Project 4: Solving (Wild) Tic-Tac-Toe using Minimax Search

CSC 215-01 Artificial Intelligence (Spring 2023)

Raul Arambula (SID:302741322) and Bashar Allwza (SID:301780232)

Due April 24, 2023

Problem Statement

The desire to build A.I.'s that can play games better than humans has been a hot topic for a while. The topic really took off when Deep Blue beat a chess champion. Some of these games are more difficult to build A.I. for due to the branching factor, the complexity of the gameplay, and other factors. The goal here is to create A.I.'s that can play Wild and regular Tic-Tac-Toe using minimax search.

Methodology

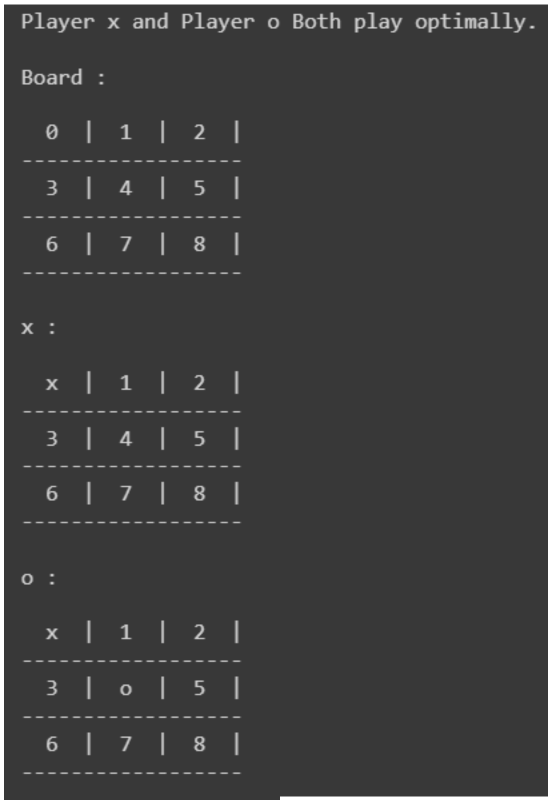
Most of the code was already given. The only function that needed to be finished was the minimax function. For regular Tic-Tac-Toe minimax we iterated over every open move. We then simulated making the move based on the current player and then recorded the score returned for that move when recursively calling the function. Finally, based on whichever player's turn it was either the highest or lowest move was selected and that move was officially taken. The score for that move was returned. For Wild Tic-Tac-Toe every open move is iterated over and a check is made to see if it is the first turn and the move being checked is the middle square. This is necessary for running simulations for when the first player does not make the middle move first. After that check the 'x' and 'o' moves are iterated over. The rest is very similar to regular Tic-Tac-Toe except that whether an 'x' or 'o' was used is recorded, as well as the turn, and whether the middle move can be made. All 10 game simulations were run with the alpha beta minimax algorithms to save time.

Experimental Results and Analysis

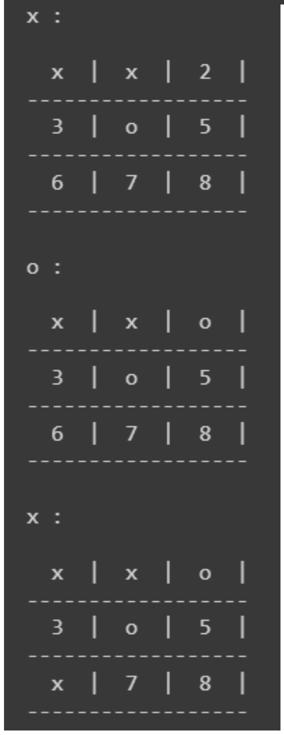
Tic-Tac-Toe W/Minimax

For regular Tic-Tac-Toe with minimax a game of optimal vs optimal takes around 3 seconds. For a game of random vs optimal it takes around .5 seconds.

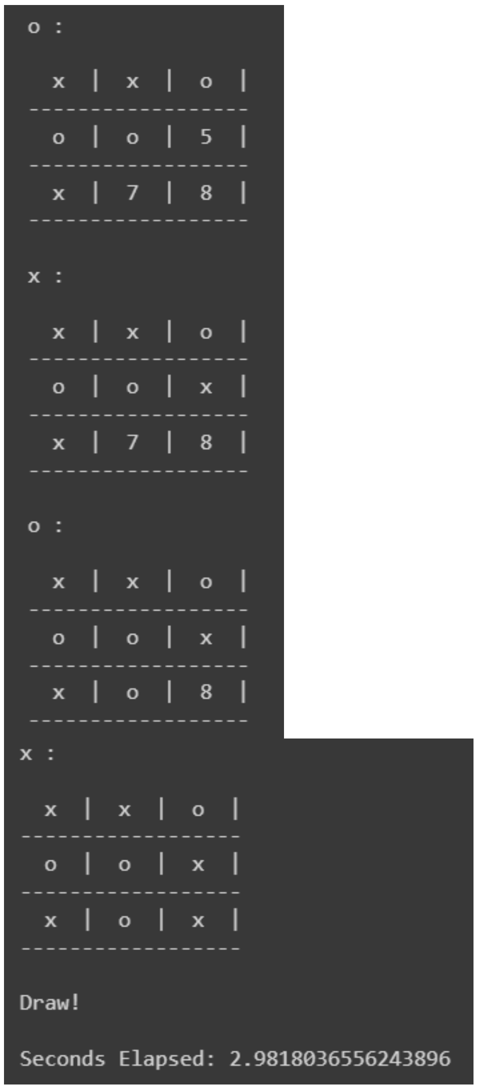
Optimal vs Optimal:



```
Player x and Player o Both play optimally.

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------

x :

  x  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------

o :

  x  |  1  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  8  |
------------------
```

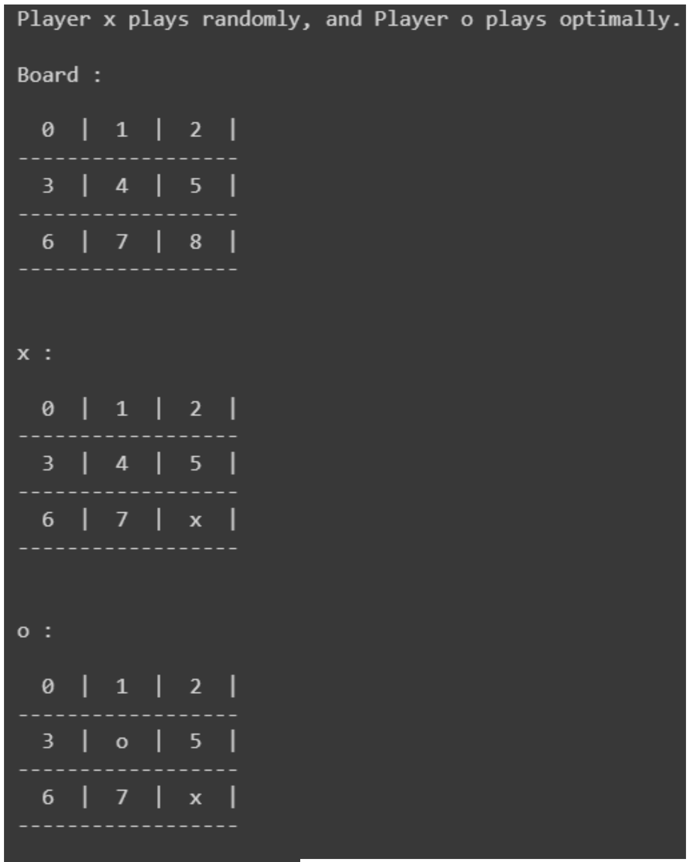


```
x :

  x  |  x  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  8  |
------------------

o :

  x  |  x  |  o  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  8  |
------------------

x :

  x  |  x  |  o  |
------------------
  3  |  o  |  5  |
------------------
  x  |  7  |  8  |
------------------
```

