Report Bot Assignment

Team 16

AI - BOT

HEURISTIC

The heuristic takes into account various factors:

- Winning on a small board is given weight (wt = 1500).
- Winning on big boards is given weight (wt = 2500).
- Blocking on big boards (wt = 1700).
- Blocking on small boards (wt = 1200).
- Having a pattern oo* on big boards (wt = 900)
- Having a pattern oo* on small boards (wt = 700)

The above weights are added in the heuristic value for ‘o’ who is the maximizer and subtracted for ‘x’ who is the minimizer.

Discount Factor

We have also used the concept of Discount factor in cases of the above mentioned factors. We iterate in reverse fashion and multiply the weights for each by a certain discount factor (different for different weights). The intuition behind this is that the rewards at lower depths are less significant than short term rewards (rewards at immediate move).

Defence flag

In the situations where we get a second turn we switch our strategy to defensive mode. In cases where there is a tie in heuristic of different leaves, this helps to choose the move where blocking is given slightly more weight than attacking.

ANALYSIS

Games Playing First

1. Vs team3 Winner: P1
   x: 10 o: 6 d: 0

2. Vs team17 Winner: P2
   x: 5 o: 7 d: 1

3. Vs team41 Winner: P2
   x: 6 o: 5 d: 0

4. Vs team51 Winner: P2
   x: 9 o: 7 d: 0

5. Vs team57 Winner: P2
   x: 8 o: 5 d: 1

6. Vs team55 Winner: P1
   x: 4 o: 3 d: 0

Games Playing Second

1. vs team3 Winner: P1
   x: 4 o: 5 d: 0

2. vs team17 Winner P1
   x: 3 o: 3 d: 0

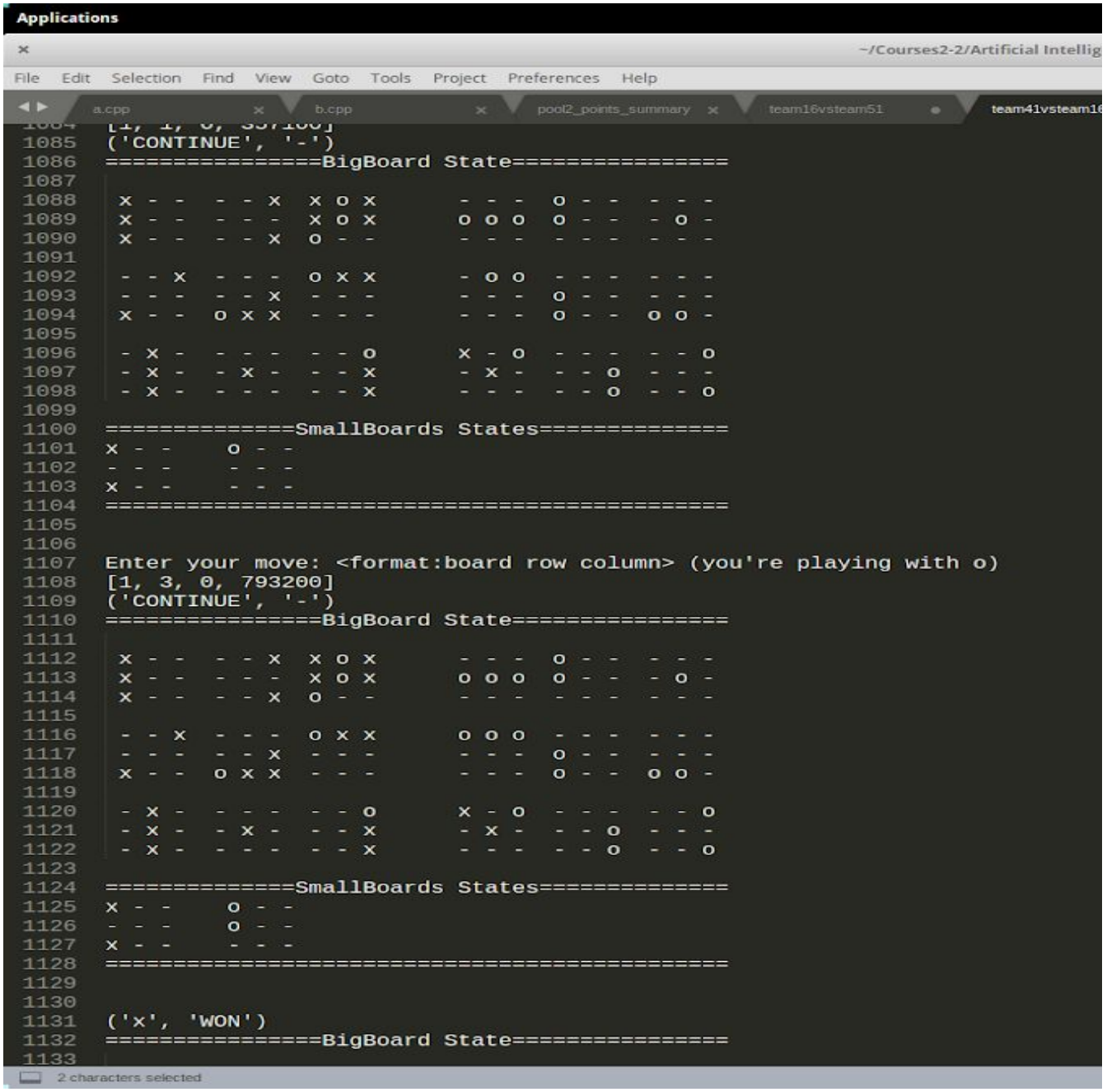
3. vs team41 Winner P1
   x: 3 o: 2 d: 0 (No Blocking performed allowed opponent to make 3 in a row :-(

