

Introduction to Robocode

How to get Robocode:

If you don't already have it, you will need to install Java first:

http://www.java.com/en/download/download_the_latest.jsp

Next download Robocode:

http://sourceforge.net/project/showfiles.php?group_id=37202

How to run Robocode:

- Open the start menu
- Navigate to Programs--Robocode and click Robocode

Two windows should popup(see below), if it's your first time running Robocode, it may be slow, be patient. DO NOT close this MSDOS window, it needs to stay open the entire time you run Robocode.

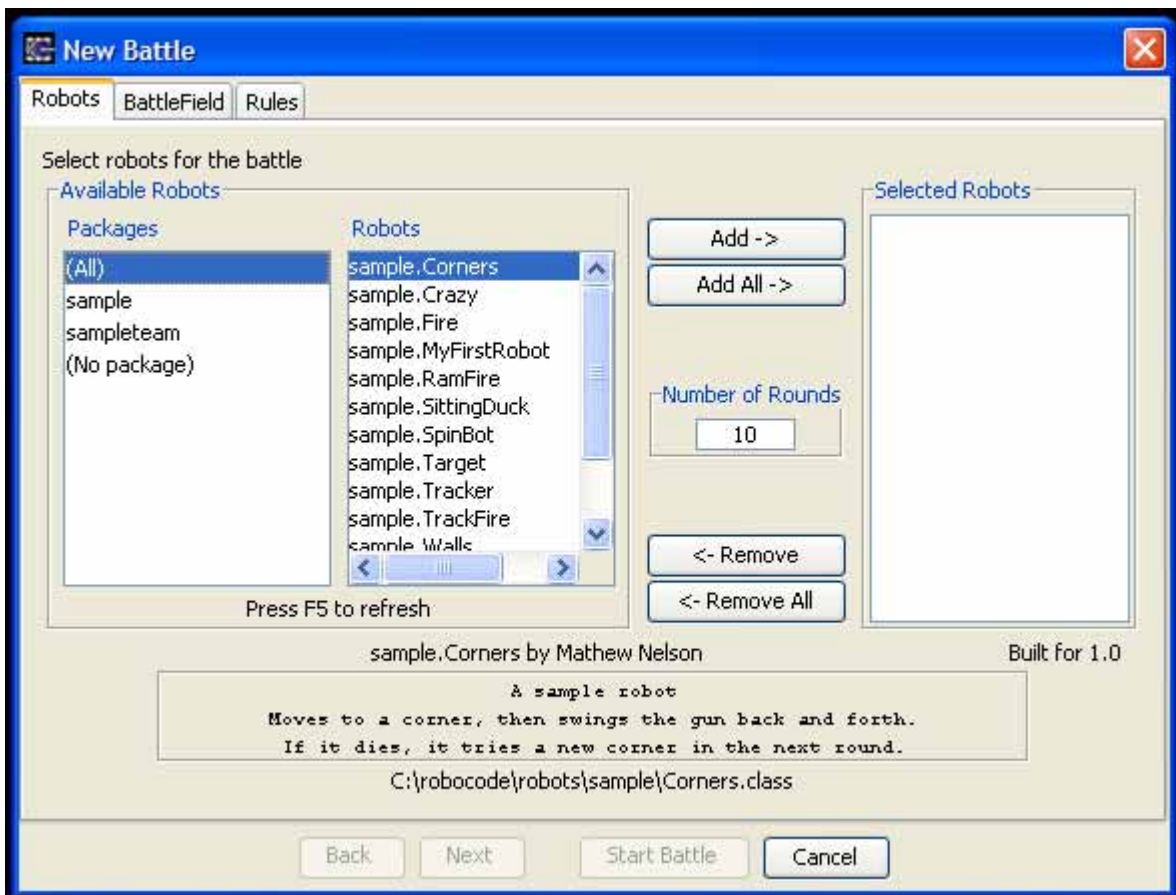


The image shows a screenshot of a Windows window titled "Robocode". The window contains a command prompt with the following text: `c:\robocode>java -Xmx256M -jar robocode.jar`. The window has a blue title bar and standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt area is black with white text.

This next window is just a loading window and will eventually go away.



To start a new robot battle, select Battle from the top menu then new.



Select a few robots from the list and press add and then start battle.

