor. From the main Robocode screen, click on the **Robot** menu, then select **Editor**.

When the editor window comes up, click on the **File** menu, then select **New Robot**.

In the dialogs that follow, type in a name for your robot, and enter your initials.

Voila! You now see the code for your own robot.

## A New Robot

This is what you should be looking at (names have been changed to protect the innocent):

```
package man;
import robocode.*;

public class MyFirstRobot extends Robot {
    public void run() {
        while (true) {
            ahead(100);
            turnGunRight(360);
            back(100);
            turnGunRight(360);
        }
    }

    public void onScannedRobot(ScannedRobotEvent e) {
        fire(1);
    }
}
```

We're only concerned with the bits in **bold** here... you won't need to change anything else. Not that much, right?

By the way, if you're REALLY concerned about the rest of it, right now, I describe it here:

```
package man;
import robocode.*;

public class MyFirstRobot extends Robot {
    public void run() {
    }
}
```

<b>import robocode.*;</b>	Tells Java that you're going to use Robocode objects in your robot.
<b>public class MyFirstRobot extends Robot</b>	Tells Java: "The object I'm describing here is a type of Robot, named MyFirstRobot".
<b>public void run() { }</b>	The game calls your run() method when the battle begins.
<b>{ }</b>	"Curly brackets" ( { } ) group things together. In this case, they're grouping together all the code for the robot.

## Let's move somewhere

Let's add a couple lines so that it will do something.

First, we'll examine the run() method:

```
while (true) {
    ahead(100);
    turnGunRight(360);
    back(100);
    turnGunRight(360);
}
```

`while(true) { }` means: "While the condition `true` is true, do everything between the curly brackets `{ }`".

Since `true` is always true (no kidding? ;-), it means: "Do the stuff inside my curly brackets, forever".

So this robot will:

1. move ahead 100 pixels
2. turn the gun right by 360 degrees
3. move back 100 pixels
4. turn the gun left/back by 360 degrees

The robot will continue doing this over and over and over, until it dies, due to the `while(true)` statement.

Not so bad, right?

## Fire at Will!

When our radar scans a robot, we want to fire:

```
public void onScannedRobot(ScannedRobotEvent e) {
    fire(1);
}
```

The game calls your **onScannedRobot** method whenever you can see another robot. It sends along an event that can tell us lots of information about the robot -- its name, how much life it has, where it is, where it's heading, how fast it's going, etc.

However, since this is a simple robot, we're not going to look at any of that stuff. Let's just fire!

# Compile your robot

First, save your robot by selecting the **Save** in the **File** menu. Follow the prompts to save your robot.

Now, compile it by selecting **Compile** in the **Compiler** menu.

If your robot compiles without any errors, you can start a new battle with your robot. Start a new battle by selecting **New** in the **Battle** menu. If you cannot see your robot, you might have to refresh the list of robots by **pressing F5**. Add your robot to the battle together with at least one other robot as e.g. sample.Target, and press the **Start Battle** button to let the games begin!

## What's next?

You should have a look at all the sample robots to see how certain things are done.

You'll eventually want to look at the Robocode API (<http://robocode.sourceforge.net/docs/robocode/>) to see all the other things your robot can do.

Above all, good luck, have fun, and enjoy!

## See also

### Robot API

- Robot API (<http://robocode.sourceforge.net/docs/robocode/>)

### Tutorials

- System Requirements for Robocode
- How to download and install Robocode
- The anatomy of a robot
- Getting started with Robocode
- Robocode Game Physics
- Scoring in Robocode
- Using the robot console
- Downloading other robots
- Learning from other robots
- Package your robot
- Frequently Asked Questions (FAQ)
- Articles about Robocode
- Starting Robocode from the command line
- Graphical debugging
- Using Eclipse as IDE
- Creating a project for your robots
- Creating a robot in Eclipse
- Running your robot from Eclipse
- Debugging your robot with Eclipse

### News and Releases

- RSS Feeds for the Robocode project ([http://sourceforge.net/export/rss2\\_project.php?group\\_id=37202](http://sourceforge.net/export/rss2_project.php?group_id=37202))
- Robocode file releases ([http://sourceforge.net/project/showfiles.php?group\\_id=37202&package\\_id=29609](http://sourceforge.net/project/showfiles.php?group_id=37202&package_id=29609))

## Home pages

- Classic homepage (<http://robocode.sourceforge.net/>)
- Robocode project at SourceForge (<http://sourceforge.net/projects/robocode>)
- Robocode Repository (<http://robocoderepository.com/>)
- Wikipediaentry for Robocode

Retrieved from "[http://robowiki.net/wiki/Robocode/My\\_First\\_Robot](http://robowiki.net/wiki/Robocode/My_First_Robot)"

Categories: Robocode Documentation | Tutorials

---

- This page was last modified on 15 October 2012, at 07:01.