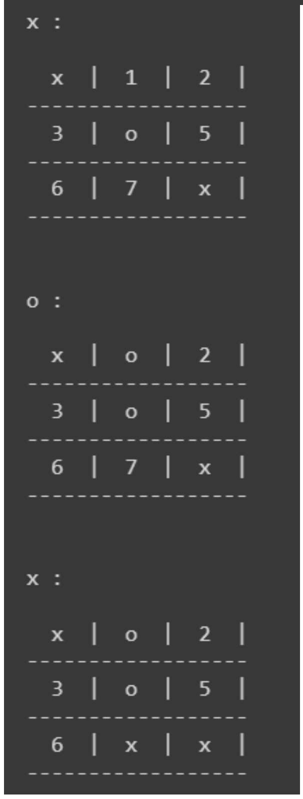


```
o :

  x | x | o |
-----------------
  o | o | 5 |
-----------------
  x | 7 | 8 |
-----------------

x :

  x | x | o |
-----------------
  o | o | x |
-----------------
  x | 7 | 8 |
-----------------

o :

  x | x | o |
-----------------
  o | o | x |
-----------------
  x | o | 8 |
-----------------
x :

  x | x | o |
-----------------
  o | o | x |
-----------------
  x | o | x |
-----------------

Draw!

Seconds Elapsed: 2.9818036556243896
```

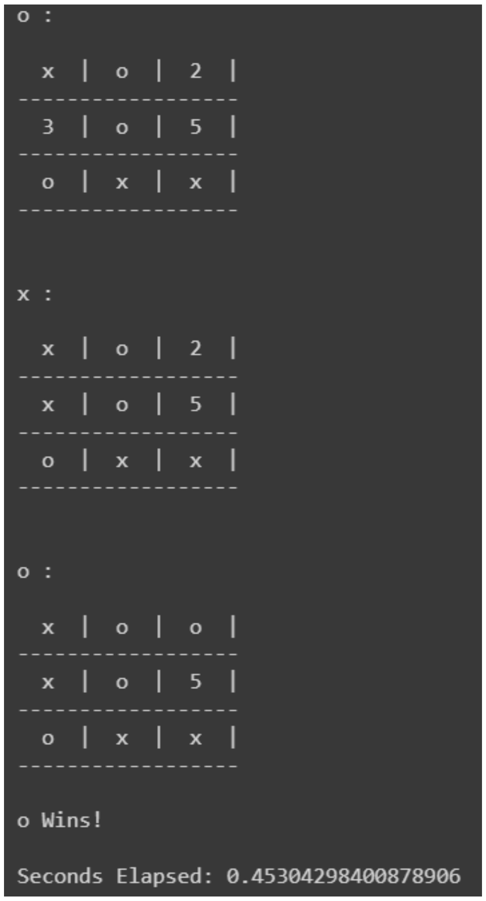
Random vs Optimal:



```
Player x plays randomly, and Player o plays optimally.

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------


x :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  x  |
------------------


o :

  0  |  1  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  x  |
------------------
```



```
x :

  x  |  1  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  x  |
------------------


o :

  x  |  o  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  x  |
------------------


x :

  x  |  o  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  x  |  x  |
------------------
```

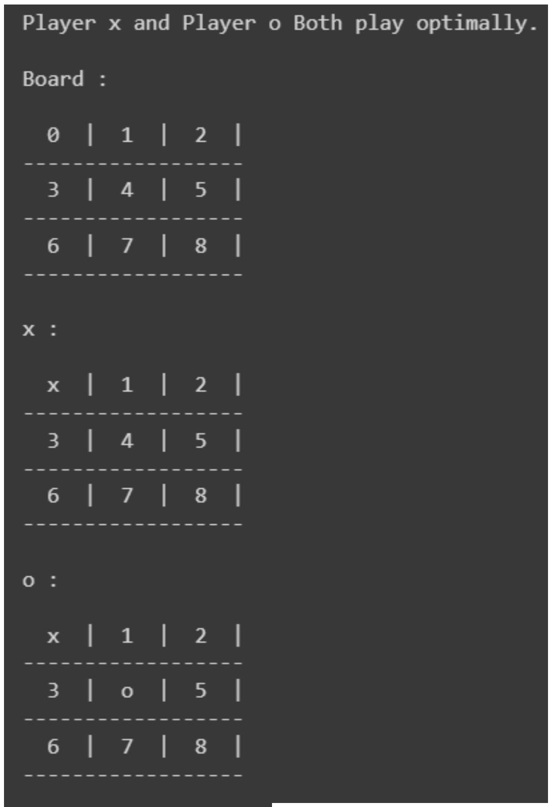


```
o :

  x | o | 2 |
-----------------
  3 | o | 5 |
-----------------
  o | x | x |
-----------------


x :

  x | o | 2 |
-----------------
  x | o | 5 |
-----------------
  o | x | x |
-----------------


o :

  x | o | o |
-----------------
  x | o | 5 |
-----------------
  o | x | x |
-----------------

o Wins!

Seconds Elapsed: 0.45304298400878906
```

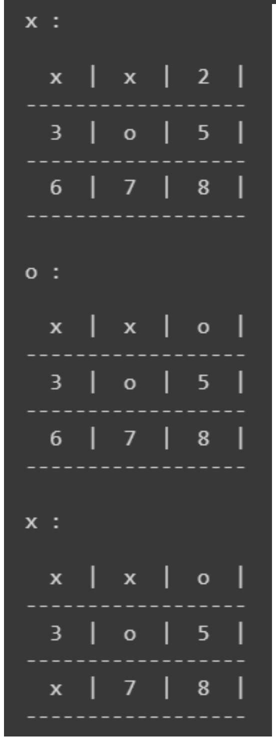
Tic-Tac-Toe W/Alpha Beta Minimax

For regular Tic-Tac-Toe with alpha beta minimax a game of optimal vs optimal takes around .2 seconds. For a game of random vs optimal it takes around .09 seconds. In our 10 game simulations of optimal vs optimal each one ended in a draw. In our 10 game simulations of random vs optimal 4 ended in a draw and 6 ended in victory for the optimal player. All our simulations verified the given characteristics.

