

Victoria University of Wellington
School of Engineering and Computer Science

SWEN221: Software Development

Lab Handout

The purpose of this lab is to explore some subtle aspects of encapsulation. In particular, a program is given which appears (for the most part) to be properly encapsulated. However, there remain certain ways in which code can break encapsulation. Your job is to harden the code base against such attacks.

Before the end of the lab, you should submit your solutions via the *online submission system* which will automatically mark it. You may submit as many times as you like in order to improve your mark and the final deadline will be Friday @ 23:59.

1 RobotWar

A small software program, called Robot War, is provided which implements a battle arena in which robot with *artificial intelligence* can compete against each other.

You can download the `robotwar.jar` from the lecture schedule on the course home page. You should begin by running the software and getting a feel for how it works. Figure 1 shows a screenshot from RobotWar.

2 What to do

You should find that a number of the *invalid* tests are failing. Your job is to harden the robot war code base against the attempts to break encapsulation embodied in the invalid tests. **Please note that just changing the test code is insufficient for this lab.** Rather, you instead need to ensure certain objects do not give away as much information as they currently are.

Submission

Your lab solution should be submitted electronically via the *online submission system*, linked from the course homepage. The required files are:

```
robotwar/core/Robot.java
robotwar/core/Battle.java
robotwar/core/Action.java
robotwar/util/Position.java
robotwar/actions/Shoot.java
robotwar/actions/Move.java
robotwar/robots/PatrolBot.java
```

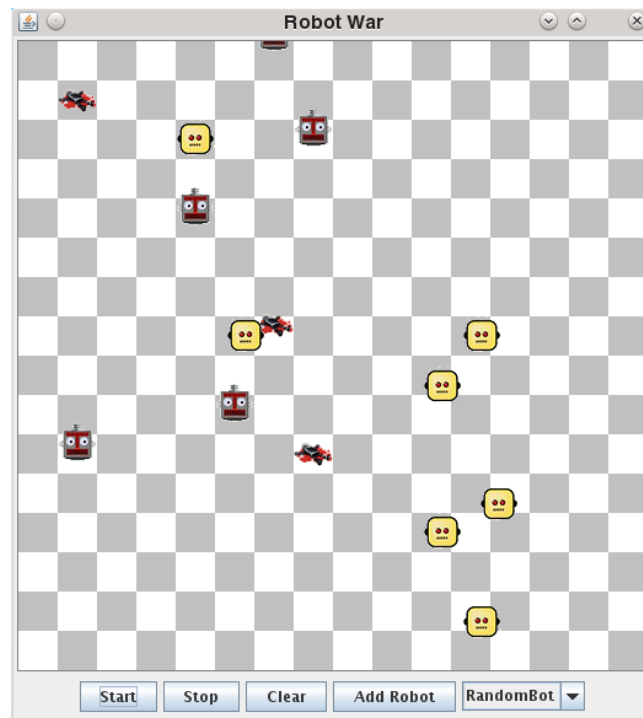


Figure 1: A screenshot from the robot war game.

You must ensure your submission meets the following requirements (which are needed for the automatic marking script):

1. **Your submission is packaged into a jar file, including the source code.** *Note, the jar file does not need to be executable.* See the following Eclipse tutorials for more on this:

<http://ecs.victoria.ac.nz/Support/TechNoteEclipseTutorials>

2. **The names of all classes, methods and packages remain unchanged.** That is, you may add new classes and/or new methods and you may modify the body of existing methods. However, you may not change the name of any existing class, method or package. *This is to ensure the automatic marking script can test your code.*
3. **All JUnit test files supplied for the assignment remain unchanged.** Specifically, you cannot alter the way in which your code is tested as the marking script relies on this. This does not prohibit you from adding new tests, as you can still create additional JUnit test files. *This is to ensure the automatic marking script can test your code.*
4. **You have removed any debugging code that produces output, or otherwise affects the computation.** *This ensures the output seen by the automatic marking script does not include spurious information.*

Note: Failure to meet these requirements could result in your submission being reject by the submission system and/or zero marks being awarded.