Tic-Tac-Toe

Firstly, I've Implemented two AI functions by completely removing the human input so that two AI can battle each other. Secondly, both the AI pretty much follow the same patterns of countering the other but despite that each time the game is initiated there's always a different winner which is the outcome I was hoping for (Hahaha) *PS: Ignore the AI name I picked them from Sword Art Online, the anime I was watching*

Ordinal AI CODE:

```
def get_ordinal_ai_move(self):
    '''Get a ordinal AI player move'''
    board = self.board
    if self.move == ' ':
        return randrange(9)
    elif self.move == 1:
        return randrange(3, 9)
    elif board[1] == ' ':
        return 1
    else:
        for row in self.WIN_SET:
            if self.move in row:
                if self.board[row[0]] == ' ':
                    return row[0]
                    break
                if self.board[row[1]] == ' ':
                    return row[1]
                    break
                if self.board[row[2]] == ' ':
                    return row[2]
                    break
```

Explanation: First and foremost by using the self.move we check if the board is empty if it is then just put a random number from the range(9)[0,8] otherwise it would check for the next condition to see if the block '1' on the board has been occupied and if it is then randomly choose from 6 positions on the board else check if the position 1on board is empty if it is then choose that position. Else search for possible wins in the Win_Set and check if one of the individual rows is empty take that spot before the enemy does.

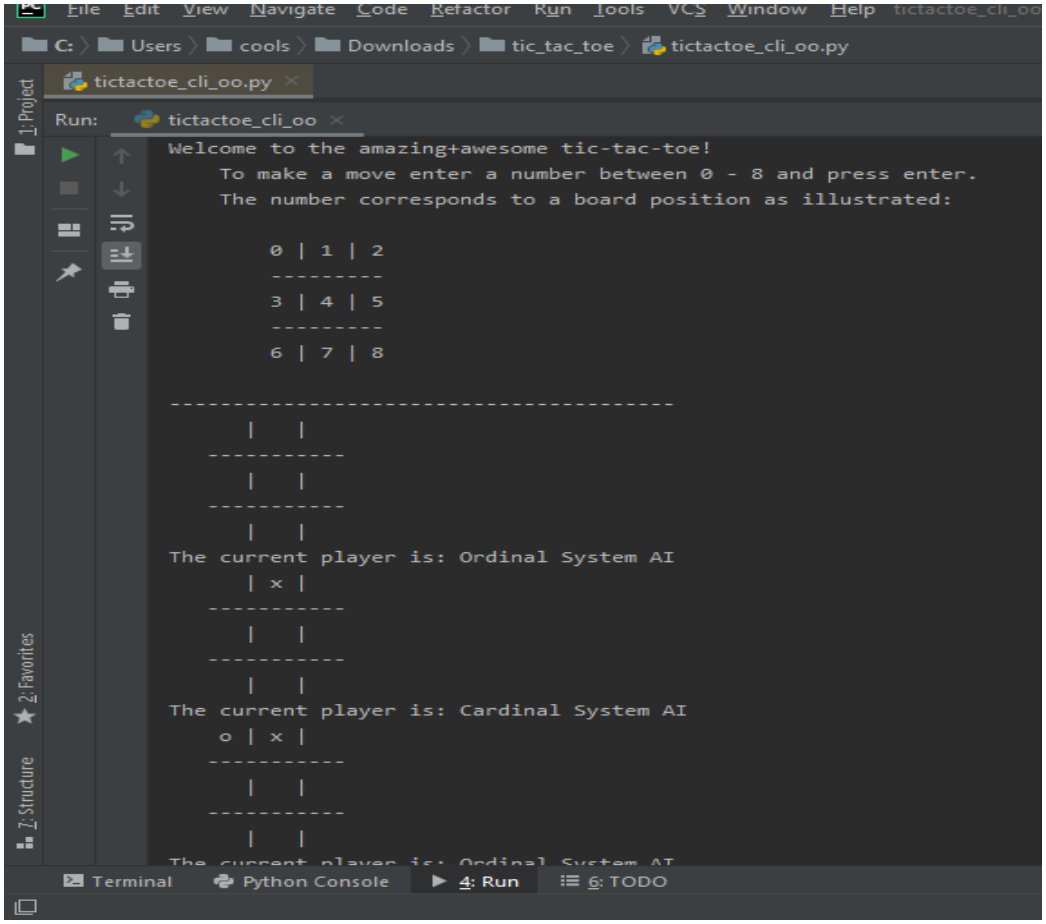
Cardinal AI CODE:

