Name:P.Simhadri

H.No:2203a51777

[1]:
```
import random
import math
import functools

class Game:
    def actions(self, state):
        """Return a collection of the allowable moves from this state."""
        raise NotImplementedError

    def result(self, state, move):
        """Return the state that results from making a move from a state."""
        raise NotImplementedError

    def is_terminal(self, state):
        """Return True if this is a final state for the game."""
        return not self.actions(state)

    def utility(self, state, player):
        """Return the value of this final state to player."""
        raise NotImplementedError

class TicTacToe(Game):
    def __init__(self):
        self.board = [' '] * 9

    def actions(self, state):
        return [i for i, v in enumerate(state) if v == ' ']

    def result(self, state, move):
        new_state = state[:]
        new_state[move] = 'X' if new_state.count('X') == new_state.count('O') ␣
 ↪else 'O'
        return new_state

    def is_terminal(self, state):
        return self.utility(state, 'X') != 0 or self.utility(state, 'O') != 0 ␣
 ↪or ' ' not in state
```

```
    def utility(self, state, player):
        lines = [[0, 1, 2], [3, 4, 5], [6, 7, 8], [0, 3, 6], [1, 4, 7], [2, 5,␣
 ↪8], [0, 4, 8], [2, 4, 6]]
        for line in lines:
            if state[line[0]] == state[line[1]] == state[line[2]] == player:
                return 1
        return 0

class Player:
    def play_game(self, game, strategies: dict, verbose=False):
        """Play a turn-taking game. 'strategies' is a {player name: function}␣
 ↪dict,
        where function(state, game) is used to get the player's move."""
        state = game.board
        while not game.is_terminal(state):
            player = 'X' if state.count('X') == state.count('O') else 'O'
            move = strategies[player](game, state)
            state = game.result(state, move)
            if verbose:
                print('Player', player, 'move:', move)
                print(state)
        return state

# Example random player function
def random_player(game, state):
    return random.choice(game.actions(state))

# Example usage of play_game function
game = TicTacToe()
player = Player()
result = player.play_game(game, {'X': random_player, 'O': random_player},␣
 ↪verbose=True)
```

```
Player X move: 0
['X', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']
Player O move: 2
['X', ' ', 'O', ' ', ' ', ' ', ' ', ' ', ' ']
Player X move: 8
['X', ' ', 'O', ' ', ' ', ' ', ' ', ' ', 'X']
Player O move: 3
['X', ' ', 'O', 'O', ' ', ' ', ' ', ' ', 'X']
Player X move: 6
['X', ' ', 'O', 'O', ' ', ' ', 'X', ' ', 'X']
Player O move: 5
['X', ' ', 'O', 'O', ' ', 'O', 'X', ' ', 'X']
```

```
Player X move: 1
['X', 'X', 'O', 'O', ' ', 'O', 'X', ' ', 'X']
Player O move: 4
['X', 'X', 'O', 'O', 'O', 'O', 'X', ' ', 'X']
```