

Battleship

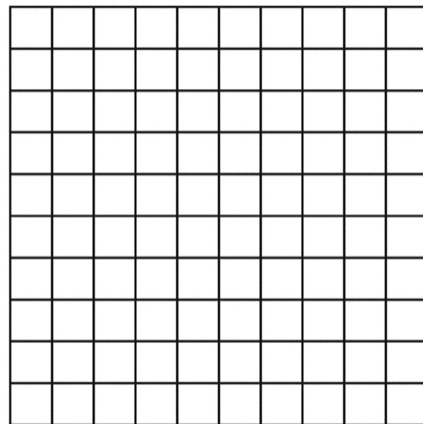
A faint, light blue line drawing of a battleship is visible in the background, spanning the width of the slide. The ship is shown from a side profile, facing right, with its complex superstructure, masts, and gun turrets clearly outlined. The drawing is semi-transparent, allowing the dark blue background to show through.

Jasper Wilkerson
Jade Sherer

Things to Consider:

Rules of Battleship:

- Players first place 5 ships, lengths: 5, 4, 3, 3, 2
- Ships may be next to one another, but cannot overlap
- Players then take turns firing missiles in order to sink their opponent's ships
- In our version, players do not take another turn on hit



Limits of AI

Battleship is not without strategy. Consider a 3 wide ship in the two environments below



At the beginning of the game, there are **30,093,975,536** possible ship combinations, which would be unreasonable to calculate

However we *can* simplify and use some strategy to more effectively guess

Versions

- Four versions were built as a series of improvements on the previous.
- The final version is entirely original.
- All versions are reflex agents.

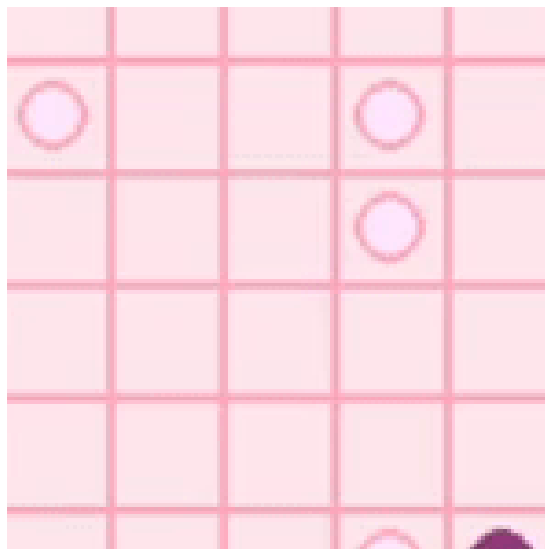
Random

- The simplest reflex agent.
- Served to test the framework and serve as a baseline.
- Easily outclassed.



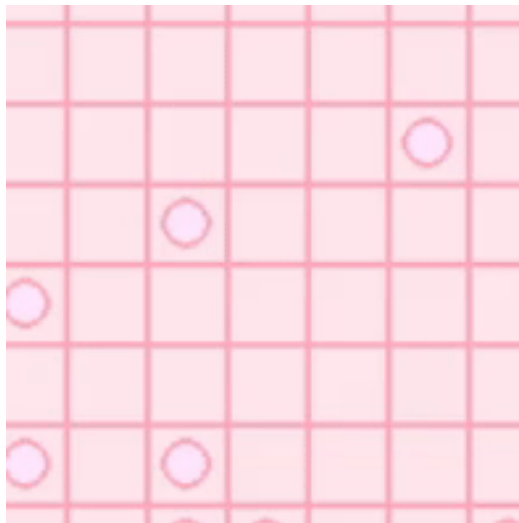
Random Hunt

- Adds a subroutine: "hunt".
- "Hunt" restricts the choices when there is a hit.
- Begins to mimic sub-optimal human play.



Random Smart Hunt

- Allows the “hunt” sub-routine to account for orientation.
- Can make incorrect assumptions with ship clusters.
- Similar to a slightly below average human player.
- Still behaves like Random when there are no hits.