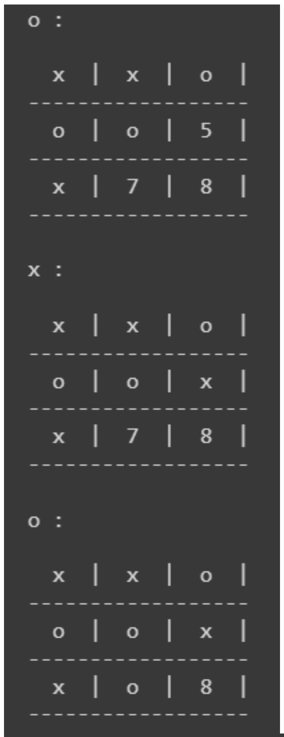
Optimal vs Optimal:



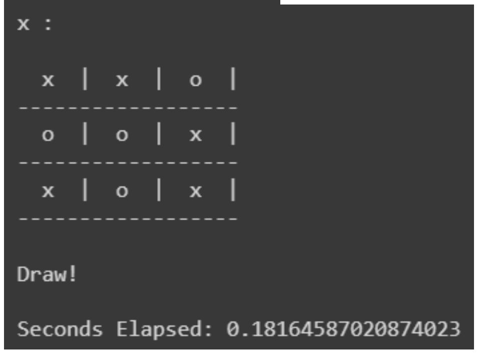
```
Player x and Player o Both play optimally.

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------

x :

  x  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------

o :

  x  |  1  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  8  |
------------------
```



```
x :

  x  |  x  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  8  |
------------------

o :

  x  |  x  |  o  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  8  |
------------------

x :

  x  |  x  |  o  |
------------------
  3  |  o  |  5  |
------------------
  x  |  7  |  8  |
------------------
```

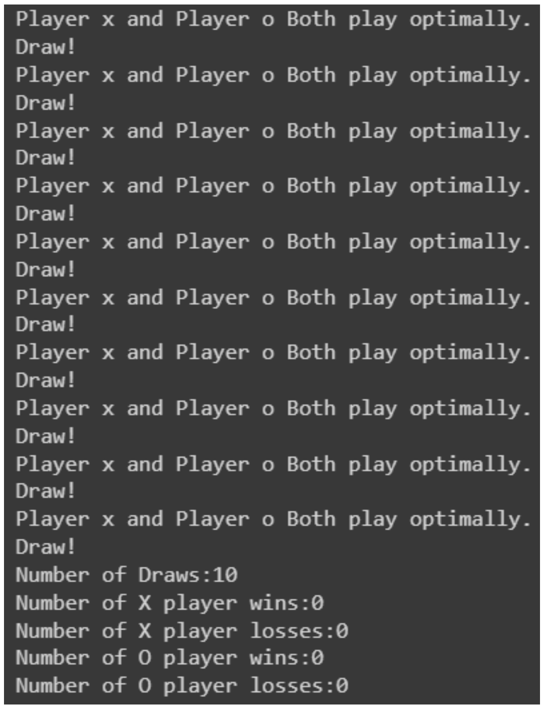


```
o :

  x  |  x  |  o  |
-----------------
  o  |  o  |  5  |
-----------------
  x  |  7  |  8  |
-----------------

x :

  x  |  x  |  o  |
-----------------
  o  |  o  |  x  |
-----------------
  x  |  7  |  8  |
-----------------

o :

  x  |  x  |  o  |
-----------------
  o  |  o  |  x  |
-----------------
  x  |  o  |  8  |
-----------------
```



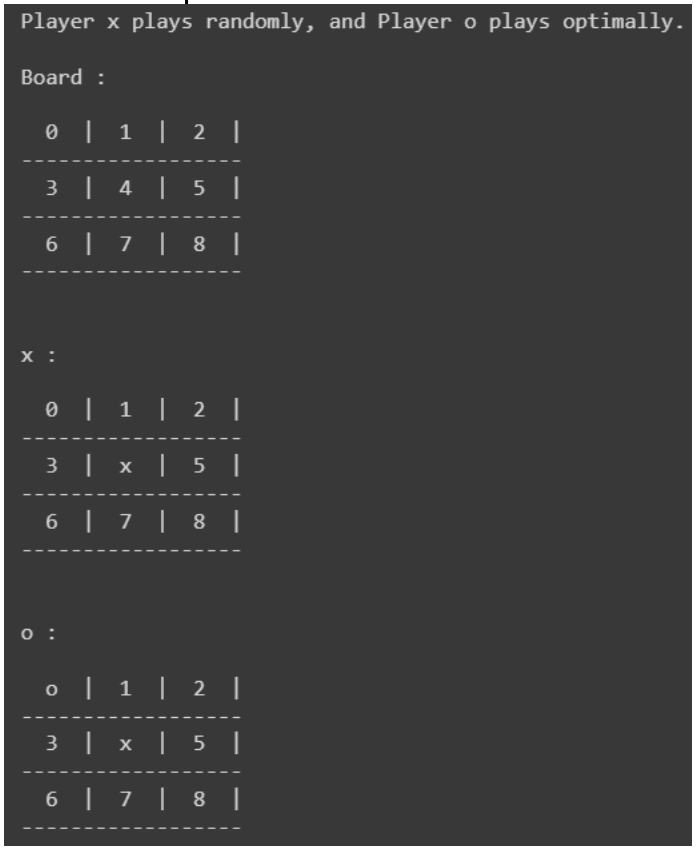
```
x :

  x  |  x  |  o  |
-----------------
  o  |  o  |  x  |
-----------------
  x  |  o  |  x  |
-----------------

Draw!

Seconds Elapsed: 0.18164587020874023
```

Optimal vs Optimal 10 Simulations:

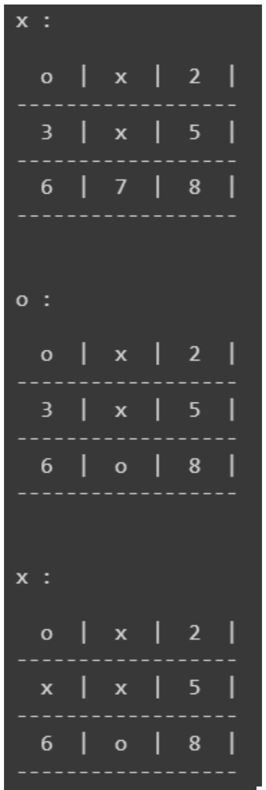


```
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Player x and Player o Both play optimally.
Draw!
Number of Draws:10
Number of X player wins:0
Number of X player losses:0
Number of O player wins:0
Number of O player losses:0
```

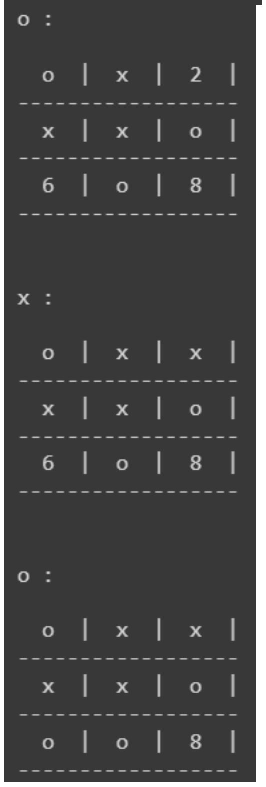
Random vs Optimal:



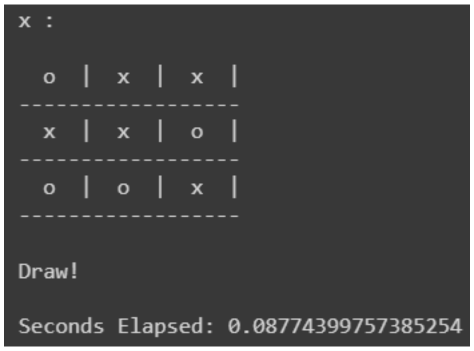
```
Player x plays randomly, and Player o plays optimally.

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------


x :

  0  |  1  |  2  |
------------------
  3  |  x  |  5  |
------------------
  6  |  7  |  8  |
------------------


o :

  o  |  1  |  2  |
------------------
  3  |  x  |  5  |
------------------
  6  |  7  |  8  |
------------------
```



```
x :

  o  |  x  |  2  |
-----------------
  3  |  x  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


o :

  o  |  x  |  2  |
-----------------
  3  |  x  |  5  |
-----------------
  6  |  o  |  8  |
-----------------


x :

  o  |  x  |  2  |
-----------------
  x  |  x  |  5  |
-----------------
  6  |  o  |  8  |
-----------------
```

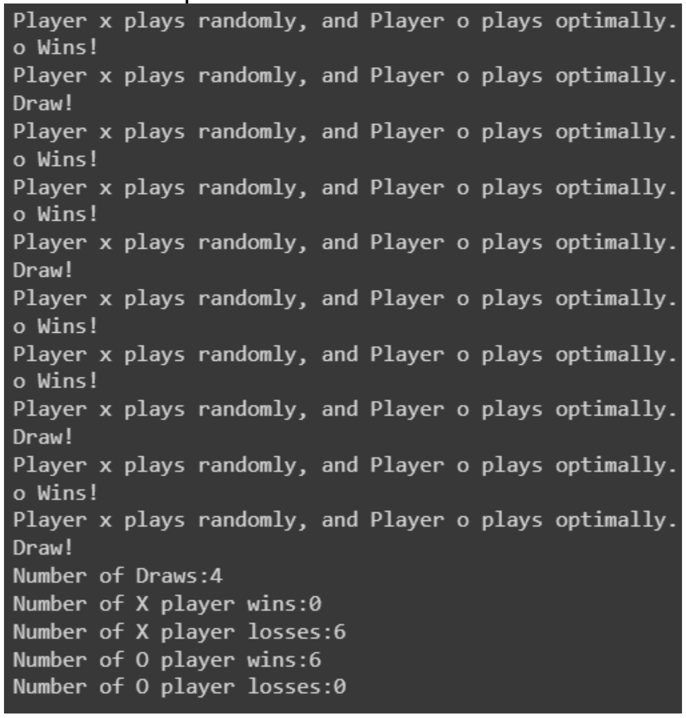


```
o :

  o  |  x  |  2  |
-----------------
  x  |  x  |  o  |
-----------------
  6  |  o  |  8  |
-----------------


x :

  o  |  x  |  x  |
-----------------
  x  |  x  |  o  |
-----------------
  6  |  o  |  8  |
-----------------


o :

  o  |  x  |  x  |
-----------------
  x  |  x  |  o  |
-----------------
  o  |  o  |  8  |
-----------------
```



```
x :

  o | x | x |
-----------------
  x | x | o |
-----------------
  o | o | x |
-----------------

Draw!

Seconds Elapsed: 0.08774399757385254
```

Random vs Optimal 10 Simulations:

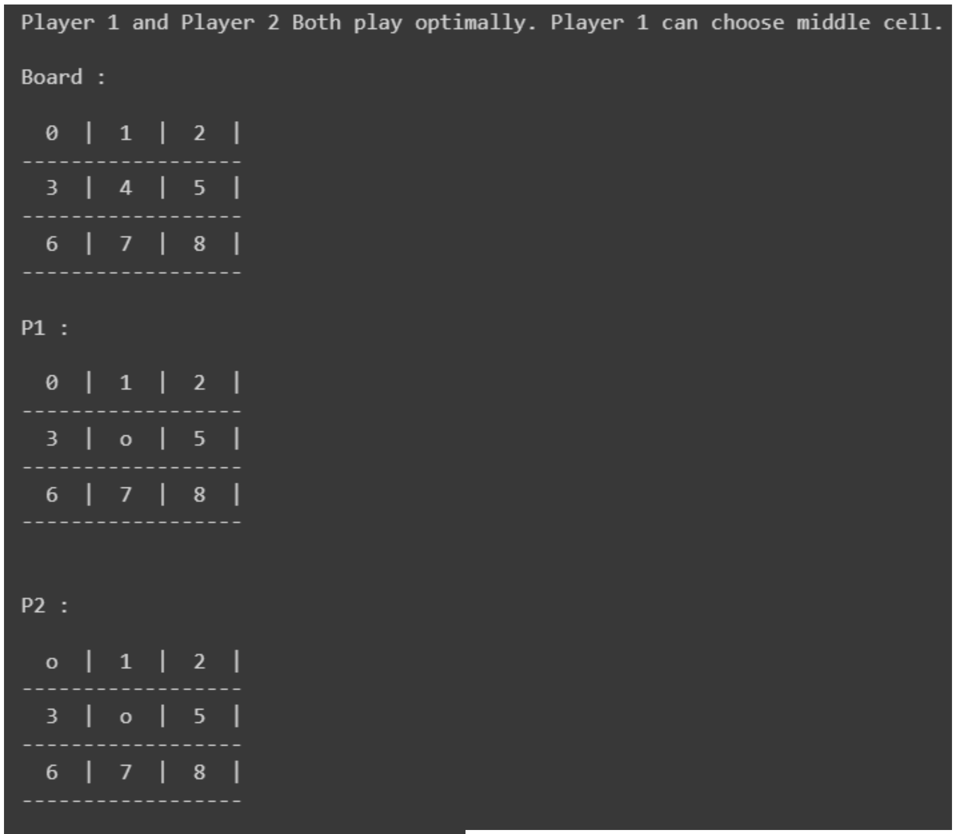


```
Player x plays randomly, and Player o plays optimally.
o Wins!
Player x plays randomly, and Player o plays optimally.
Draw!
Player x plays randomly, and Player o plays optimally.
o Wins!
Player x plays randomly, and Player o plays optimally.
o Wins!
Player x plays randomly, and Player o plays optimally.
Draw!
Player x plays randomly, and Player o plays optimally.
o Wins!
Player x plays randomly, and Player o plays optimally.
o Wins!
Player x plays randomly, and Player o plays optimally.
Draw!
Player x plays randomly, and Player o plays optimally.
o Wins!
Player x plays randomly, and Player o plays optimally.
Draw!
Number of Draws:4
Number of X player wins:0
Number of X player losses:6
Number of O player wins:6
Number of O player losses:0
```

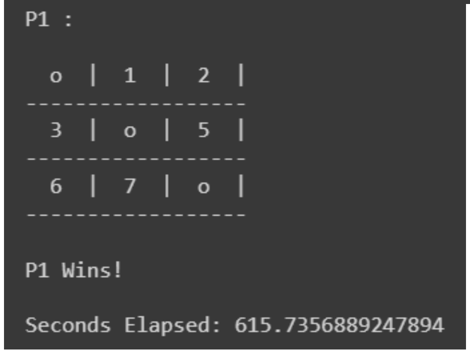
Wild Tic-Tac-Toe W/Minimax

For Wild Tic-Tac-Toe with minimax a game of optimal vs optimal takes around 615 seconds. For a game of random vs optimal it takes around 30 seconds.