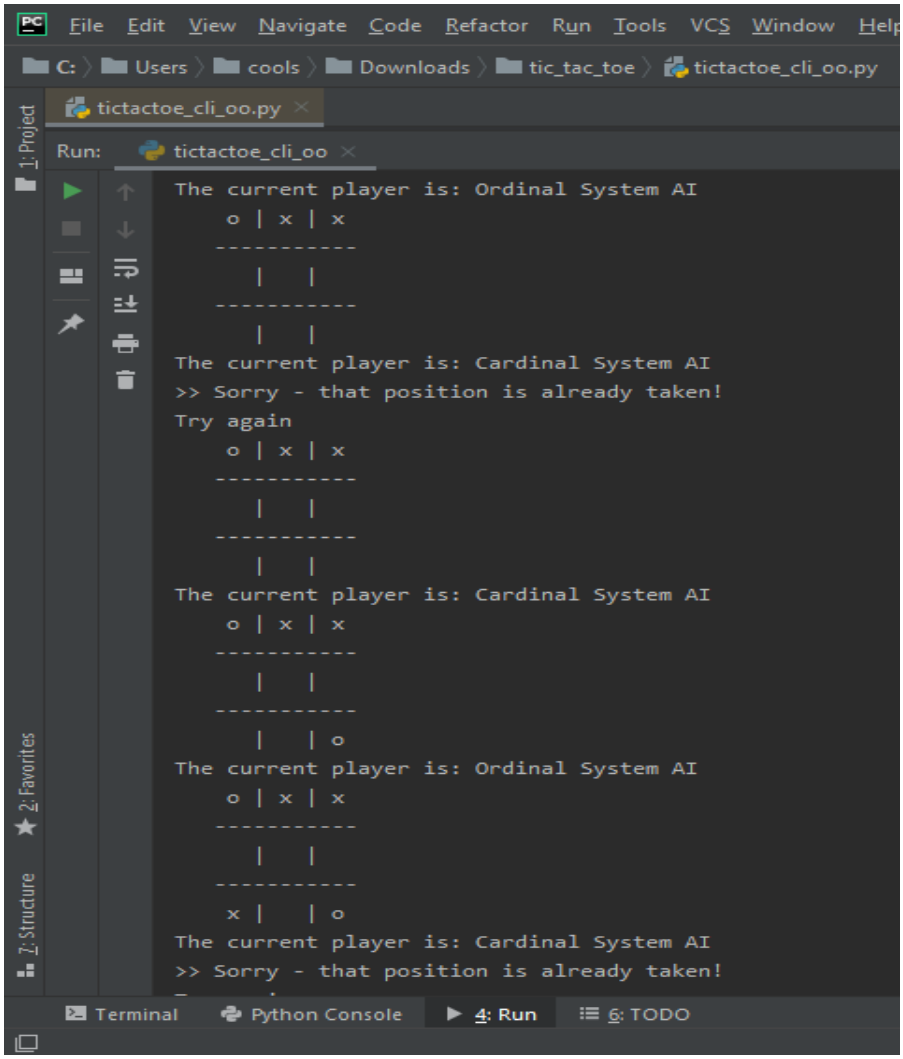
```
def get_cardinal_ai_move(self):
    '''Get the cardinal AI's next move '''
    # A simple dumb random move - valid or NOT!
    # Note: It is the models responsibility to check for
valid moves...
    board = self.board
    if self.move == 0:
        return randrange(1,9)
    elif board[0] == ' ':
        return 0
    else:
        for row in self.WIN_SET:
            if board[row[0]] == board[row[1]] != ' ':
                if board[row[2]] == ' ':
                    return row[2]
                    break
            elif board[row[0]] == board[row[2]] != ' ':
                if board[row[1]] == ' ':
                    return row[1]
                    break
            elif board[row[1]] == board[row[2]] != ' ':
                if board[row[0]] == ' ':
                    return row[0]
                    break
            else:
                return randrange(9)
```

