Battleship Playing Program Utilizing Probability Density Functions

Authors: Jacob Boyd, Department of Physics, Kent State University, Kent, OH 44242
Jonathan Boyd, Department of Physics, Kent State University, Kent, OH 44242
Mentor: Dr. Darwin L. Boyd, College of Applied Engineering, Sustainability, and Technology, Kent State University, Kent, OH 44242

Introduction

Battleship is a classic Navy strategy guessing game in which players guess at the location of an opponent's ships. As the game progresses the possibility of a ship to be in any location changes and the probability that a ship occupies any cell can be calculated based on the current state of the board. By evaluating the probability, an optimal move can be made such that it selects the most likely position of any ship thus reducing the number of turns required to win.

Research Problem

To create a program capable of playing Battleship against another player utilizing probability density functions to optimize performance.

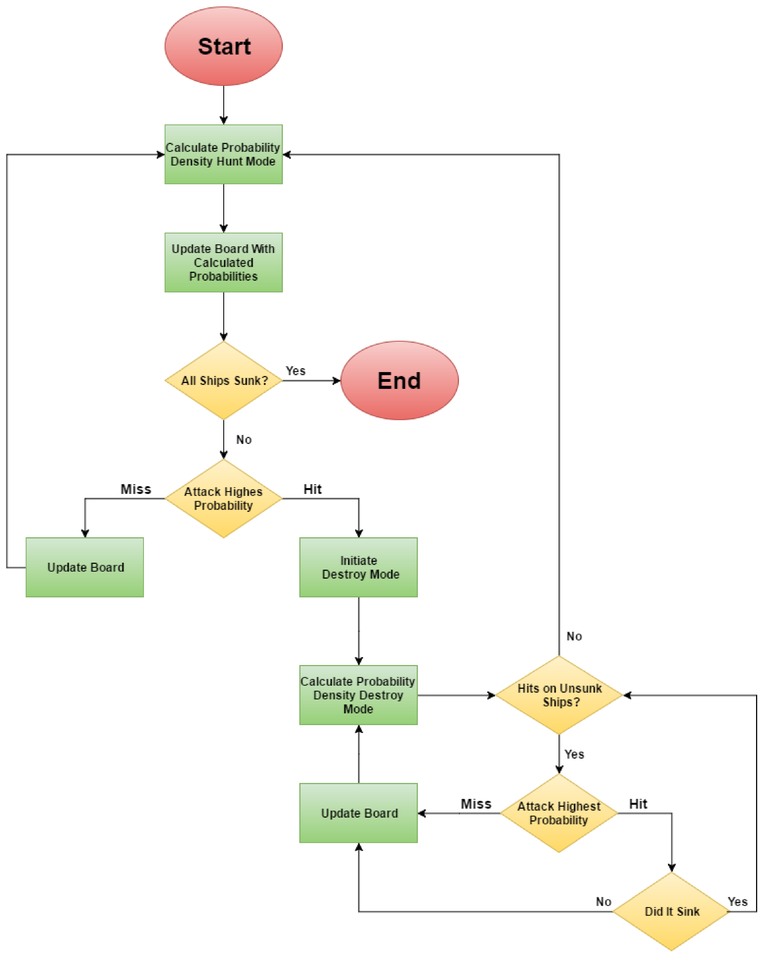


Figure 1: Flow Chart explaining the basic logic of the program

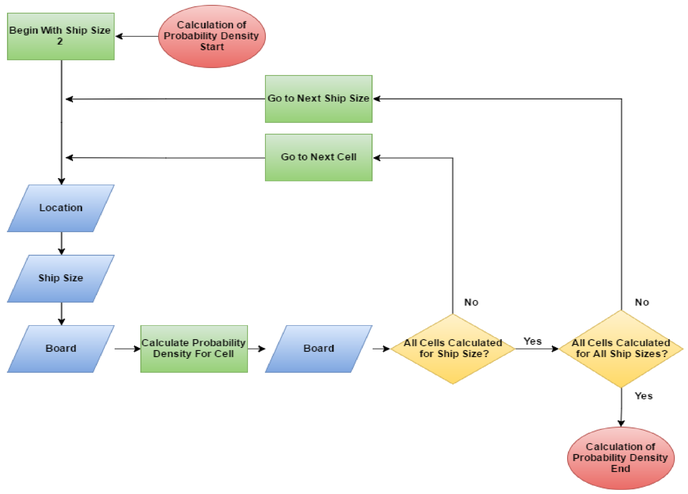
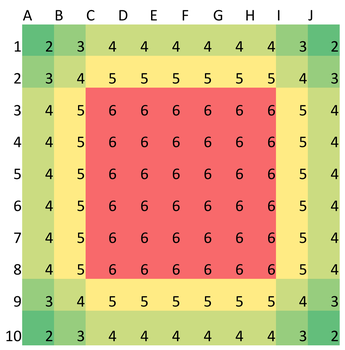
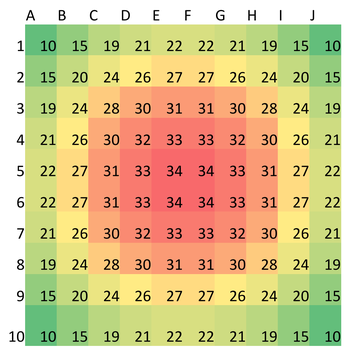