How to edit a robot:

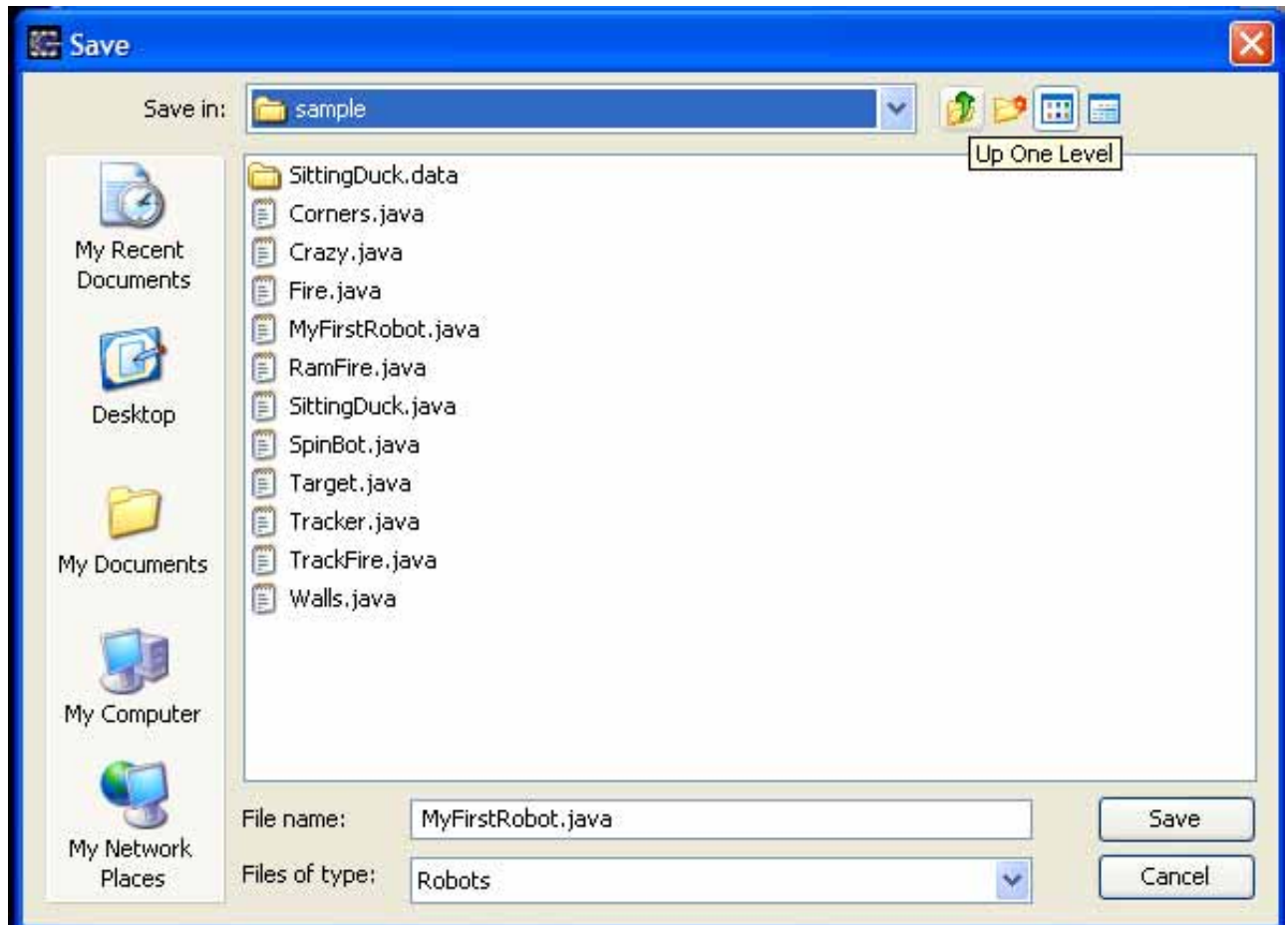
Select Robot from the top menu and then Robot Editor.

Now, from the Robot Editor menu, choose File--Open.

Look in the folder called Sample and open the file called *MyFirstRobot.java*

First, find the line at the top that says *package sample;* and replace "sample" with your name, no caps.

After you have edited your bot, press File--Save As and press the up one level button.

