Robocode – the final lesson

1 Before doing this tutorial

- Make sure you have installed Java and Robocode
- Make sure you have completed **Getting started with Robocode**
- Make sure you have completed **Robocode - evolving your first robot**

2 What is the plan for today?

Today we will create our Day3 robot, we also call it RevengeBot. It will enhance our Day2 robot in several ways. The goal is that our Day3 robot will beat our Day2 robot.

Let's start coding!

3 Start Robocode and enter the code editor

3.1 Start Robocode

- Click Start on your computer and type **`Robocode`**, then click on the Robocode app to start
- Now click on Robot > Source Editor to start the Java Editor

3.2 Create your Day3 Robot

- Click **`File > Open`**
- Open the **`HomeChallenge`** folder and then Open **`Day2.java`**
- In the editor change the line

```
public class Day2 extends JuniorRobot
```

to

```
public class Day3 extends JuniorRobot
```

- Save the file and select Yes to save to a new file.

4 Planning our new robot

Today we will code a robot that is designed to defeat the **`Day2`** robot. We begin by listing everything we know about our **`Day2`** robot's behaviour, and how we can defeat it.

Day2 Robot Behaviour	How we can defeat it
Day2 expects the target to stand still	**Always move** We will always be moving around and scanning constantly
Day2 starts moving as soon as its hit	**Big Bullets** We won't shoot as fast and as often, but we will use much bigger bullets when we do shoot
Day2 stands still when it shoots	**Stationary Target** • If we scan an enemy, we will shoot • If we get hit by a bullet, we will immediately shoot back in the direction the bullet came from
Day2 keeps on shooting when it scans an enemy	**Evasive Action** As soon we are hit and have shot back, we will move out of the way quickly so that Day2 wastes its energy

Believe it or not, but the list above is what computer programmers call a requirements document and is something that real programmers do before they write any code. Understand the problem, describe how you are going to solve it, and then do the coding.

5 Preparing the code

We will first delete a lot of code from the Day2 robot that we do not need to use:

Delete this import from the top of the file (no tricky maths for this robot):

```
~~import java.awt.geom.Point2D;~~
```

Delete the following 3 variables from the variable declaration section

```
~~private boolean bScannedEney = false;~~
private boolean bGoForward = true;
~~rivate int iEnemyX = -1;~~
~~private int iEnemyY = -1;~~
```

Clean up the **run()** method so it looks like this:

```
public void run() {
}
```

Clean out the **onScannedRobot()** method so it looks like this:

```
public void onScannedRobot() {
}
```

Clean out the **onHitByBullet()** method so it looks like this:

```
public void onHitByBullet() {
}
```

Keep the direction switch code in the **onHitWall()** method but delete the rest, it should look like this:

```
public void onHitWall() {
     bGoForward = !bGoForward;
}
```

Delete the entire **absoluteBearing()** method but remember to not delete the very last } in the file (the one that closes our Day3 class.

Save and Compile the code. You should not get any errors. You also have a robot that does absolutely nothing. Now we start implementing our requirements!

6 Implementing Day3 a.k.a. RevengeBot

We will code each requirement listed in the table above, one by one:

6.1 Requirement 1: Always Move

We do this in the run method like this:

```
public void run() {
	// Always Move
	turnGunRight(5);
	if (bGoForward) {
		ahead(10);
	}
	else {
		back(10);
	}
}
```

6.2 Requirement 2 & 3: Big Bullets & Stationary Target

Before we do this there are two things we have to cover: If you call the **fire()** method you can give it a value from 0.1 to 3 (in increments of 0.1). The higher the number the slower the bullet travels, the more energy it drains, the slower the rate of fire but the more damage it does. We will turn the dial right up to 3. Also, when you get hit by a bullet, there is a variable you can use called **hitByBulletAngle**. This variable tells you the angle (in degrees) from where the bullet that hit you came from. Now change the **onScannedRobot()** and **onHitByBullet()** methods to:

```
public void onScannedRobot() {
	// Big Bullets & Stationary Target
	fire(3);
}

public void onHitByBullet() {
	// Stationary Target
	turnGunTo(hitByBulletAngle);
	// Big Bullets
	fire(3);
}
```

6.3 Requirement 4: Evasive Action

Finally, we must code our evasive action. We will use two methods that allow us to quickly go forwards and turn or go backwards and turn. Add the following underlined lines of code to the onHitByBullet() method:

```
public void onHitByBullet() {
	// Stationary Target
	turnGunTo(hitByBulletAngle);
	// Big Bullets
	fire(3);
	// Evasive Action
	if (bGoForward) {
		turnAheadRight(50,45);
	}
	else {
		turnBackLeft(50,45);
	}
}
```

That's it! Save and compile your code. Once you have fixed any errors you can start a new battle with a **`Day2`** and a **`Day3`** robot. Unless you are very unlucky you will see that the

Day3 robot beats the **Day2** robot every time! The **Day3** robot will even put up a good fight against a Day1 robot.

We have reached the end of today's training. We will give you a couple of resources to look at, and next time we will give you the Ultimate Challenge.

7 More Resources

A reminder from yesterday, the documentation for the JuniorRobot class showing you all the variables and methods you can use:

https://robocode.sourceforge.io/docs/robocode/index.html?robocode/JuniorRobot.html

There are also **Robot** and **AdvancedRobot** classes that you can extend which give you access to a lot more variables and methods but are more complicated to work with:

https://robocode.sourceforge.io/docs/robocode/index.html?robocode/Robot.html

https://robocode.sourceforge.io/docs/robocode/index.html?robocode/AdvancedRobot.html

And finally, the RoboWiki, a huge resource for Robocode robot coders:

http://robowiki.net/wiki/Main_Page

Just in case the wiki is not reachable, try to find it on the web archives site

https://web.archive.org/web/20190828182533/http://www.robowiki.net/wiki/Main_Page

The authors have also put up a temporary copy of the Wiki on
http://robowiki.duckdns.org/index.php/Main_Page