Optimal vs Optimal:

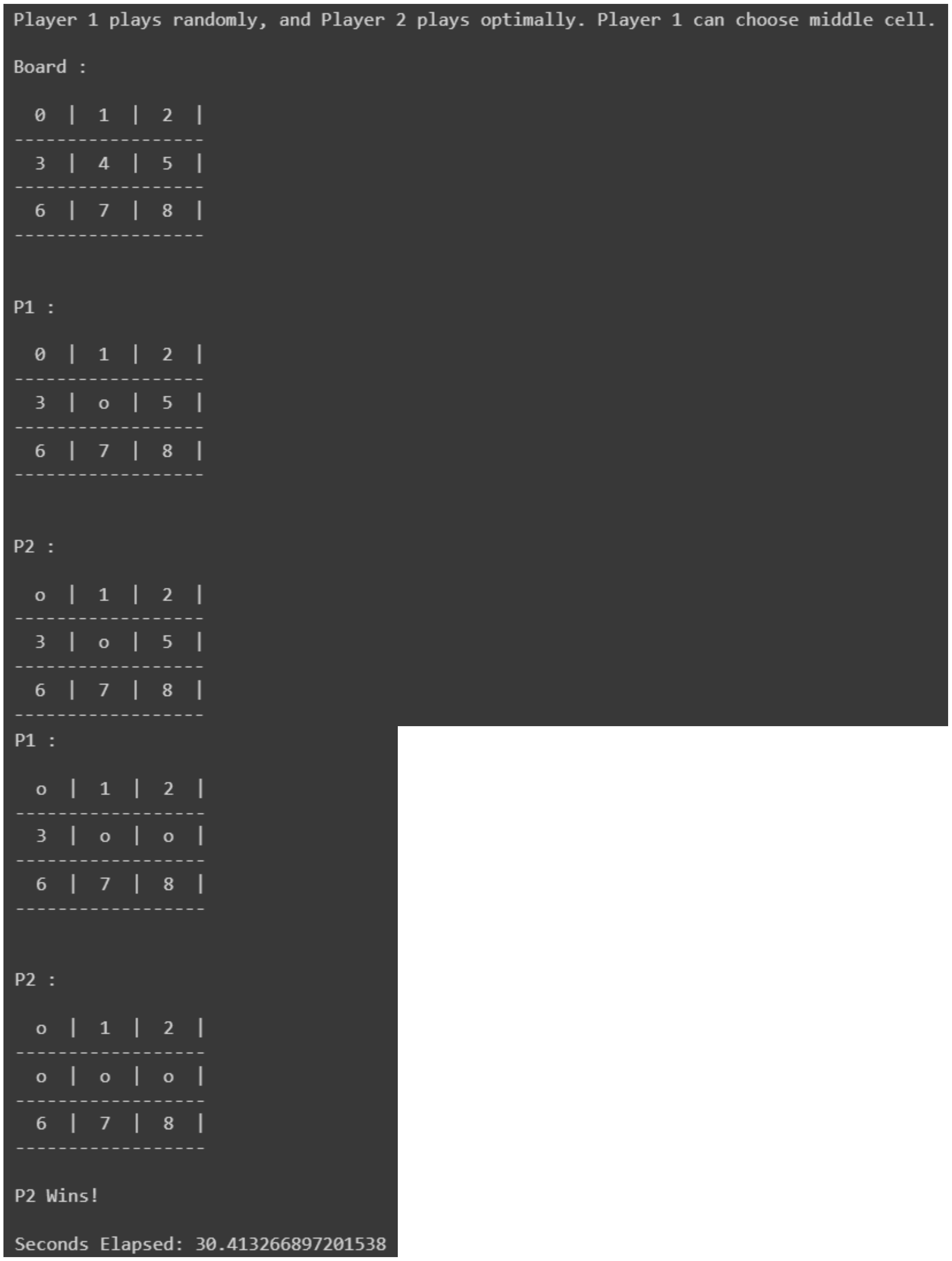


```
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.

Board :

  0  |  1  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------

P1 :

  0  |  1  |  2  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


P2 :

  o  |  1  |  2  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------
```



```
P1 :

  o  |  1  |  2  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  o  |
-----------------

P1 Wins!

Seconds Elapsed: 615.7356889247894
```

Random vs Optimal:

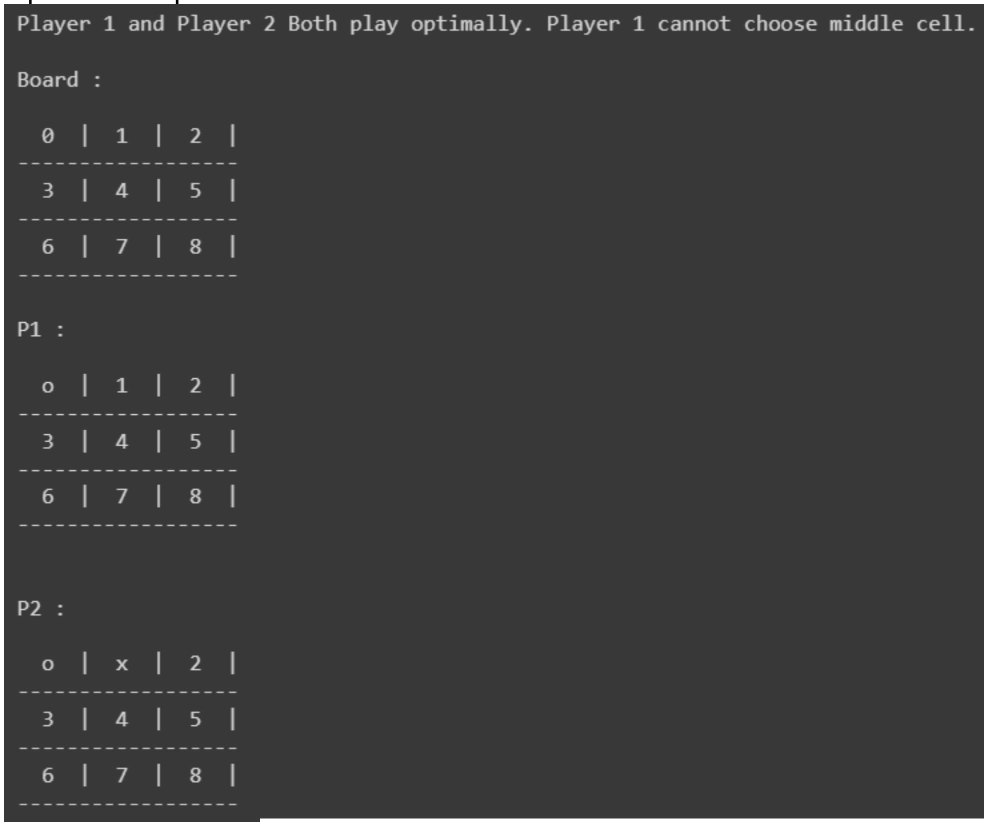


```
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.

Board :

  0  |  1  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


P1 :

  0  |  1  |  2  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


P2 :

  o  |  1  |  2  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------
P1 :

  o  |  1  |  2  |
-----------------
  3  |  o  |  o  |
-----------------
  6  |  7  |  8  |
-----------------


P2 :

  o  |  1  |  2  |
-----------------
  o  |  o  |  o  |
-----------------
  6  |  7  |  8  |
-----------------

P2 Wins!

Seconds Elapsed: 30.413266897201538
```

