

Pre QEC 2023 - Artificial Intelligence (10 points)

Scenario

The Potato Cloud Fest CEO says he wants to find a way to keep his kid busy when he brings him to work in the weeks leading up to the festival. He tells you that his child is a big fan of the board game **BATTLESHIP**. You're not very good with children and you don't like to play **BATTLESHIP**, but you love programming.

To do him a favor while demonstrating your programming skills, you decide to prepare an artificial intelligence to play against his child. It is very important to have an artificial intelligence capable of winning quickly otherwise the child will discover the deception and his father will have to find a new way to keep him busy!

Required Software

-Python 3.7+

Rules - Battleship For Dummies

It will have **5 ships of width 1 and variable length**. They will be arranged secretly on **a grid of size 10x10**. Each turn, you must state a square where you suspect the presence of some part of any ship. You will know immediately if it is a successful shot (*HIT*), a missed shot (*MISS*) or a fatal shot (*SUNK*).

You win if you sink all the opposing ships.

Please note that this is a single player version of Battleship. Your goal is to write an AI that minimizes the number of turns required to sink ships.

Instructions

Do not modify anything other than `Bot.py`

- You must implement the `do_turn` method of `Bot.py` ;
- The latter must return a pair (tuple) having:
 - First component: the letter of the desired column (A to J)
 - Second component: the index of the desired row ([0, 9])
- You can add attributes to the `Bot` class as you want

Evaluation

You will be scored on the average number of turns required to sink all ships.

If your bot takes more than 90 turns to sink all ships, the execution is stopped `***`

Delivery

You just need to put the `Bot.py` file back.

Execute

To run the program you can run the command below. You can also add an argument that indicates the number of games to play and calculate the average compared to

```
$python3 Main.py <number of game>
```

Dots

The team with the lowest average will directly have 10 points . The rest of the teams will have a score weighted to the best team.

The total out of 10 points is calculated as follows:

$$Total = \frac{NbrAverageLap_{\{best\ team\}}}{NbrAverageLap_{\{team\}}} * 10$$