

CS255 Artificial Intelligence

Coursework: Robocode robot tank

Introduction

Your task in this coursework is to design and implement a robot tank for the Robocode environment. “Robocode is a programming game where the goal is to code a robot battle tank to compete against other robots in a battle arena. [...] The player is the programmer of the robot, who will have no direct influence on the game. Instead, the player must write the AI of the robot telling it how to behave and react on events occurring in the battle arena. Battles are running in real-time and on-screen.”¹

The task

You should use appropriate AI techniques to design the control of your robot tank, and then implement your design using Robocode. You may use any AI techniques, but should justify your choices and, if you combine techniques, should explain how they interact.

There are two deliverables for this task: (i) a report describing your design and (ii) your robot implementation. Your robot will be evaluated using a range of battle field sizes, a selection of competitors, and will be run in both one-to-one and melee (a set of robots picked at random battling against each other, with no teams) competitions.

You **must** use the Java implementation of Robocode, version 1.9.2.4, **as supplied in the zip on the module webpage**². Do not use any other version. Your code should only use the standard Robot API, i.e. you should extend the Robot class (and **not** the AdvancedRobot). You should only use standard Java as installed on joshua.dcs.warwick.ac.uk. Ensure that your code compiles on joshua before submission. Non-compiling code will receive zero marks for quality of implementation and performance of the robot. No external APIs are permitted, although the standard Java APIs (java.util etc.) are permitted — if you are unsure whether you can use an API then check with the module organiser.

The report and submission

Reports **must** use the LaTeX class available on the module webpage, including adhering to the structure defined by the headings, and be a maximum of 5 pages (including references). You must put your student ID in the author field. You should submit your robot (source files as detailed below) and your report online using Tabula. A printed copy of your report should also be submitted to the cabinet in DCS.

In your report you do not need to introduce the robocode problem or the setting in terms of environment and agent properties. However, you should introduce your approach and give an overview of related work. The report should describe the design of your strategy, implementation, testing methodology, results and evaluation.

¹<http://robocode.sourceforge.net/docs/ReadMe.html>

²<http://www2.warwick.ac.uk/fac/sci/dcs/teaching/material/cs255/coursework/>

Your source (.java) files should be submitted as a zip to Tabula. This zip should contain **only** your java files and supplementary resource files. The java files should be in the correct directory structure for their package name. The top-level package name should be of the form `uXXXXXXX` where `XXXXXXX` is replaced with your University ID Number. An example submission is available in the zip from the coursework webpage.

You must use the template supplied in the zip for your robot.

Hints

- There is extensive documentation on Robocode, including the API and example robots, online at <http://robocode.sourceforge.net>.
- In developing your robot you should consider the key choices that must be made. The RoboWiki³ gives example strategies for radar, movement, and targeting and you can use these as a starting point for your design (with appropriate referencing).
- When you are developing your robot you might want to test against other robots from previous competitions.
- You can use any AI techniques, but you might want to consider GAs, rule-based systems, and/or plan libraries.

Deadline

The deadline for submission of your report and robot is **noon 16 January 2018**.

Marking Scheme

30% Justification and description of design

20% Quality of the strategy design

20% Quality of implementation

20% Performance of robot

10% Quality of the report

You will be docked marks for failing to follow the submission instructions for the code or report.

Cheating/Plagiarism

The report and robot that you submit is an **individual** piece of work. All submissions will be put through plagiarism detection software which compares against a number of sources, including other submissions for CS255, submissions at other universities, web sources, conference papers, journals and books. Your robot will be checked against the example robots that exist, and you **must build your own robot**. While the use of literature sources and discussions of general concepts/ideas is a healthy academic activity, any work submitted for assessment must not include that of others. Similarly, although you may discuss your design and implementation with others, your coursework submission should be undertaken individually. You should take care to reference other work appropriately. Please see the student handbook for more information, or ask if you need guidance.

³<http://robowiki.net/>