AI Assignment 1: Xtreme Tic Tac Toe
- How the Bot works
  - Heuristics
- Results and Analysis
  - Case study: Local Check Heuristic Failure
- Improvements that can be done

AI Assignment 1: Xtreme Tic Tac Toe

How the Bot works

In the initial report submitted, we planned to use Monte-Carlo Search on Tree. However, due to constraints added by the submission format, we used a simple Minimax with Alpha-Beta pruning, and custom made heuristics. Optimisations were also made to exclude moves that might lead to local victories for the opponent that might not be considered by the search itself. Possible extensions we considered were to use a lookup table generated by MCTS, but we decided against it in favour of improving the bot with better heuristics.

Heuristics

We tested the bot with three heuristics, ranging from absolutely trivial to rather complex. Afterwards, we added some additional considerations on top of the third heuristic to form an improved **fourth**, which is the one we used eventually.

	HEURISTIC 1	**HEURISTIC 2**	**HEURISTIC 3**	**HEURISTIC 4**
Definition	Play the first move available to the bot	If smallboard won in future move, then +2, if opponent wins then -1.	Same as Heuristic 2, but with weighted increase or decrease based on weights assigned to corners and sides.	Same as Heuristic 3, with local optimisation to block any wins for opponent if possible
Performance Against Random	96%	97%	97%	98%

Results and Analysis

On the whole, we were pretty disappointed with the results. In a lot of cases the heuristics applied did not seem to be reflected in the decisions eventually taken by the bot.

A very notable example of this is in the local optimisation made in Heuristic 4. The way it is supposed to work is: if a move by Our Player (p1) allows Other Player (p2) to win a smallboard in a single move (i.e, it checks if there are two p2 flags in a row), then p1 will not make the move. As evident, it fails:

Case study: Local Check Heuristic Failure

> *Team 14: p2, x*
>
> *Team 24: p1, o*

Game state before move. Move to be made in Row 1, Column 0 of larger board.

```
================BigBoard State================

 x o o  x o x  o x o      - - -  x - -  - - -
 o o x  - - -  x - -      - - -  - - -  - - -
 x x -  - - -  - - -      x - -  - x x  - - -

 - - -  - - -  o o -      x - -  - - -  - - -
 - - -  - - -  - - -      - - -  - - x  - - -
 - - -  - - -  - - -      - - -  - - -  - - -

 o o -  o o o  o - -      - - -  - - -  - - -
 - - -  - - -  - - -      - - -  - - -  - - -
 - - -  - x -  - - -      - - -  - - -  - - -

==============SmallBoards States==============
- - -    - - -
- - -    - - -
- o -    - - -
==============================================
```

p1 identifies (0, 3, 0) as a possible move.

This move allows **p2** to win (0,0) of larger board in the next move. So it should not be allowed. But, **p1** makes the move anyway and **p2** wins the (0,0) board.

```
('CONTINUE', '-')
================BigBoard State================

 x o o  x o x  o x o      - - -  x - -  - - -
```

```
 o o x  - - -  x - -      - - -  - - -  - - -
 x x -  - - -  - - -      x - -  - x x  - - -

 o - -  - - -  o o -      x - -  - - -  - - -
 - - -  - - -  - - -      - - -  - - x  - - -
 - - -  - - -  - - -      - - -  - - -  - - -

 o o -  o o o  o - -      - - -  - - -  - - -
 - - -  - - -  - - -      - - -  - - -  - - -
 - - -  - x -  - - -      - - -  - - -  - - -

==============SmallBoards States==============
- - -    - - -
- - -    - - -
- o -    - - -
==============================================


('CONTINUE', '-')
================BigBoard State================

 x o o  x o x  o x o      - - -  x - -  - - -
 o o x  - - -  x - -      - - -  - - -  - - -
 x x x  - - -  - - -      x - -  - x x  - - -

 o - -  - - -  o o -      x - -  - - -  - - -
 - - -  - - -  - - -      - - -  - - x  - - -
 - - -  - - -  - - -      - - -  - - -  - - -

 o o -  o o o  o - -      - - -  - - -  - - -
 - - -  - - -  - - -      - - -  - - -  - - -
 - - -  - x -  - - -      - - -  - - -  - - -

==============SmallBoards States==============
x - -    - - -
- - -    - - -
- o -    - - -
==============================================
```

A possible reason for this failure is that there is no secondary move option chosen beforehand that the bot can take in such cases. A simple programming error, cost us in at least 2 games (against Bot 13, both that and ours had near identical moves with the exception of win blocking and one other move sequence).

Another

Improvements that can be done