

COMP222 - 2017 - Second CA Assignment
Individual coursework
Robocode tank

Assessment Information

Assignment Number	2 (of 2)
Weighting	10%
Assignment Circulated	17 March 2017
Deadline	Wednesday April 26, 15:00
Submission Mode	Electronic
Learning outcome assessed	2. An appreciation of the fundamental concepts associated with game development: game physics, game artificial intelligence, content generation; 3. The ability to implement a simple game using an existing game engine
Purpose of assessment	To design and implement a tank bot for the Robocode tank battle game
Marking criteria	The marking scheme can be found in Section 3
Submission necessary in order to satisfy Module requirements?	No
Late Submission Penalty	Standard UoL Policy.

1 Robocode

Robocode is a programming game, where the goal is to develop a robot battle tank to battle against other tanks. The robot battles are running in real-time and on-screen. Robots can move, shoot at each other, scan for each other, and hit the walls (or other robots). More details can be found on the project web site, <http://robocode.sourceforge.net/>

2 Objectives

This assignment requires you to design and implement a tank bot for the Robocode tank battle game. You need to choose a game AI behaviour model (such as, for example, finite state machine, decision trees, behaviour trees, or any other mechanism of your choice) and implement your robot based on this behaviour model.

3 Marking scheme

I enforce a “no error policy” in this module: If your code does not compile, your overall assignment mark will be capped at 40%. Thus, you may get a higher mark for an incomplete solution than for an advanced sketch.

If you want to show me your attempt to add some features that does not compile TOGETHER with your working code, feel free to submit TWO jar files clearly indicating which one of them contains working code and which contains an incomplete one. In this case, you will not be penalised and you can get a higher mark.

You are required to submit Java code and an electronic document describing your design and implementation. The contribution of subtasks is as follows.

3.1 Documentation (40% of the mark)

You are required to submit a 700 to 1000 words document containing:

1. A short description of the behaviour control model of your choice (e.g., FSM, Decision trees, etc.). You only need to write a couple of paragraphs to show your understanding of how the model works. **10 marks**
2. A Robocode bot design description. In your design you should use the chosen behaviour control mechanism. For example, if you choose FSMs to represent bot's behaviour, give a graphical representation of states, transitions, and conditions under which the machine switches from one state to another. If you choose a tree-based model, give a graphical representation of the tree and clearly indicate tests and actions. Justify your design decisions, in particular, comment on why you believe these design decisions makes your bot more likely win the tournament. **20 marks**

3. A description of your implementation. Explain what classes and methods are used to implement the chosen behaviour model. You are not restricted in HOW you implement the bot (you can hard-code the behaviour in an ad-hoc manner, implement a general scheme, or use a third-party library) but your mark will depend on how closely you follow the design. You are allowed to deviate from the design; however, if your implementation does differ from the design, clearly identify and justify the modifications.

10 marks

3.2 Implementation (30% of the mark)

When you create a new robot in the editor use the following naming convention

Robot name: Please try to give your robot a unique name. That could be `FirstnameSecondname` (for example, I would use `BoriskKonev`) without spaces and special characters or a name that is unlikely to be chosen by others, e.g., `Crusher15041991`.

Please put your full name and student ID as a comment in the beginning of every Java file that you submit.

Package name: use `comp222`

If you use a different package name, your bot might be lost and not make it to the competition.

The implementation will be marked as follows:

- Providing response to battle events (`onScannedRobot`, `onHitByBullet`, `onHitWall`,...) **10 marks**
- Following the design **10 marks**
- Clarity and style of code **10 marks**

3.3 Battle competition(30% of the mark)

Submitted bots will take part in a tournament. In order to qualify to participate in the tournament, you should make a reasonable effort to modify the default (bot skeleton in the editor) behaviour. Additionally, **no robot with code taken from elsewhere (with or without acknowledging the source) will be allowed in the competition.**

In the tournament bots will be randomly divided in groups of about 10 bots each (depending on the number of submission); group winners will progress to battle in the next round.

The battlefield size is Robocode default.

In the end, bots will be ranked by the **Robocode Total Score**. Those in the upper third of the ranking will get extra 30%, those in the middle third will get extra 20%, and those in the lower third will get extra 10%.

If for some reason you do not wish your robot to take part in the tournament, please let me know in advance.

4 Deadlines and How to Submit

- Deadline for submitting the second assignment is Wednesday, 26 April 2017 at 3pm.
- Submission is via the departmental submission system accessible (from within the department) from
<http://intranet.csc.liv.ac.uk/teaching/modules/module.php?code=COMP222>.
- In order to submit your bot, export it as a Jar file by choosing Robot→“Package robot for upload” in the Robocode menu.