Figure 2: Flow Chart explaining the logic of the probability calculations



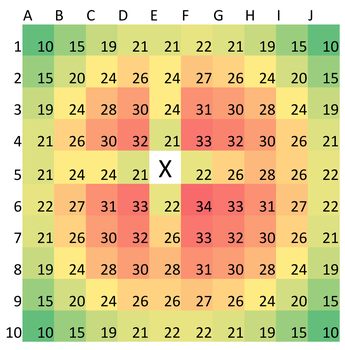
	A	B	C	D	E	F	G	H	I	J
1	2	3	4	4	4	4	4	4	3	2
2	3	4	5	5	5	5	5	5	4	3
3	4	5	6	6	6	6	6	6	5	4
4	4	5	6	6	6	6	6	6	5	4
5	4	5	6	6	6	6	6	6	5	4
6	4	5	6	6	6	6	6	6	5	4
7	4	5	6	6	6	6	6	6	5	4
8	4	5	6	6	6	6	6	6	5	4
9	3	4	5	5	5	5	5	5	4	3
10	2	3	4	4	4	4	4	4	3	2

Figure 3: The probability density for a single size 3 ship



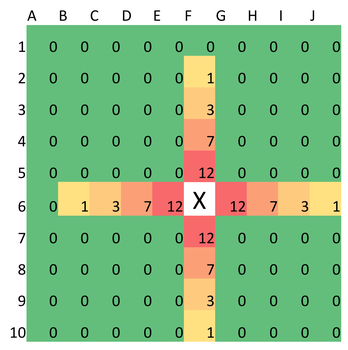
	A	B	C	D	E	F	G	H	I	J
1	10	15	19	21	22	22	21	19	15	10
2	15	20	24	26	27	27	26	24	20	15
3	19	24	28	30	31	31	30	28	24	19
4	21	26	30	32	33	33	32	30	26	21
5	22	27	31	33	34	34	33	31	27	22
6	22	27	31	33	34	34	33	31	27	22
7	21	26	30	32	33	33	32	30	26	21
8	19	24	28	30	31	31	30	28	24	19
9	15	20	24	26	27	27	26	24	20	15
10	10	15	19	21	22	22	21	19	15	10

Figure 4: The sum of the probabilities for all ships



	A	B	C	D	E	F	G	H	I	J
1	10	15	19	21	21	22	21	19	15	10
2	15	20	24	26	24	27	26	24	20	15
3	19	24	28	30	24	31	30	28	24	19
4	21	26	30	32	21	33	32	30	26	21
5	21	24	24	21	X	22	26	28	26	22
6	22	27	31	33	22	34	33	31	27	22
7	21	26	30	32	26	33	32	30	26	21
8	19	24	28	30	28	31	30	28	24	19
9	15	20	24	26	26	27	26	24	20	15
10	10	15	19	21	22	22	21	19	15	10

Figure 5: A miss at E5 with Hunt Mode Probability



	A	B	C	D	E	F	G	H	I	J
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	0	0
3	0	0	0	0	0	3	0	0	0	0
4	0	0	0	0	0	7	0	0	0	0
5	0	0	0	0	0	12	0	0	0	0
6	0	1	3	7	12	X	12	7	3	1
7	0	0	0	0	0	12	0	0	0	0
8	0	0	0	0	0	7	0	0	0	0
9	0	0	0	0	0	3	0	0	0	0
10	0	0	0	0	0	1	0	0	0	0

Figure 6: A hit at F6 with Destroy Mode Probability

Method

The program calculates discrete values for the probability of a ship being located at a given cell. These are not true probability density functions which evaluate the integral of a function. The calculated values used in the program would be better classified as a probability density matrix as they are discrete values.

In Hunt Mode the probability value given to a certain cell is determined by the number of ways a single ship can be oriented to fit on a that cell. The values for each remaining ship size are then summed to produce a final value. Destroy Mode is activated when a ship is hit in Hunt Mode. In Destroy Mode the probability value given to a certain cell is determined by the number of ways a single ship can be oriented to fit on that cell and the hit simultaneously. Again, the values for each remaining ship size are summed to produce a final value.

The program was written in C++ utilizing an object oriented design approach. The state of the board was stored in a 2-dimentional array of a programmer defined class. The probability variable of each cell was calculated based on the state of the board and the cell with the highest value was selected to attack. If two cells held the same value one was selected at random. If a ship was hit, the destroy mode was initiated in which the ship that was hit was searched for in adjacent cells based on a different probability value given. Once a ship was sunk, the ship resumed a hunting mode with the probability value of sunken ships left out of the calculation. The program proceeds in this manner until all five opponent ships are sunk at which point the program terminates. The program was tested against human players to determine its effectiveness.

Conclusions

The program with the implementation of probability density functions showed a marked reduction in the number of moves required to win a game from random guessing. In tests against human players it was capable of winning. Improvements could be made with the implementation of predictive algorithms to more accurately identify opponent ship placement.

References

Berry, N. (2011, December 3). Battleship. Retrieved March 14, 2016, from http://www.datagenetics.com/blog/december32011/

Bridon, J. G., Correll, Z. A., Dubler, C. R., & Gotsch, Z. K. *An Artificially Intelligent Battleship Player Utilizing Adaptive Firing and Placement Strategies* (Doctoral dissertation, Pennsylvania State University). PA: Pennsylvania State University.