Strategies to play Battleship

By Kuncheng Feng

Background

Battleship is a 2 player guessing game, each player have a 10 by 10 gridded board to place their 5 ships of varying sizes, at each turn each player fires a salvo at one of the opponent's grid, the winner is the one that sinks all the opponent's ships.

| Ship Type | Length |
|------------|--------|
| Carrier | 5 |
| Battleship | 4 |
| Cruiser | 3 |
| Submarine | 3 |
| Destroyer | 2 |



Image from https://battleship.fandom.com/wiki/Battleship_(game)

Mission

My goal is to explore the effectiveness of some strategies when playing battleship (classic version), by effectiveness I specifically want to measure the win rates of each strategy against other strategies.

For this purpose I created a few game playing machines that follow different strategies, and have them play with each other to gather the results.

Set up

I modeled the game in Common Lisp programming language, that includes modeling the board, its grids, and the ships that will reside on top of the board. I also included a "display" method to visually confirm that it's working.

| | Comm | and Pr | rompt - | clisp | | | | | | | | 0000 | | | | × |
|------------------|------|--------|----------|--------|--------|---------|--------|----------|--------|----------|----|------|--|--|--|---|
| [5]> |) (d | ispl | ay b | oard |) | - | | 112 | | 4 | | | | | | |
| + | A | в + | ر + | U + | E + | ۲ ++ | G + | н + | ⊥ + | J +- | | | | | | |
| 0 | | I | | I | | l | l | | | | | | | | | |
| 1 | | | | | | | | | | | -+ | | | | | |
| 2 | | | | | | 2 | | | | ļ | | | | | | |
| 3 | | | 1 | | | 2 | | | | ļ | | | | | | |
| 4 | | | 1 | | | 2 | | 1 | | l | | | | | | |
| 5 | | 5 | | | | | | | | ļ | -+ | | | | | |
| 6 | | 5 | | | 4 | | | | | ļ | -+ | | | | | |
| 7 | | 5 | | | 4 | | | 1 | | ļ | | | | | | |
| 8 | | 5 | | | 4 | | | I | | | | | | | | |
| 9 | | 5 | | | 4 | | I | | | ļ | | | | | | |
| + NIL [6]> | , | | T | | | | | - | | T | | | | | | |

Set up part 2

I then created game playing functions that allows simulation of the game on computers, there is a function to manually test the game, and a function to let machines play against each other.

| 🚾 Mark Command Prompt - clisp | - | |
|---|---|--|
| 6 6 1 1 | | |
| 7 7 2 2 2 | | |
| | | |
| | | |
| Enemy fired at J, 1 Enter target location: e 1 It's a hit! | | |
| Your markers and your board: A B C D E F G H I J A B C D E F G H I J | | |
| 0 0 0 x 0 0 0 | | |
| 1 x 1 0 5 5 x 5 5 0 | | |
| | | |
| 3 3 4 4 4 4 1 | | |
| | | |
| 5 5 3 3 3 1 | | |
| | | |
| | | |
| | | |
| 9 | | |
| Enemy fired at A, 1 Enter target location: | | |

Available AIs: RANDOMPLAYER 2 - RANDOMPLAYERPLUS 3 - RANDOMPLAYERPLUSPLUS 4 - TIERLISTPLAYER 5 - TIERLISTPLAYERPLUS 6 - TIERLISTPLAYERPLUSPLUS Enter a corresponding number to choose AI 1: 1 Enter a corresponding number to choose AI 2: 1 Enter the number of iterations: 1 Simulating games ... Game statistics: Number of games played: 1 Average number of turns: 91 Player 1 victories: 1 Player 2 victories: 0 Draws: 0 Average hits achieved by player 1: 17 Average hits achieved by player 2: 15 NIL [8]>

Manually playing

Machines playing

Random Player

For baseline comparison, an player that plays randomly is created.







Random Player

Performance (from 1,000 games)

| | Random Player | |
|---------------|--|---|
| Random Player | RP wins: RP wins: Draws: Average turns: RP average Hits: RP average Hits: | 46.4 % 44.1 % 9.5 % 95.7 16.7 16.6 |

Random Player Plus

A strategy is that we should prioritize firing at locations that are next to confirmed hits, and this play follows that strategy.







Random Player Plus

Performance (from 1,000 games):

| | Random Player | | Random Player Plu | S |
|--------------------|----------------------|--------|----------------------|--------|
| Random Player Plus | Random player+ wins: | 98.5 % | Random player+ wins: | 47.3 % |
| | Random player wins: | 1.1 % | Random player+ wins: | 51.4 % |
| | Draws: | 0.4 % | Draws: | 1.3 % |
| | Average turns: | 68.8 | Average turns: | 61.4 |
| | Average R+Hits: | 17.0 | Average R+Hits: | 15.5 |
| | Average R Hits: | 12.8 | Average R+Hits: | 15.8 |

Random Player Plus Plus

We know that if a line of hits are achieved, then it's most likely hitting a ship at a right direction, and if that line stop hitting then it's likely that the ship is gone. This player implements this strategy on top of the previous one.



