ARTIFICIAL INTELLIGENCE

REPORT

EXTREME TIC TAC TOE

TEAM 6 - 20171213 Rishabh & 20171168 Pratyush [Heuriustic Approach]

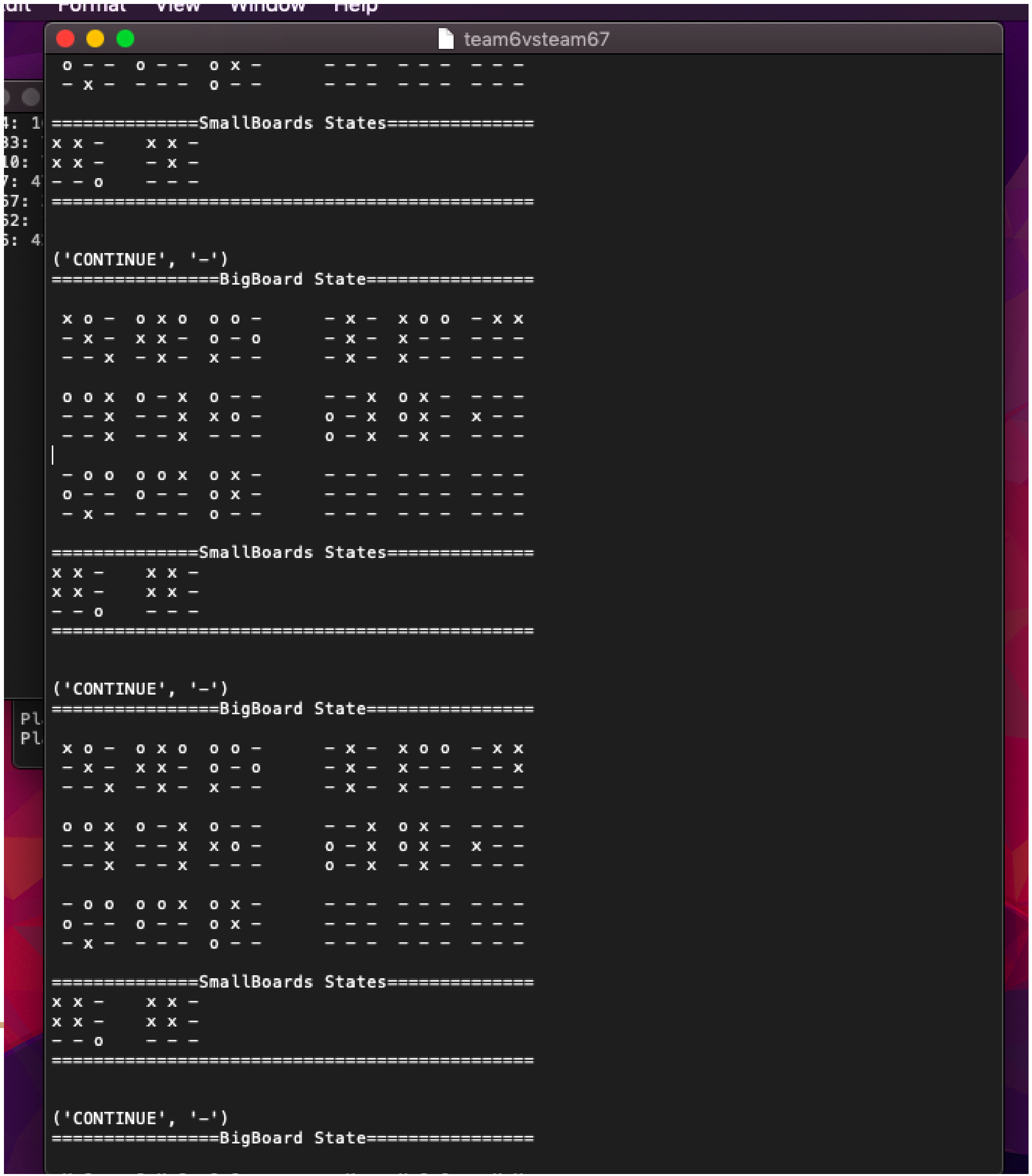
NOTE: Because, there was a bug in the implementation discovered by us as explained at appropriate places in this pdf, late after submission the heuristic didn't work well. And hence these consequence;

Instead of playing X on (0,6) board 2 For winning the match. It moved at different place.

And hence, in later stage went into draw instead of winning.

Hence, losing lots of points.