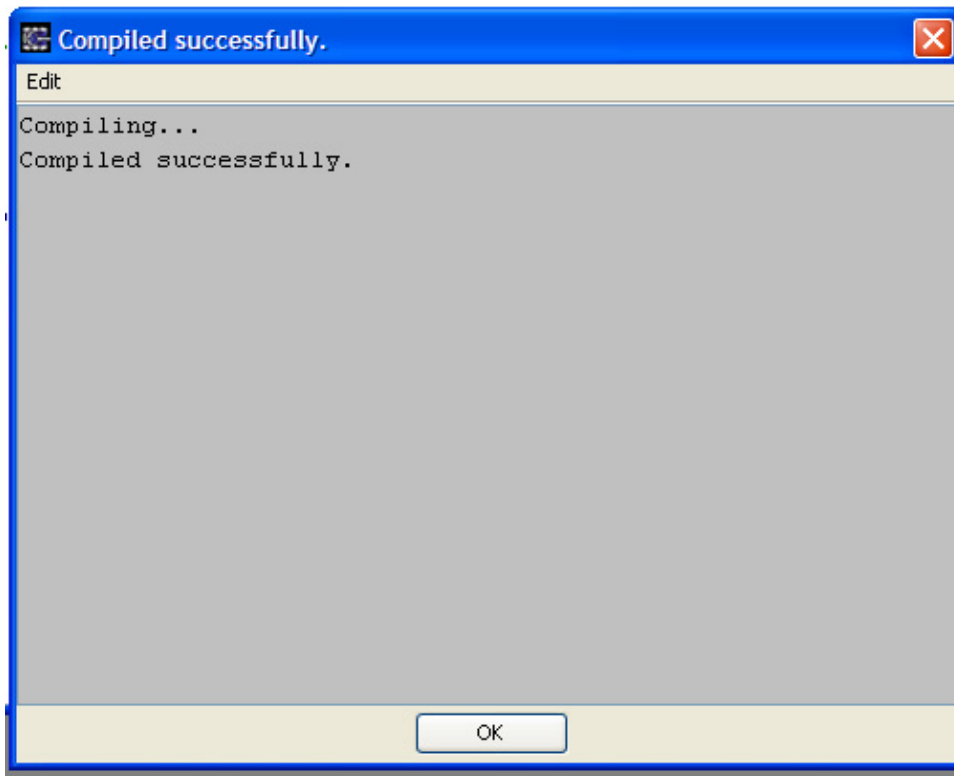


Save you work in an new directory of your network drive called "Robocode files"

After saving, select Compile from the Robot Editor menu, if you get the compiling

successful message(see below) then go ahead and have fun battling else { if (haha programming joke!) you received errors then go back and check your work until you get it right.

Once you are ready to test your robot, make a copy of the file located on your network drive and paste it into the "C:\Program Files\robocode\robots" directory



Coding Tutorial

event: *An action or occurrence detected by a program.*

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

events include:

- onHitByBullet(HitByBulletEvent e) - This method will be called when your robot is hit by a bullet.
- onHitRobot(HitRobotEvent e) - This method will be called when your robot collides with another robot.

- onHitWall(HitWallEvent e) - This method will be called when your robot collides with a wall.
- onScannedRobot(ScannedRobotEvent e) - This method will be called when your robot sees another robot.
- onWin(WinEvent e) - This method will be called if your robot wins a battle.

method: *A named section of a program that performs a specific task.*

```
public void run() {  
  
    while (true) {  
        ahead(100);           // Move ahead 100  
        turnGunRight(360);    // Spin gun around  
        back(100);           // Move back 100  
        turnGunRight(360);    // Spin gun around  
    }  
}
```

methods include:

- ahead(degrees) - Moves your robot forwards.
- back(degrees) - Moves your robot backwards.
- fire(1 to 3) - Makes your robot fire a bullet.
- turnLeft(degrees) - Makes your robot turn left.
- turnRight(degrees) - Makes your robot turn right.
- turnGunLeft(degrees) - Makes your robots gun rotate to the left.
- turnGunRight(degrees) - Makes your robots gun rotate to the right.
- turnRadarLeft(degrees) - Makes your robots radar rotate to the left.
- turnRadarRight(degrees) - Makes your robots radar rotate to the right.

parameter: *A value that is passed to a method or recieved from an event.*

```
public void onHitByBullet(HitByBulletEvent e) {  
    turnLeft(90 - e.getBearing());  
}
```

In this case, **e** is the parameter for the HitByBullet event.
getBearing() is a method that returns the bearing to the bullet.

My First Robot

```
package sample;
import robocode.*;
import java.awt.Color;

/**
 * MyFirstRobot - a sample robot by Mathew Nelson
 *
 * Moves in a seesaw motion, and spins the gun around at each end
 */
public class MyFirstRobot extends Robot
{
    /**
     * MyFirstRobot's run method - Seesaw
     */
    public void run()
    {
        setColors(Color.red, Color.blue, Color.green);
        while (true)
        {
            ahead (100);      // Move ahead 100
            turnGunRight(360); //Sping gun around
            back (100);      //Move back 100
            turnGunRight(360); //Spin gun around
        }
    }

    /**
     * Fire when we see a robot.
     */

    public void onSannedRobot(ScannedRobotEvent e){
        fire(1);
    }

    /**
     * We were hit! Turn perpendicular to the bullet,
     * so our seesaw might avoid a future shot.
     */

    public void onHitByBullet(HitByBulletEvent e)
    {
        turnLeft(90- e.getBearing());
    }
}
```

}

run is the main method for your robot. We override this to set your robot's basic behavior. What it will do is continually loop(unless an event occurs) the instructions to move ahead(100), turnGunRight(360), move back(100) and turnGunRight(360). Inside the events, we tell the robot what to do when that event occurs. For example, using the onScannedRobot event, we tell the robot to fire a bullet with an energy of one when the robot detects another robot (using its radar).