Battleship AI: documentation

Vardhan Gupta

may-june 2017

Contents

1 Introduction

This project aims to make an AI which can play the Battleship game intelligently. the code for this is written in python and we aim to use the pygame library of python to implement the graphical part of the game. We also try to look into various algorithms to play the game and finish it efficiently.

2 the battleship game

Battleship is a board game played between two players. each player is supposed to place ships in two different oceans i.e 10x10 grids. the ships are of different sizes namely, aircraft carrier (5x1), battleship (4x1), destroyer (3x1), cruiser (3x1) and a patrol boat (2x1). they may be placed anywhere on the grids either horizontally or vertically. they may never overlap with each other but can be placed adjacent to each other.The players take turns alternatively. On each turn a player is supposed to fire (guess a point on opponent's grid) at the opponents ships which may result in either a 'miss' or a 'hit'. when all the points of a ship have been hit, it results in sinking of that ship. the first player to sink all of opponent's ships is declared the winner.

3 work done so far

so far a we have made a basic console based version of the game. it is a very basic version which uses quite simple algorithms.

the algorithm used here is called hunt/target with parity. in this algorithm, we first randomly shoot in a checker board like pattern till a hit is found. once a hit has been found we try to sink that ship on which the hit was found. if there are no hits lift then and everthing is eithir in 'sink' state or 'miss' state or 'unguessed' state then we go back to the random shooting.

the functions made so far are:

- placeship, places ships on the board.
- checksink, checks the sinking status of any ship after a hit.
- shootingdirection, shoots in a given direction in order to sink a ship.
- targetmode, finds the best possible direction to shoot in and recalls itself if a hit is left on the board after sink a ship

4 future plans

we aim to use a better algorithm to play the game as this algorithm is really slow and gives an average of about 55-60 moves in a game. we aim to bring this number down to at least 40-45 moves. we have seen that this may be done by calculating probability density over the entire board and hitting the position with maximum probability.

After that, we will learn about pygame and work on the graphics of the game. After which, if time permits, we will work on an intelligent placement of ships based on past matches with the same player, instead of the random placement we have right now.