We got 42 points just because it went into draw, otherwise it



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

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


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

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

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

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

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


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

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

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

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

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


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

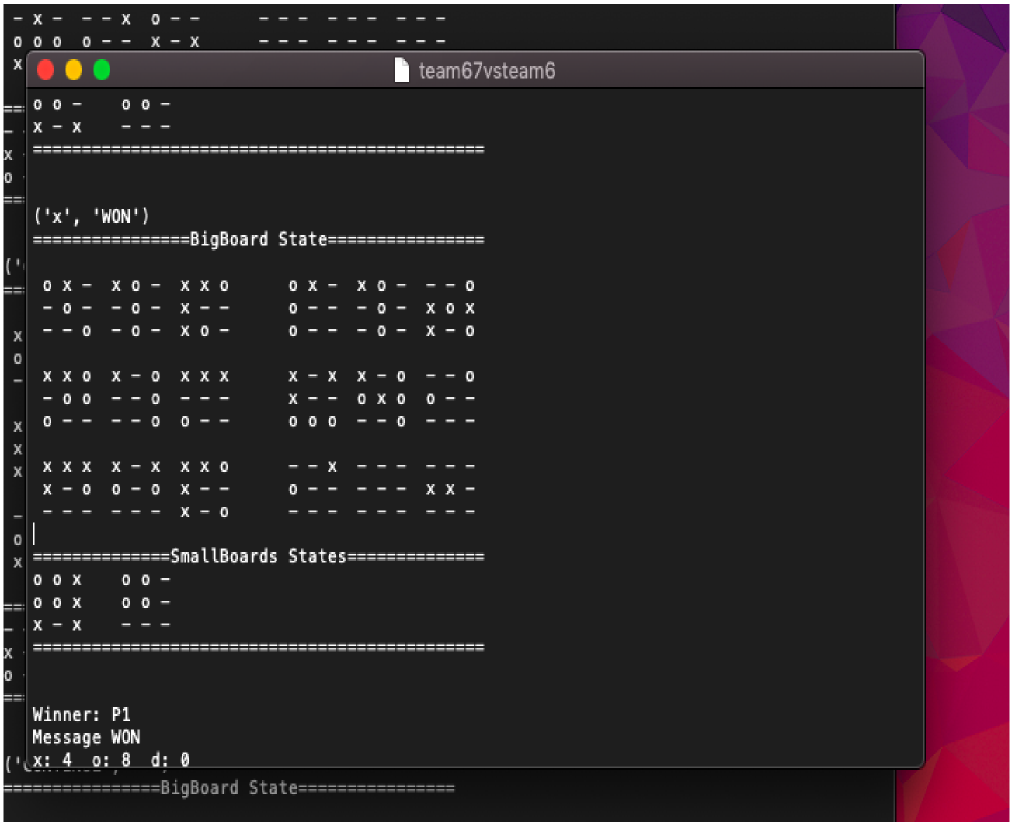
would have been a plain 0.

Similar, thing happened when we played opposite with the same team.

We were able to make 0-0-(-)

But couldn't complete it, because of our bug - when the board state was winning state was of opponent it was returning a very high heuristic

Value where it should have returned a very low heuristic value.



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

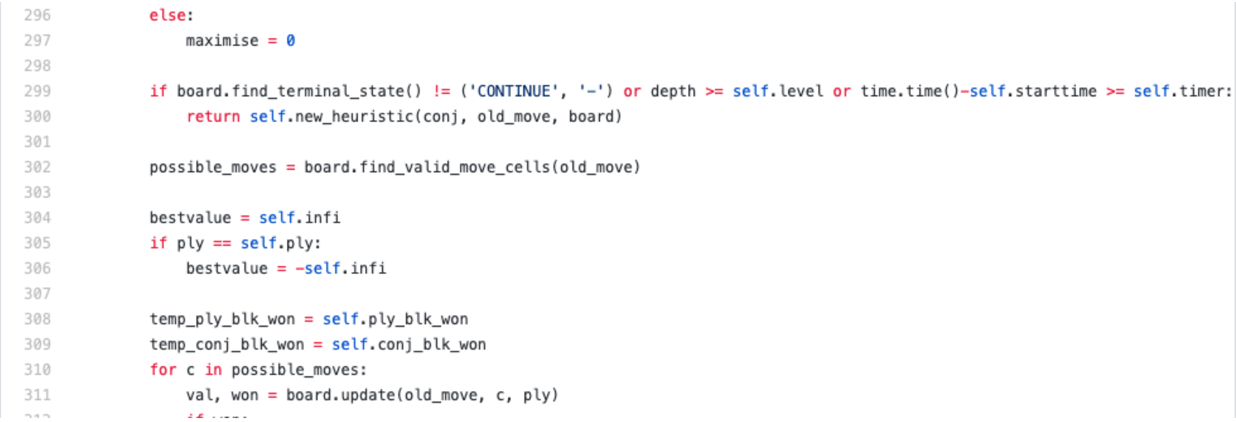
```
o o -    o o -
x - x    - - -
==============================================


('x', 'WON')
================BigBoard State================

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

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

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

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


Winner: P1
Message WON
x: 4  o: 8  d: 0
```