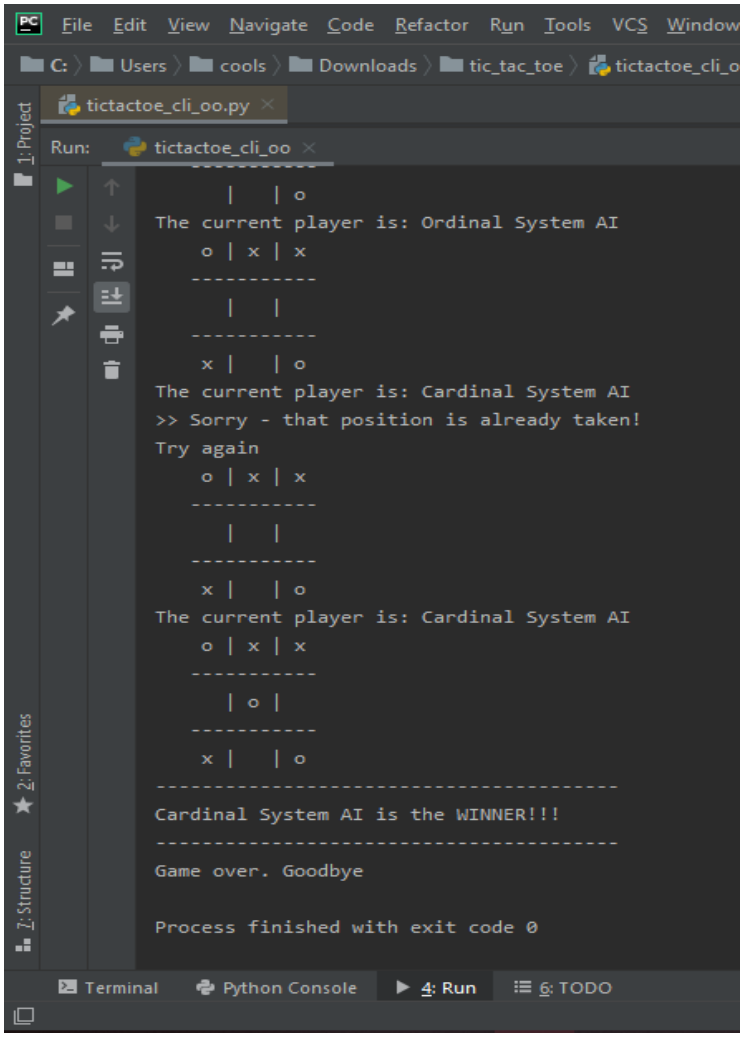
Explanation: Cardinal here pretty much does the same thing except that instead of starting it's initial position from1 it starts from zero so that it's random position options are more than that of ordinal system AI and with regards to for loop; it checks that if corresponding rows in Win_Set patterns has any empty slots remaining to prevent enemy from taking up consecutive positions on the board, in order to do so it takes up that slot for itself. To elaborate more, if you see the first if condition it checks if the row with position 0 and row with position 1 in the board are not

empty if they are not then it most likely means that they’re being occupied by the enemy in the worse case if that is true take up the adjacent position in the Win_set to prevent a win or score a win for itself.

Some Screen Captures of Running:




File Edit View Navigate Code Refactor Run Tools VCS Window Help
C: Users cools Downloads tic_tac_toe tictactoe_cli_oo.py
tictactoe_cli_oo.py
Run: tictactoe_cli_oo
The current player is: Ordinal System AI
o | x | x
The current player is: Cardinal System AI
>> Sorry - that position is already taken!
Try again
o | x | x
The current player is: Cardinal System AI
o | x | x
o
The current player is: Ordinal System AI
o | x | x
x | | o
The current player is: Cardinal System AI
>> Sorry - that position is already taken!
Terminal
Python Console
4: Run
6: TODO
1: Project
2: Favorites
7: Structure



Different Outcomes:

1. (TIE)

```
The current player is: Cardinal System AI
    o | x | x
   -----------
    x | o | o
   -----------
    o | x |
The current player is: Ordinal System AI
    o | x | x
   -----------
    x | o | o
   -----------
    o | x | x
----------------------------------------
TIE!
----------------------------------------
Game over. Goodbye

Process finished with exit code 0
```

2. Ordinal System AI Winner

3. Cardinal System AI Winner