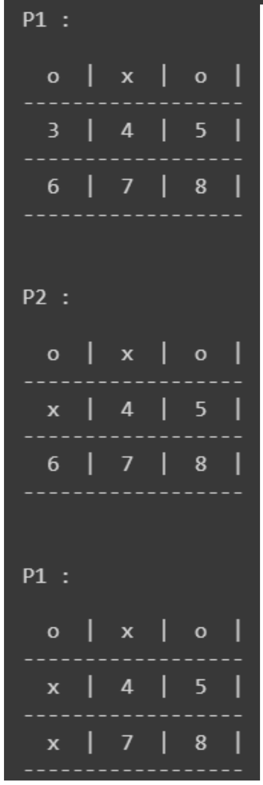
Wild Tic-Tac-Toe W/Alpha Beta Minimax

For Wild Tic-Tac-Toe with alpha beta minimax a game of optimal vs optimal takes around 15 seconds. For a game of random vs optimal it takes around 3 seconds. In our 10 game simulations of optimal vs optimal with player one allowed to use the middle square each one ended in a victory for player one. In our 10 game simulations of optimal vs optimal with player one not allowed to use the middle square each one ended in a draw. In our 10 game simulations of random vs optimal with player one being random each one ended in victory for

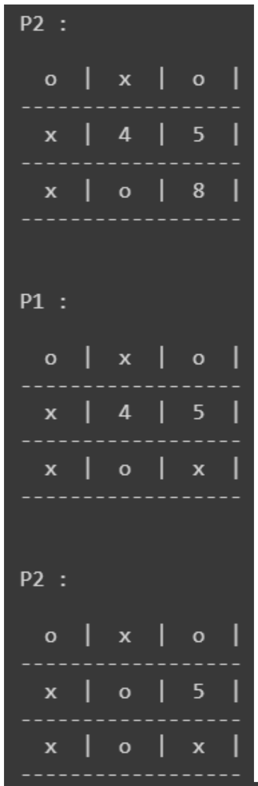
player two. In our 10 game simulations of random vs optimal with player two being random each one ended in victory for player one. All our simulations verified the given characteristics.
Optimal vs Optimal:



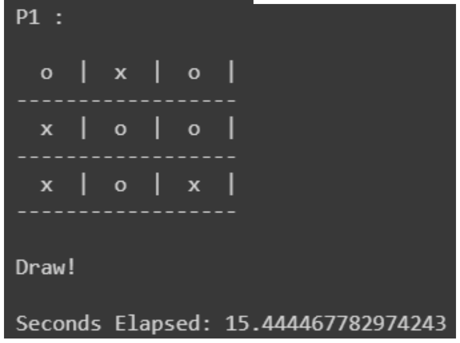
```
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.

Board :

  0  |  1  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------

P1 :

  o  |  1  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


P2 :

  o  |  x  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------
```



```
P1 :

  o  |  x  |  o  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


P2 :

  o  |  x  |  o  |
-----------------
  x  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


P1 :

  o  |  x  |  o  |
-----------------
  x  |  4  |  5  |
-----------------
  x  |  7  |  8  |
-----------------
```

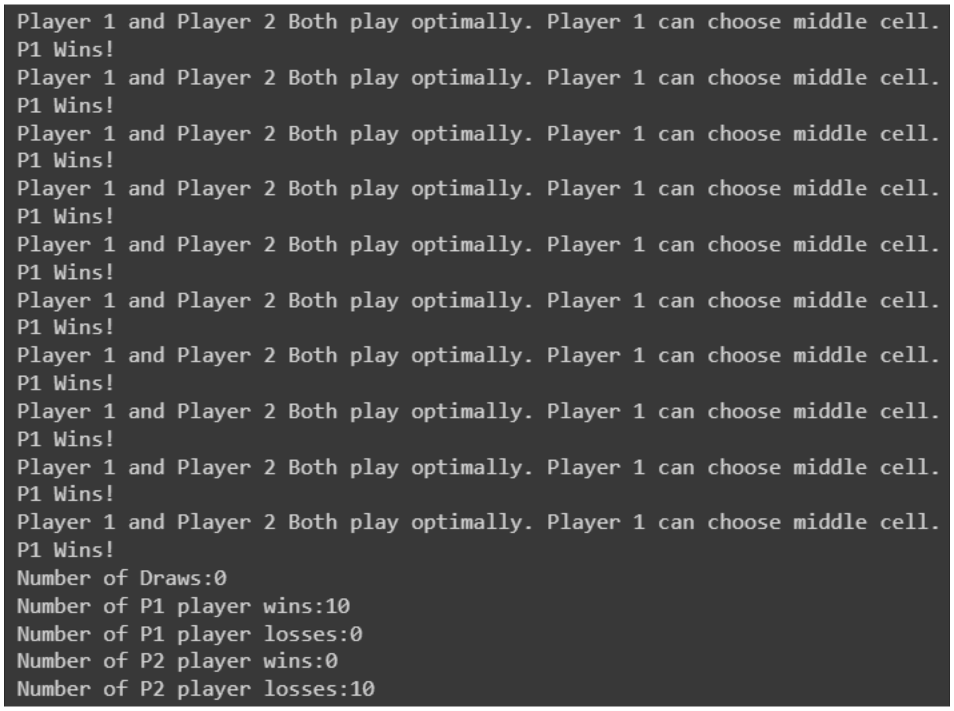


```
P2 :

  o | x | o |
-----------------
  x | 4 | 5 |
-----------------
  x | o | 8 |
-----------------


P1 :

  o | x | o |
-----------------
  x | 4 | 5 |
-----------------
  x | o | x |
-----------------


P2 :

  o | x | o |
-----------------
  x | o | 5 |
-----------------
  x | o | x |
-----------------
```



