

Spring 2022 Project

The Menace

Instructions

- Watch [YouTube video snippet](#) (end at 42:39);
- Implement “The Menace” by replacing matchboxes with values in a hash table (key will be the state of the game);
- Train the Menace by running games played against “human” strategy, which is based upon optimal strategy (see [Tic-tac-toe](#)).
 - You will need to choose values for:
 - *alpha* (the number of “beads” to in each “matchbox” at the start of the game—may be different for each move: first move, second move, etc.)
 - *beta* (the number of “beads” to add to the “matchbox” in the event of a win)
 - *gamma* (the number of “beads” to take to the “matchbox” in the event of a loss)
 - *delta* (the number of “beads” to add to the “matchbox” in the event of a draw)

Instructions (2)

- Human strategy:
 - Chooses optimal strategy with probability p^* . In the “zone,” chooses random move.
- Implement logging:
 - Log each training run with date/time, win/loss/draw, and p . If you vary alpha, beta, etc. then record these values also.
 - For your final match(es), log every move taken by the “Menace” and its opponent.
 - Choose [SLF4j](#) as your logging framework (if you’re used to something else, or not using Java, then choose whatever logging framework you like).

* A suitable value of p might be 0.9

Instructions (3)

- Unit tests
 - You must run your unit tests *before* you start training. Show the date/time of your most recent unit test run.
 - Tests must have good coverage (each method must be covered).

Team sizes

- Teams of 2 or fewer:
 - Implement everything mentioned so far.
- Teams of 3:
 - Implement everything for teams of 2;
 - Implement a GUI that shows the progress of a game of tic-tac-toe.
- Teams of 4:
 - Implement everything for teams of 3;
 - Implement a game server* that can accept a challenge from a Menace (either your own, or that of another team—special bonus for this last).

* Any mechanism which allows a Menace-Menace game is OK

Deadline and deliverables

- There is no opportunity to present your results, although I will accept offers to demonstrate any inter-menace games (not auto-matches).
- You must post your report—which outlines your method, your conclusions and also references your log files and unit test results—to your designated github repository.
- Your *clean** repository (including report and source code) which includes instructions for running the program (in README.md) must be frozen by:
 - Midnight April 27th

* Do not include *.class* files or other non-intellectual files.

Notes

- Implementations in Java should use Maven to define dependencies and build the application.
 - In such a case, you will not need to describe how to build it.
 - In this case, you will use the “Java” version of *.gitignore* in your repo.
- Other implementations will need to describe how to build and run the program.