Conniption AI Evaluation Function

For our implication, we found that there is 69 possible winning lines on the board. Using those lines, we iterate through those positions and check how many pieces the player has between in the winning lines in the board. Next, we first check if the line is a horizontal. If the line is horizonal and a player has 3 pieces in a line, then the player should have higher weight than a line that is vertical or diagonal since a player can use either left or right of the board to place a piece. We found that is it better to weight the opponent score almost double the players weight if they have control a line. This will give the board a lower score if an opponent is close to a winning line. For example, if the AI has 3 pieces in a non-horizontal line and the opponent has one piece, the value of the line will be (3\*128+1\*(-1)) = 384. However, If the line is horizonal, then we only multiply by the player that has control of the line. To score flipping, we used the difference between the two player flips. Then, square the result so it can be a positive number and multiply by 1000. If the opponent has more flips than we subtract from the board score. These will help from choosing a board that flips unnecessary and give a better value if the player has more flips. Last, we add 150 if the board can flip or subtract if the board cannot be flipped.

Our Weights:

AI’s weights = [0, 1, 8, 128, 99999, 99999, 99999, 99999]

Opponent weights = [0, -1, -16, -200, -99999, -99999, -99999, -99999]

Win lines = [[(0, 0), (1, 0), (2, 0), (3, 0)], [(1, 0), (2, 0), (3, 0), (4, 0)], [(2, 0), (3, 0), (4, 0), (5, 0)],

[(3, 0), (4, 0), (5, 0), (6, 0)], [(0, 1), (1, 1), (2, 1), (3, 1)], [(1, 1), (2, 1), (3, 1), (4, 1)],

[(2, 1), (3, 1), (4, 1), (5, 1)], [(3, 1), (4, 1), (5, 1), (6, 1)], [(0, 2), (1, 2), (2, 2), (3, 2)],

[(1, 2), (2, 2), (3, 2), (4, 2)], [(2, 2), (3, 2), (4, 2), (5, 2)], [(3, 2), (4, 2), (5, 2), (6, 2)],

[(0, 3), (1, 3), (2, 3), (3, 3)], [(1, 3), (2, 3), (3, 3), (4, 3)], [(2, 3), (3, 3), (4, 3), (5, 3)],

[(3, 3), (4, 3), (5, 3), (6, 3)], [(0, 4), (1, 4), (2, 4), (3, 4)], [(1, 4), (2, 4), (3, 4), (4, 4)],

[(2, 4), (3, 4), (4, 4), (5, 4)], [(3, 4), (4, 4), (5, 4), (6, 4)], [(0, 5), (1, 5), (2, 5), (3, 5)],

[(1, 5), (2, 5), (3, 5), (4, 5)], [(2, 5), (3, 5), (4, 5), (5, 5)], [(3, 5), (4, 5), (5, 5), (6, 5)],

[(0, 0), (0, 1), (0, 2), (0, 3)], [(0, 1), (0, 2), (0, 3), (0, 4)], [(0, 2), (0, 3), (0, 4), (0, 5)],

[(1, 0), (1, 1), (1, 2), (1, 3)], [(1, 1), (1, 2), (1, 3), (1, 4)], [(1, 2), (1, 3), (1, 4), (1, 5)],

[(2, 0), (2, 1), (2, 2), (2, 3)], [(2, 1), (2, 2), (2, 3), (2, 4)], [(2, 2), (2, 3), (2, 4), (2, 5)],

[(3, 0), (3, 1), (3, 2), (3, 3)], [(3, 1), (3, 2), (3, 3), (3, 4)], [(3, 2), (3, 3), (3, 4), (3, 5)],

[(4, 0), (4, 1), (4, 2), (4, 3)], [(4, 1), (4, 2), (4, 3), (4, 4)], [(4, 2), (4, 3), (4, 4), (4, 5)],

[(5, 0), (5, 1), (5, 2), (5, 3)], [(5, 1), (5, 2), (5, 3), (5, 4)], [(5, 2), (5, 3), (5, 4), (5, 5)],

[(6, 0), (6, 1), (6, 2), (6, 3)], [(6, 1), (6, 2), (6, 3), (6, 4)], [(6, 2), (6, 3), (6, 4), (6, 5)],

[(3, 0), (2, 1), (1, 2), (0, 3)], [(4, 0), (3, 1), (2, 2), (1, 3)], [(3, 1), (2, 2), (1, 3), (0, 4)],

[(5, 0), (4, 1), (3, 2), (2, 3)], [(4, 1), (3, 2), (2, 3), (1, 4)], [(3, 2), (2, 3), (1, 4), (0, 5)],

[(6, 0), (5, 1), (4, 2), (3, 3)], [(5, 1), (4, 2), (3, 3), (2, 4)], [(4, 2), (3, 3), (2, 4), (1, 5)],

[(6, 1), (5, 2), (4, 3), (3, 4)], [(5, 2), (4, 3), (3, 4), (2, 5)], [(6, 2), (5, 3), (4, 4), (3, 5)],

[(3, 0), (4, 1), (5, 2), (6, 3)], [(2, 0), (3, 1), (4, 2), (5, 3)], [(3, 1), (4, 2), (5, 3), (6, 4)],

[(1, 0), (2, 1), (3, 2), (4, 3)], [(2, 1), (3, 2), (4, 3), (5, 4)], [(3, 2), (4, 3), (5, 4), (6, 5)],

[(0, 0), (1, 1), (2, 2), (3, 3)], [(1, 1), (2, 2), (3, 3), (4, 4)], [(2, 2), (3, 3), (4, 4), (5, 5)],

[(0, 1), (1, 2), (2, 3), (3, 4)], [(1, 2), (2, 3), (3, 4), (4, 5)], [(0, 2), (1, 3), (2, 4), (3,5)]]