ASSIGNMENT 2 – OTHELLO BOT in C++

ALGORITHM

AlphaBeta MiniMax algorithm

APPROACH

For a given state of board where the game is to be played, all possible moves are checked and evaluated. The move to be returned will be the best among these.

Evaluation is done by constructing the game MiniMax tree upto maximum depth 5.

For the final level the board is evaluated using a static Heuristic function.

For a given colour and board configuration, this heuristic function computes a score as follows –

score = 10 * piece_difference_heuristic + 801.724 * corner_heuristic + 382.026 * corner_adjacent_heuristic + 78.922 * mobility_heuristic + 74.396 * frontier_heuristic + 10 * disk_heuristic

Here each heuristic parameter is assigned a weight and the corresponding sum is the heuristic score.

Finally before each move, a time constraint is implemented to return the result under 2 seconds.

HEURISTIC PARAMETERS

- **piece_difference_heuristic-** It is the simple percentage of number of coins placed on the board, it is positive if more than half the coins are our colour else it is the negative of the percentage of opponent coins.
- **corner_heuristic-** It is the difference in the number of coins of our colour placed on the corners and the opponent colour multiplied by 25.
- **corner_adjacent_heuristic-** It is the negative of the difference in the number of coins of our colour and the opponent colour placed adjacent to the unoccupied corners of the board multiplied by 12.5.
- **mobility_heuristic-** It is the available mobility percentage from the valid moves to our colour compared with the opponent colour. (Note that this parament will not arise if both sides have 0 moves)
- **frontier_heuristic-** If a square **(i, j)** is occupied then we check its row and column to make new x and y coordinates. The x coordinate **= i + horizontal_bias** and y coordinate **= j + vertical_bias**. If this derived coordinate is under the bounds of the board then if the square **(i, j)** is our colour then we increment **our_frontier_tiles** otherwise **opponent_frontier_tiles** are incremented. The horizontal and vertical bias are arrays that hold arbitrary values determined experimentally. Finally the **frontier_heuristic** is the negative of percentage of **our_frontier_tiles** if more than half of these are our tiles. Otherwise it is the positive percentage of **opponent_frontier_tiles**.
- **disk_heuristic-** It is the difference in the place sums of our colour coins and opponent colour coins on the board. The squares are arbitrarily assigned fixed values in a matrix which indicate if they are favourable.