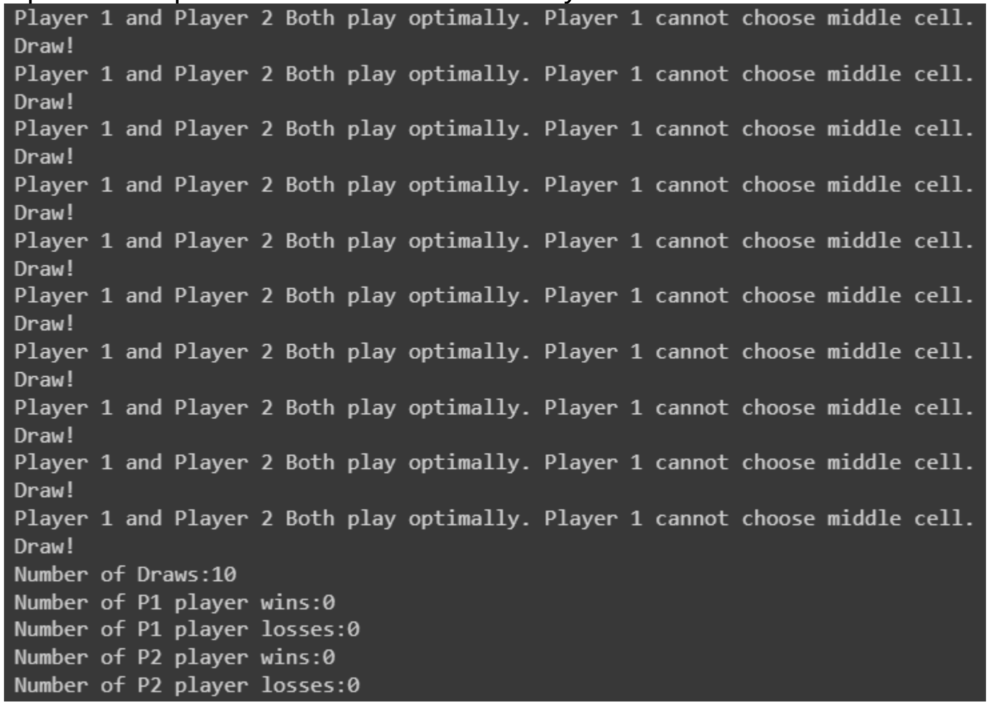
```
P1 :

  o | x | o |
-----------------
  x | o | o |
-----------------
  x | o | x |
-----------------

Draw!

Seconds Elapsed: 15.444467782974243
```

Optimal vs Optimal 10 Simulations with Player One allowed to use middle square:



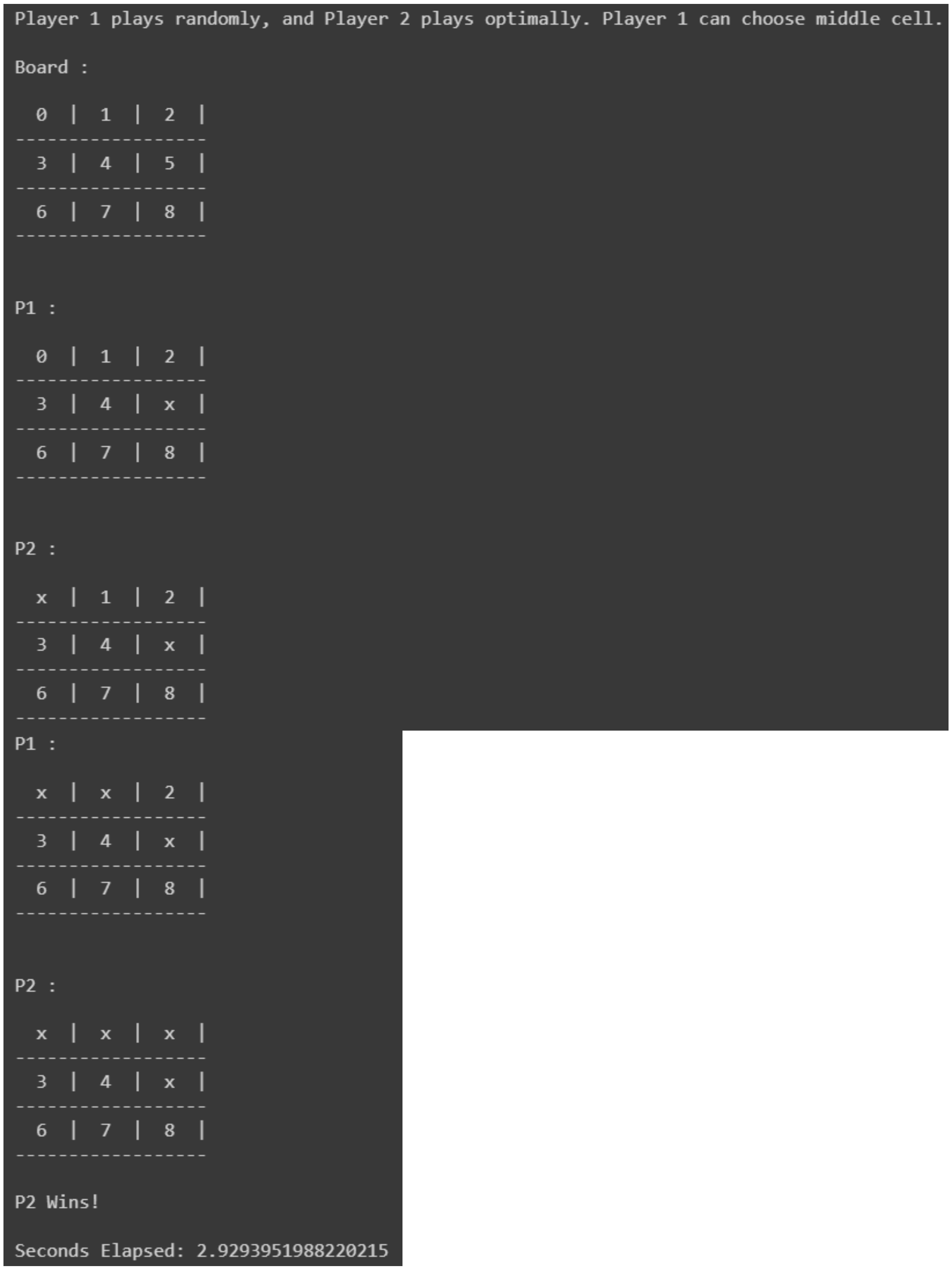
```
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Player 1 and Player 2 Both play optimally. Player 1 can choose middle cell.
P1 Wins!
Number of Draws:0
Number of P1 player wins:10
Number of P1 player losses:0
Number of P2 player wins:0
Number of P2 player losses:10
```

Optimal vs Optimal 10 Simulations with Player One not allowed to use middle square:



