COMP222 - 2018 - Second CA Assignment
Individual coursework
Robocode tank

Assessment Information

Assignment Number	2 (of 2)
Weighting	10%
Assignment Circulated	12 April 2020
Deadline	Friday May 15, 17:00
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 or to extend Assignment 1 with an AI opponent
Marking criteria	The marking scheme can be found in Secition 2
Submission necessary in order to satisfy Module requirements?	No
Late Submission Penalty	Standard UoL Policy.

1 Disclaimer

There are two options to complete this assignment:

1. Extend assignment 1 in order to add an artificially intelligent opponent.
2. Implement a tank in Robocode.

1.1 Objectives for extending Assignment 1

This assignment requires you to implement an artificially intelligent opponent in Assignment 1. 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.

The gameplay should not be too easy or too hard. The gameplay should be getting gradually harder as time progresses.

Hint Replace the upper wall with a paddle controlled by AI. The paddle can define its behaviour (stop, go left, go right, go fast, go slow) according to input read from data of the environment, e.g. the coordinates of the ball, the user's paddle, the obstacles, etc. The behaviour should abide by some model, as stated above. That can be for example following the ball coordinates with speed increasing every time an obstacle disappears. Increasing the difficulty of beating the AI can be based on elapsed time, decrease in obstacles, increasing number of wins, etc.

1.2 Objectives for 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/`

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.

2 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.

2.1 Documentation (40% of the mark)

You are required to submit a 700 to 1 000 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. An AI paddle or 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 you 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 deviated from the design; however, if your implementation does differ from the design, clearly identify and justify the modifications. **10 marks**

2.2 Implementation (30% of the mark)

Note for Robocode Submissions 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 KonstantinosTsakalidis) 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 gameplay: 1. The paddle responds to gameplay. OR 2. the bot responds to battle events (onScannedRobot, onHitByBullet, onHitWall,...) **10 marks**
- Following the design **10 marks**
- Clarity and style of code **10 marks**

2.3 Runtime evaluation(30% of the mark)

Assignment 1 Extension The behaviour of the AI opponent will be assessed by the level that it achieves the described behaviour, the gradual progression of the gameplay difficulty.

Robocode Battle Competition Submitted bots will take part in a tournament against 11 other standard bots. At least 10 rounds will be played in a battlefield of default size. In the end, your bot will be ranked by the **Robocode Total Score**. If it ends in the upper third of the ranking, it will get extra 30%; in the middle third, it will get extra 20%; and in the lower third, it will get extra 10%.

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**.

3 Deadlines and How to Submit

- Deadline for submitting the second assignment is Friday, 15 May 2020 at 5pm.
- Submission is via the departmental submission system accessible from `https://sam.csc.liv.ac.uk/COMP/`.

1. Submit the extended game as you did in Assignment 1.
2. In order to submit your bot, export it as a Jar file by choosing Robot→“Package robot for upload” in the Robocode menu.