Parity

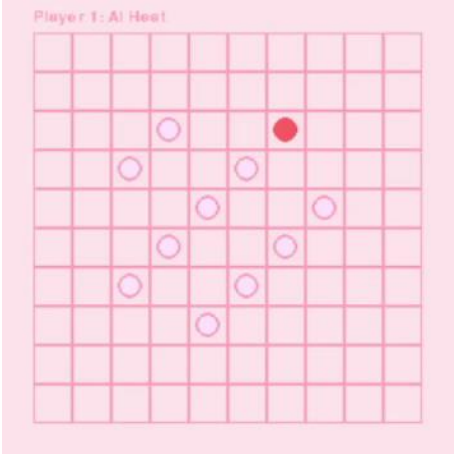
- Utilizes the fact that the smallest ship is 2-wide.
- Fires on a checkered subset of the board randomly.
- Still implements “smart hunt” when a ship is hit.



Heatmap AI

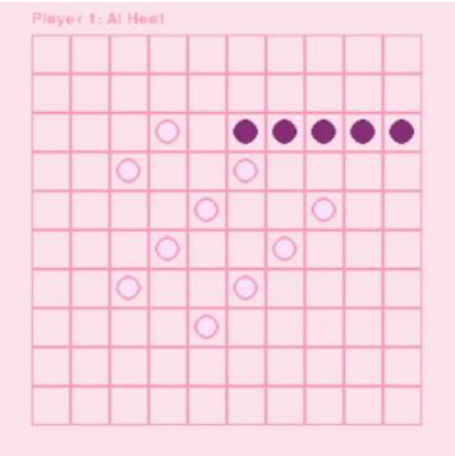
- Replaces the random movement model of the previous AI with a probabilistic one.
- Generates a heatmap off all currently possible ship placements.
- Still implements the “smart hunt” subroutine to save on processing time.
- Has a predetermined opening series of moves as the possibilities of a blank board are too large.
- Explanation on next slide.

Heatmap AI



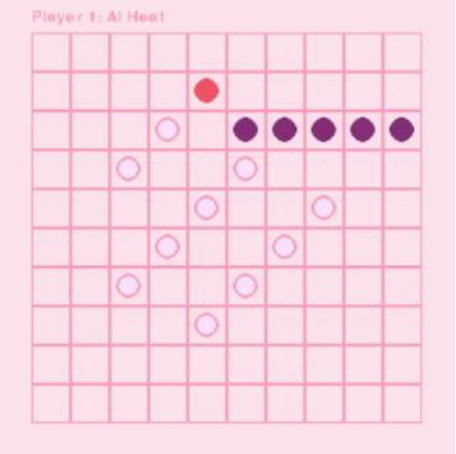
Opening firing pattern

Generate Heatmap



Hunt First Ship

3217352	4658774	5454434	5196920	6373954	5233444	5374582	4951720	3659964	2018982
4659690	5926586	5472974	4993932	7259378	5136140	5323444	4936090	3662390	2019652
5444300	5461174	2028498	0	2591452	0	0	0	0	0
5165092	4951976	0	596238	1932740	0	2071120	2900824	4443658	3363880
6294990	7141394	2863366	1923594	0	622408	608114	0	3360890	3491696
5813150	5881596	1284178	0	622408	622408	0	2493860	5473498	5448652
5155560	4909500	0	1875162	609682	0	2954138	5226200	6881138	6285952
5859496	6573424	3467458	4016360	0	2467692	5231698	8950444	6678176	5096786
4603688	5801014	5346232	7046426	4885790	5768220	6801686	6660526	5586708	4523678
3194286	4593608	5374682	6294332	5086402	5757346	6195974	5676494	4526198	3176332

