CIS4930 Final Project Report

Battleship

By Ryan Arredondo, Shawn Joseph, Fernando Sierra

Description of the system

Our project is based on the “Battleship” game, which is a two-player guessing game played on ruled grids on which the players' ships are marked. The locations of the ships are concealed from the other player and each take turns calling “shots” at the other player's ships. The objective of the game is to destroy the opposing player's fleet.

This project was built using Python, and offers the user three difficulty modes to choose from (easy, medium, hard) which affect the difficulty of the A.I to play against. Easy mode has the A.I choose all random positions regardless of hits or misses. Medium mode is random until it hits a cell containing a ship, and then it switches to a target mode that adds the four surrounding grid squares to a stack of 'potential' targets (or less than four if the cell was on an edge/corner). Hard mode uses even parity to only shoot at specific points (every other cell, since the minimum size of a ship is two cells) until it gets a hit and then starts hunting for that target using the same method as in Medium mode.

Performance Evaluation of System

Function Name	Performance (O(n))
print_board()	O(n^2), where *n* is the number of rows or columns
to_be_placed()	O(1)
target(Player)	O(1)
get_coordinates(Player)	O(1)
location_check()	O(*n*), where *n* is the number of rows or columns
place_ship()	O(*nm*), where *m* is the number of ships
shoot()	O(*m*), where *m* is the number of ships
game_over()	O(n^2), where *n* is the number of rows or columns
target(Computer)	O(1)
get_coordinates(Computer)	O(1)
reset()	O(n^2), where *n* is the number of rows or columns
main()	O(n^2), where *n* is the number of rows or columns

Summary of experimental results

To gather large amounts of data, the game was set up so that two A.I players compete against each other in all three modes in test sets of 1, 10, 100, 1,000, 10,000, 100,000, and 1,000,000 games.

For each mode, the results for the average amount of turns, time of execution, and shortest amount of turns to win a game were gathered for each set:

EASY (All Random):

```
--- Number of games: 1 ---
--- Took 0.005476951599121094 seconds to execute ---
--- avg amt of turns: 95.0 ---
--- shortest game: 95 ---

--- Number of games: 10 ---
--- Took 0.043111324310302734 seconds to execute ---
--- avg amt of turns: 94.0 ---
--- shortest game: 81 ---

--- Number of games: 100 ---
--- Took 0.44407081604003906 seconds to execute ---
--- avg amt of turns: 93.15 ---
--- shortest game: 71 ---

--- Number of games: 1000 ---
--- Took 4.3430869579315186 seconds to execute ---
--- avg amt of turns: 92.99 ---
--- shortest game: 61 ---

--- Number of games: 10000 ---
--- Took 40.782471895217896 seconds to execute ---
--- avg amt of turns: 92.8959 ---
--- shortest game: 58 ---

--- Number of games: 100000 ---
--- Took 411.1259169578552 seconds to execute ---
--- avg amt of turns: 92.91451 ---
--- shortest game: 51 ---


--- Number of games: 1000000 ---
--- Took 4392.390584945679 seconds to execute ---
--- avg amt of turns: 92.934473 ---
--- shortest game: 47 ---
```

MEDIUM (Targeting):

```
--- Number of games: 1 ---
--- Took 0.002665996551513672 seconds to execute ---
--- avg amt of turns: 44.0 ---
--- shortest game: 44 ---

--- Number of games: 10 ---
--- Took 0.017752885818481445 seconds to execute ---
--- avg amt of turns: 57.7 ---
--- shortest game: 36 ---

--- Number of games: 100 ---
--- Took 0.14551806449890137 seconds to execute ---
--- avg amt of turns: 58.16 ---
--- shortest game: 34 ---

--- Number of games: 1000 ---
--- Took 1.5311942100524902 seconds to execute ---
--- avg amt of turns: 61.74 ---
--- shortest game: 32 ---

--- Number of games: 10000 ---
--- Took 15.28752088546753 seconds to execute ---
--- avg amt of turns: 61.2209 ---
--- shortest game: 26 ---

--- Number of games: 100000 ---
--- Took 147.60599493980408 seconds to execute ---
--- avg amt of turns: 61.37012 ---
--- shortest game: 22 ---

--- Number of games: 1000000 ---
--- Took 1478.5944907665253 seconds to execute ---
--- avg amt of turns: 61.345584 ---
--- shortest game: 22 ---
```

HARD (Targeting & Even Parity):

```
--- Number of games: 1 ---
--- Took 0.0034160614013671875 seconds to execute ---
--- avg amt of turns: 54.0 ---
--- shortest game: 54 ---

--- Number of games: 10 ---
--- Took 0.018568754196166992 seconds to execute ---
--- avg amt of turns: 56.4 ---
--- shortest game: 47 ---

--- Number of games: 100 ---
--- Took 0.16352391242980957 seconds to execute ---
--- avg amt of turns: 55.43 ---
--- shortest game: 29 ---

--- Number of games: 1000 ---
--- Took 1.6509618759155273 seconds to execute ---
--- avg amt of turns: 55.8 ---
--- shortest game: 32 ---

--- Number of games: 10000 ---
--- Took 17.70227289199829 seconds to execute ---
--- avg amt of turns: 56.3467 ---
--- shortest game: 27 ---

--- Number of games: 100000 ---
--- Took 168.72459506988525 seconds to execute ---
--- avg amt of turns: 56.29943 ---
--- shortest game: 22 ---

--- Number of games: 1000000 ---
--- Took 1754.1763780117035 seconds to execute ---
--- avg amt of turns: 56.286124 ---
--- shortest game: 20 ---
```

Main Conclusions

With this data, the conclusion can be made that when using both targeting and even parity, the average amount of turns the A.I takes to win a game is fewer than using just random choosing or just targeting in almost every set. Using targeting by itself improved the average number of turns to win a game from ~92 to ~61. When even parity is added to this, the average is lowered from ~61 to ~56. If this project was to be done again, another A.I mode, even more efficient than the even parity with targeting would be to use a probability density function. The new algorithm would calculate the most probable location to fire at next based on a superposition of all possible locations the enemy ships could be in. As more shots are fired, some locations become less likely, and others become impossible. The result will be a superposition of probabilities for the A.I to utilize in finding the ships.