4. vs team51 Winner P2
   x: 3 o: 6 d: 0 (Won this one but not an efficient win)

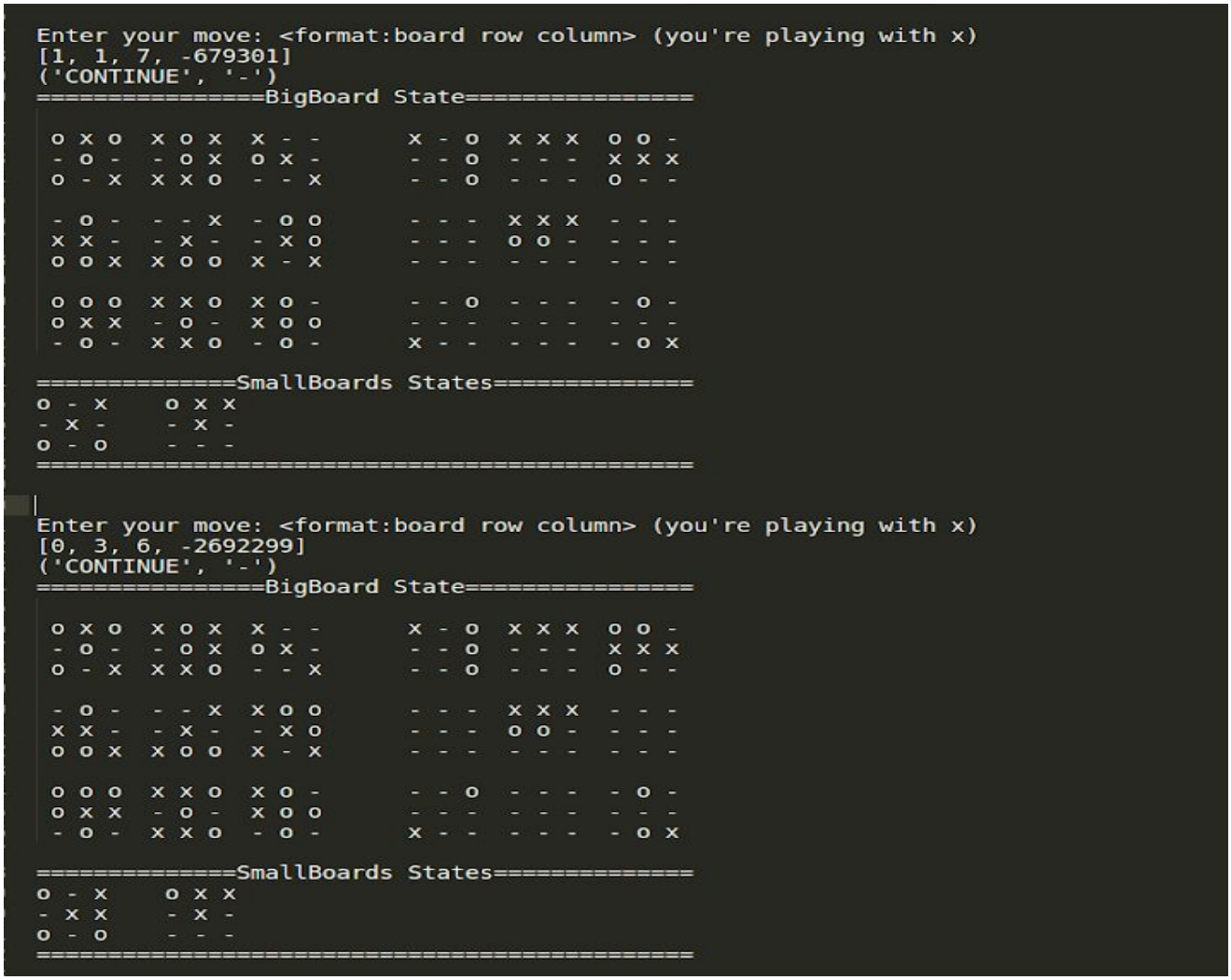
5. vs team55 Winner P2
   x: 5 o: 6 d: 0

6. vs team57 Winner P1
   x: 5 o: 7 d: 0

Some instances of wrong decision



1. The big mistake which we committed here is that instead of blocking X from making a triplet on the left board we gave higher priority to forming a smaller board on the right board. All this issues could have been avoided if discount factors and weights would have been adjusted properly.



2. Here, we had two choices of making a small board, but in one case the opponent will get a open move and will immediately make a small board to win the game, and in other case the opponent will get to make a small board which will make him win the game. But if we had chosen not to win this current small board we could have easily prevented him from winning. This drawback happened majorly because of low depth and wrong values

of parameters of winning small boards.

Drawbacks

In both cases playing first or second we won a lot of small boards but failed to win the big-board , too much weight given to winning the small board win.

Games Won = 4
In games won -- avg. small boards we won = 26/4 = **6.5**
-- avg. small boards opponent won = 17/4 = **4.25**

Games Lost = 8
In games lost -- avg. small boards we won = 45/8 = **5.625**
-- avg. small boards opponent won = 39/8 = **4.875**

Hence even in games we lost , on an average we won more (~ 1) small boards than the opponent. Reason very little weight to oo* and all.

One more drawback was that we were exploring only till depth = 3, which could have been increased to improve the performance. Our bot takes 0.2 seconds to decide a move. We could have increased depth and used a timeout to utilise the time available fully.

Reasons for wrong weights

We were adjusting the weights and didn't have any proper testing mechanism , it was very hard to get the right weights of the different parameters. In the end we drastically changed the weights (bad illogical choice) because it defeated our previous bot convincingly. Basically we hypertuned our bot to defeat the previous version of our bot.