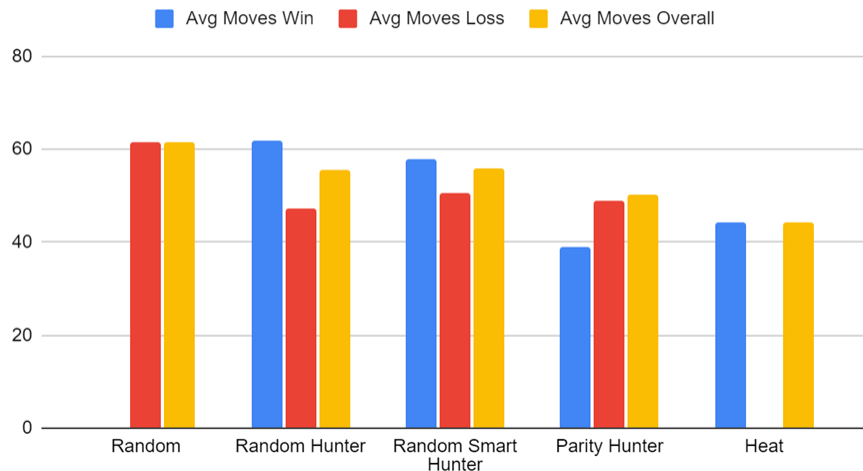


Execute Heatmap

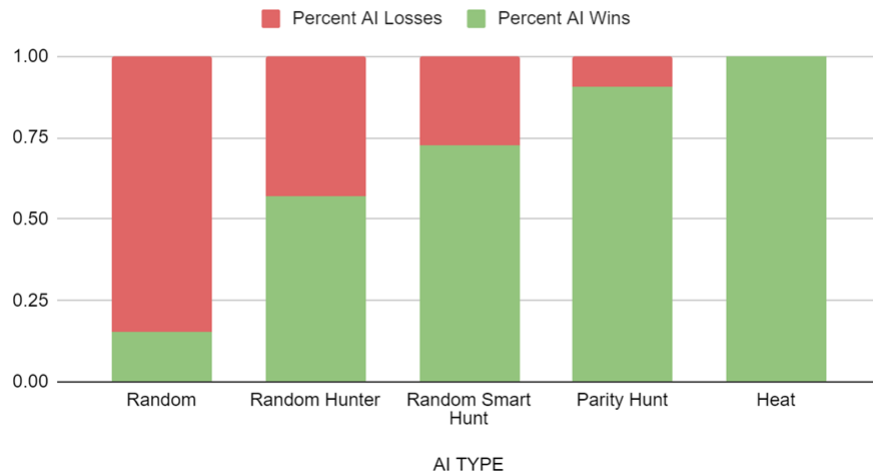
Overall Results

- Overall, Heat performed best and lost no games out of 38 it played
- It made the fewest average moves, while predictably random ai's made a lot of moves

Average Moves



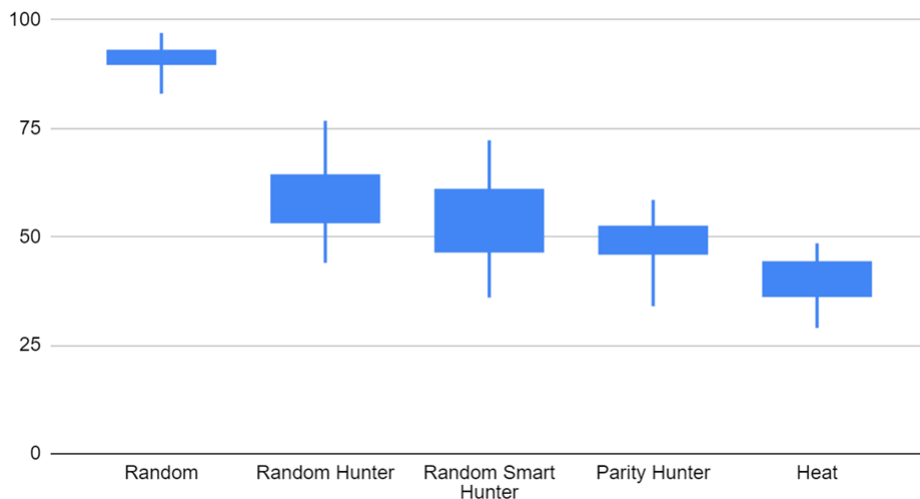
Wins and Losses



Performance vs Random AI

- Heat performed the best against random ai, average of **44 Moves**
- Parity averaged **52 Moves**
- Smart Hunter averaged **60 Moves**
- Random Hunter averaged **64 Moves**
- Random vs itself took **92 Moves** to win

Performance VS Random AI



Questions?