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

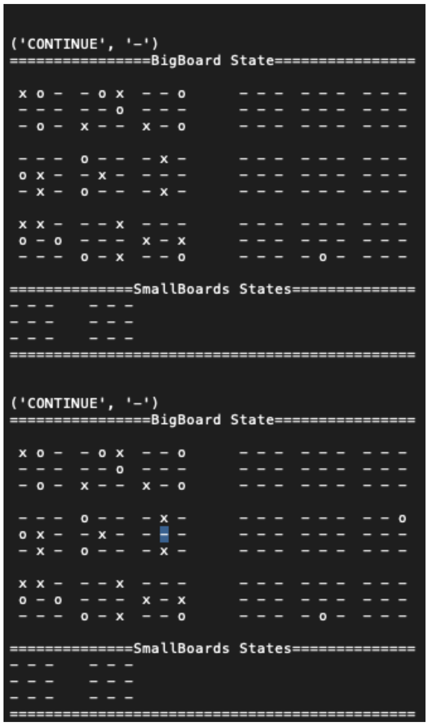
At line no 300 in our code,

We had to pass "ply" as argument to the new_heuristic function instead of "conj". (ply-the player who has to play the current move in the call to minimax , conj-the opponent of ply). In line no 121, we check if board state is won and the chance is of our player then return inf and otherwise -inf. Since we passed conjugate(opponent of our player)as argument to function new_heuristic, it returned inf if the chance was of our conjugate player and -inf if the chance was ours due to which the bot performed wrong moves at last stages.

```
117
118     def new_heuristic(self, ply, move, board):
119         boardstate = board.find_terminal_state()
120         if boardstate[1] == 'WON':
121             if ply == self.ply:
122                 return self.infi
123             else:
124                 return -self.infi
125         for i in range(2):
126             for j in range(3):
127                 for k in range(3):
```



```python
        else:
            maximise = 0

        if board.find_terminal_state() != ('CONTINUE', '-') or depth >= self.level or time.time()-self.starttime >= self.timer:
            return self.new_heuristic(conj, old_move, board)

        possible_moves = board.find_valid_move_cells(old_move)

        bestvalue = self.infi
        if ply == self.ply:
            bestvalue = -self.infi

        temp_ply_blk_won = self.ply_blk_won
        temp_conj_blk_won = self.conj_blk_won
        for c in possible_moves:
            val, won = board.update(old_move, c, ply)
```



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

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

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

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

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

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

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

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

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

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

TEAM33 vs TEAM 6

Here instead of stopping the opponent from winning , our bot played at some new random position which is irrational. It should have moved at center (highlighted) by blue.

Changes in Heuristic :

In this we are now using zobrist hashing for hashing the 3*3 small boards and utilising it for creating a dictionary (on similar lines with dynamic programming).

And Now, for the exact value of heuristic, we are iterating over all small boards of both the big boards and if a small board is already won (or going to be won , as we are looking some steps ahead) we take the value to be some " big positive number ", else if it's won by the opponent a " big negative number ". For the remaining one, we are checking our dictionary if it's already calculated , if not we are calculating it according to the following method :

X-X-X : +1000

X-X-(-) : +100

X-(-)-(-): +1

Considering X is our player. And the values for opponent are exactly opposite.

We calculate this value over all diagnols and rows, columns, and find the cumulative sum which will be the vaue for that small_board.

At the end, for the big board... We check similar combinations of overall wins and according to that add the" heuristic_values of small board ". If still the overall_hash is 0 , we sum over all the "heuristic_values of small board".

Earlier, we were taking :

<u>### Favouring -> Corners: 4 points • Center: 3 points • Remaining SmallBoards: 6 points (weighted heuristic, giving more importance to small boards having more points)</u>

This point in idea, but we later dropped it.

Also, we added zobrist hashing which was not speculated before, 'coz it was giving huge improvements in time complexity.