

# Battleships - A Game Playing Agent

## I. PROBLEM DESCRIPTION

Battleships is a 2-player guessing game originally published by the Milton Bradley Company in 1943. Each player uses two square grids of size 10, one to place the own ships, one to record the shots fired at the opponent. A total of 10 ships of various sizes has to be placed by every player in such a way that no two ships occupy adjacent squares. After the initial placement phase, players alternate in shooting at enemy ships. The goal of the game is to sink all enemy ships before one's own ships are sunk. An example of a board is shown in Figure 1.

The methods of solving guessing games that include a search component can have widespread real-world applications. Many robot-robot or robot-human interactions can be represented as adversarial games with uncertainty. Gaining insight into the structure and solution processes with toy problems such as Battleships will help us apply these methods to other problems of a similar structure.

## II. RELATED WORK

[1] introduces the problem of playing the game Battleships as a decision problem and proves that it is NP-complete. This paper, however, provides no insight into how a game playing agent can be implemented. [2] uses an approach based on genetic algorithms that adapts to an opponent's playing style over multiple games. The drawback of this method is the fact that the program has to complete a learning phase before it can play efficiently. [3] presents an extensive software design document from the initial requirements specification to the implementation of the game. The artificial intelligence is rudimentary using simple random initial placements of the ships and random shots. [4] presents a Java-based implementation using a strictly rule-based agent. These rules are hard-coded nested if-else constructs. However, such hard-coded approaches tend to be inflexible and ill-suited to react to unforeseen conditions.

## III. IMPLEMENTATION

We intend to split the problem into two sections: the ship placement and the actual game playing. The best possible placement maximizes entropy and therefore favors random placement of the ships. We will compare random placement to various strategies in placing ships to verify this hypothesis. Unlike previous methods outlined in Section II, we will implement an approach that does not depend on random shots or previous knowledge. The game will be represented as a search problem with chance nodes and a heuristic function

Fig. 1. Example of a Battleships board.<sup>1</sup>

that guides the search towards squares that contain a ship with high probability. We will encode both exploration of the board and exploitation in our heuristic, by taking into account the distance to the goal state (i.e. missing hits till every enemy ship is sunk) and the positions and orientation of found and already sunk ships. Given a set time limit of  $n$  seconds, we intend to use Iterative Deepening A\* with alpha-beta pruning. As opposed to rule-based expert systems and machine learning approaches, our approach is expected to perform well without any previous training and knowledge about the opponent. Rules are implicitly encoded in the heuristic function guiding the search, thus enabling it to react to unforeseen situations. Depth-limited A\* search has already been successfully applied to similar games and we expect it to be a well-suited approach for Battleships as well.

## IV. EVALUATION

Our main points of comparison will be [2], [3], and [4]. The source code for all of these publications is available online and we intend to interface our agent with these implementations. We will provide experimental validation of our placement and playing approach by competing with these programs. Specifically we will compute the average win rate resulting from 100 games against each implementation. We expect our approach to exhibit superior performance over the shown rule-based, knowledge-based, and genetic approaches.

## REFERENCES

- [1] M. Sevenster, "Battleships as a decision problem," *ICGA Journal*, pp. 142–149, 2004.
- [2] J. Bridon, Z. Correll, C. Dubler, and Z. Gotsch, "An artificially intelligent battleship player utilizing adaptive firing and placement strategies," 2009. [Online]. Available: [www.cores2.com/files/FinalResearchPaper.pdf](http://www.cores2.com/files/FinalResearchPaper.pdf)
- [3] S. L. Andersen, "Battleship - design and implementation," 2008. [Online]. Available: [www.rudar.ruc.dk/bitstream/1800/2826/1/Report\\_formatted.pdf](http://www.rudar.ruc.dk/bitstream/1800/2826/1/Report_formatted.pdf)
- [4] A. Ahmed, D. Harton, K. Doran, M. Colmer, R. Brown, and S. Pickford. (2010) Artificial intelligence battleship game. [Online]. Available: <http://code.google.com/p/battleshipaiproject/>

<sup>1</sup>[http://en.wikipedia.org/wiki/File:Battleship\\_game\\_board.svg](http://en.wikipedia.org/wiki/File:Battleship_game_board.svg)