# Checkers using minimax algorithm

Artificial intelligence
Nikhil Kumar(IIT2018152)
Indian institute of Information technology,Allahabad

**Abstract-**
This report describes the use of the minimax algorithm and pygame library to make player vs compute checkers game.

**Introduction-**
Checkers is strategy board games for two players which involve diagonal moves of uniform game pieces and mandatory captures by jumping over opponent pieces.
In this program, one player is controlled by ai and the other is human-controlled. Ai uses the minimax algorithm to a certain depth to calculate the next best move for it.

**Algorithm:-**

1.A 8X8 board is made using the pygame library of python and pieces of color WHITE and YELLOW are drawn in it.

2.When a piece is selected all the valid moves from that row, col is calculated and then stored in a dictionary with row and col as key, and if any opposite color pieces in between as value, then it is shown on the GUI with a blue circle.
If a piece reaches the opposite end it becomes king and it becomes eligible for all diagonal movement.

3.AI uses a minimax algorithm that finds the best_move with max value if pieces of the same color and best_min with min_value if the piece is of a different color.
It runs to the depth of 3.
Vaues are calculated using heuristic **white_left*1+white_kings*0.9-yellow_left*1-yellow_kings*0.9**

4.For every subsequent move of the player best move is calculated for every piece of ai and the best value among them is chosen as the next move.

5.The player with no remaining pieces loses.

## Pseudocode of MINIMAX

```
function minimax(node, depth, maximising player)
    if depth = 0 or node is a terminal node
        return the utility of the node

    if maximizingPlayer
        bestValue := ??
    for each child of node
        v := minimax(child, depth ? 1, FALSE)
        bestValue := max(bestValue, v)
    return bestValue

    else (* minimizing player *)
        bestValue := +?
        for each child of node
            v := minimax(child, depth ? 1, TRUE)
            bestValue := min(bestValue, v)
        return bestValue
```

.

## Conclusion:

This report concludes the working of a human vs ai checkers game