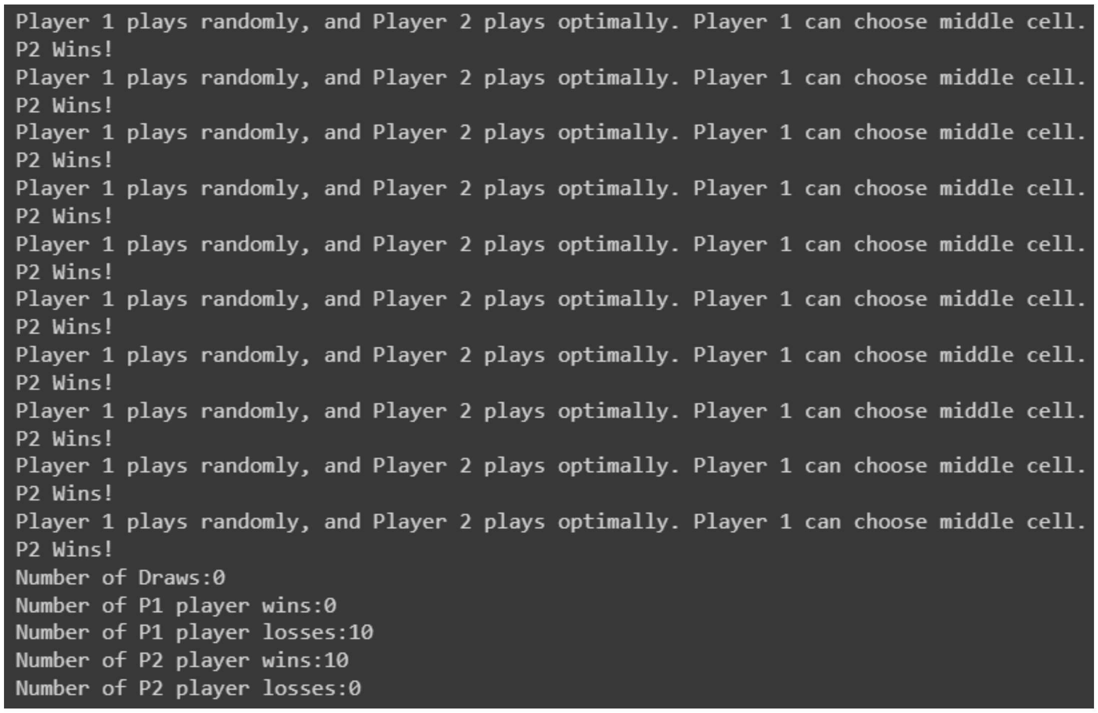
```
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Player 1 and Player 2 Both play optimally. Player 1 cannot choose middle cell.
Draw!
Number of Draws:10
Number of P1 player wins:0
Number of P1 player losses:0
Number of P2 player wins:0
Number of P2 player losses:0
```

Random vs Optimal:



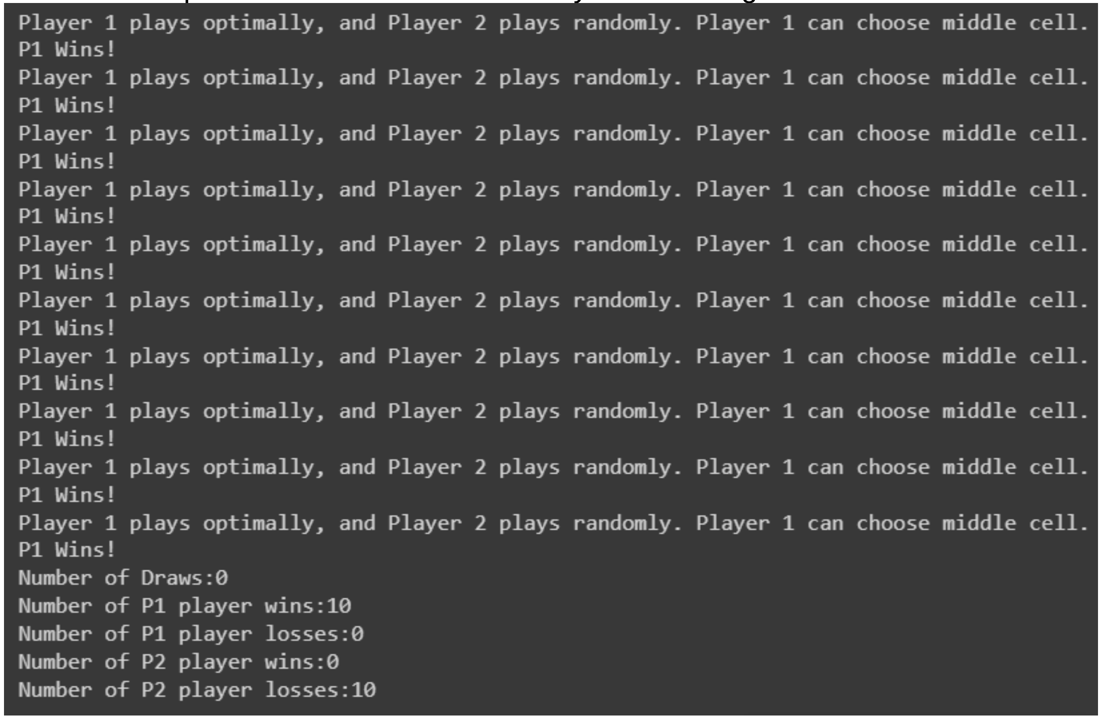
```
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------


P1 :

  0  |  1  |  2  |
------------------
  3  |  4  |  x  |
------------------
  6  |  7  |  8  |
------------------


P2 :

  x  |  1  |  2  |
------------------
  3  |  4  |  x  |
------------------
  6  |  7  |  8  |
------------------
P1 :

  x  |  x  |  2  |
------------------
  3  |  4  |  x  |
------------------
  6  |  7  |  8  |
------------------


P2 :

  x  |  x  |  x  |
------------------
  3  |  4  |  x  |
------------------
  6  |  7  |  8  |
------------------

P2 Wins!

Seconds Elapsed: 2.9293951988220215
```

Random vs Optimal 10 Simulations with Player One being random:



```
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Player 1 plays randomly, and Player 2 plays optimally. Player 1 can choose middle cell.
P2 Wins!
Number of Draws:0
Number of P1 player wins:0
Number of P1 player losses:10
Number of P2 player wins:10
Number of P2 player losses:0
```

Random vs Optimal 10 Simulations with Player Two being random:



```
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Player 1 plays optimally, and Player 2 plays randomly. Player 1 can choose middle cell.
P1 Wins!
Number of Draws:0
Number of P1 player wins:10
Number of P1 player losses:0
Number of P2 player wins:0
Number of P2 player losses:10
```

Task Division and Project Reflection

Work was done together as a team with online meetings. All code was written together in a pair programming style. The report was written together simultaneously. The first challenge was correctly implementing alpha beta pruning. This difficulty was overcome by searching for guides and tutorials. Another challenge was implementing whether the first player could make their first move in the middle square. This was solved through trial and error. As a team we have learned cooperation and effective communication regarding pair programming. We also learned that alpha beta pruning greatly improves the runtime of minimax. Based on these conclusions we suspect that minimax with alpha beta pruning and depth limited search with a heuristic greatly improves runtime even more.

Additional Features

The additional feature that we implemented was alpha beta pruning. Our implementation of alpha beta minimax for regular Tic-Tac-Toe is not much different from minimax without alpha beta pruning. The main difference is that after recording the score for the ‘x’ player alpha is set equal to the max of all scores, after recording the score of the ‘o’ player beta is set equal to the min of all scores, and after all this while still in the for loop, if beta is less than or equal to alpha we break out of the loop. Our implementation of alpha beta minimax for Wild Tic-Tac-Toe is not much different from minimax without alpha beta pruning. The main difference is that after recording the score for an ‘x’ move or a ‘o’ move a check is made to see what player is making the move. If player one is making the move then alpha is set equal to the max of all scores, if player two is making a move then beta is set equal to the min of all scores, and then after all this while still in the for loop, if beta is less than or equal to alpha we break out of the loop.