Random Player Plus Plus

Performance:

| | Random Player | Random Player + | Random Player ++ |
|------------------|---|---|---|
| Random Player ++ | R++: 89.9 % R: 8.2 % D: 1.9 % Turns: 68.9 R++Hits: 16.9 R Hits: 13.0 | R++: 33.3 % R+ : 65.2 % D: 1.8 % Turns: 66.9 R++Hits: 12.9 R+ Hits: 16.4 | R++:50.0 %R++:48.9 %D:1.1 %Turns:56.0R++Hits:14.5R++Hits:15.9 |

Surprisingly the strategy is actually an downgrade, it seems that the confirmation of ships sinking is needed, as very often the AI place ships in parallel that counters this tactic.

Tier List Player

The previous strategy is still semi followed, but it will still all around the confirmed hits locations before moving on, essentially a Random Player Puls.

The true improvementing strategy of this player is that when an location misses, it will try to avoid the adjacent locations. It is called "Tier List Player" because it classified locations into tiers of exploration order









Tier List Player

Performance:

| | Random | Random + | Random ++ | Tier |
|------|--------------|---------------|----------------|--------------|
| Tier | T: 97.8 % | T: 56.3 % | T : 71.5 % | T: 48.4 % |
| | R: 1.7 % | R+: 41.2 % | R++: 26.9 % | T: 49.2 % |
| | D: 0.5 % | D: 2.5 % | D : 1.6 % | D: 2.4 % |
| | Turns: 71.8 | Turns: 59.4 | Turns: 56.8 | Turns: 50.3 |
| | T Hits: 17.0 | T Hits: 16.2 | T Hits: 15.7 | T Hits: 14.5 |
| | R Hits: 8.99 | R+ Hits: 14.2 | R++ Hits: 16.3 | T Hits: 16.6 |

Tier List Player Plus

When a ship is sunk, a player is suppose to announce it. In this situation a reasonable human player will try to infer where the ship had sunk, and avoid the areas next to the sunken ship.





Tier List Player Plus

Performance:

| | Random | Random + | Random ++ | Tier | Tier + |
|--------|--------------|--------------|--------------|--------------|--------------|
| Tier + | T+: 98.7 % | T+: 66.8 % | T+: 77.9 % | T+: 62.1 % | T+: 49.9 % |
| | R: 1.0 % | R+: 31.7 % | R++: 20.6 % | T: 36.5 % | T+: 48.2 % |
| | D: 0.3 % | D: 1.5 % | D: 1.5 % | D: 1.4 % | D: 1.9 % |
| | Turns: 60.6 | Turns: 45.3 | Turns: 52.0 | Turns: 47.8 | Turns: 51.9 |
| | T+ Hits:16.9 | T+ Hits:15.9 | T+ Hits:15.9 | T+ Hits:16.1 | T+ Hits:15.7 |
| | R Hits: 10.7 | R+ Hits:14.1 | R++Hits:16.1 | T Hits: 13.9 | T+ Hits:13.7 |

Tier List Player Plus Plus

This player follows the exact strategies as the previous one, except it avoid placing ships next to each other





Tier List Player Plus Plus

Performance:

| | Random | Random + | Random ++ | Tier | Tier + | Tier ++ |
|---------|--------------|--------------|--------------|--------------|--------------|--------------|
| Tier ++ | T++: 99.0 % | T++: 83.3 % | T++: 72.2 % | T++: 69.0 % | T++: 41.2 % | T++: 48.7 % |
| | R: 0.9 % | R+ : 15.5 % | R++: 25.3 % | T: 29.4 % | T+: 56.3 % | T++: 48.7 % |
| | D: 0.1 % | D: 1.2 % | D: 2.5 % | D: 1.6 % | D: 2.5 % | D: 2.6 % |
| | Turns: 42.8 | Turns: 60.2 | Turns: 61.8 | Turns: 49.2 | Turns: 46.6 | Turns: 46.6 |
| | T++Hits:17.0 | T++Hits:16.4 | T++Hits:16.5 | T++Hits:16.6 | T++Hits:16.9 | T++Hits:16.8 |
| | R Hits: 9.12 | R+ Hits:11.9 | R++Hits:15.0 | T Hits: 14.4 | T+ Hits:14.1 | T++Hits:14.7 |

What I wish to do more

If the circumstances permits, I would like to introduce more search pattern strategies, as most of the strategies implemented are revolved around what to do after achieving an hit.

Also, I would love to reformat the code for more scalability and performance, then I can simulate even more games for more accurate results.

Conclusion

Even though Battleship (classic) is a guessing game, just a tiny bit of bit of strategies can bring a huge improvement in chances of winning. And some strategies that we think as beneficial (not placing ships next to each other) is easily countered as it follows human logic.