**Report:**

When we increased the search depth, the agent had to search more deeply as the search space also increased. Nevertheless, the agent was able to analyze the possible moves and gave the best of possible moves. When I increased the depth of the search the execution time of the program also increased gradually. Once the depth was above 10 the time increased drastically.

The evaluation strategy is to get the list of all possible legal moves for the player and apply all the moves to the board and select the best one among the available moves. First row, first column, seventh row and seventh column are important in this game as we can change the whole row/column by having your chip placed in those place. Thus, those places are assigned more weight. And similarly, negative points for second row/column and sixth row and column, because having your chips on those places results in higher chances of opponents having the more weight part. The evaluation function is designed in such a way each and every square on the board is considered worth of and the weight score is calculated by the value held by each player. Thus, giving the first/last rows/columns resulted in having good move ordering.

Alpha beta pruning was used to decrease the number of nodes evaluated by min-max algorithm in the search tree. Since it would completely stop evaluating a move when at least one possibility has been found that proves to be worse that a previously examined moves as it would not give better option, I used Alpha Beta Pruning. It would enable me to skip some sub-trees which we need not evaluate further based in knowledge of bounds of alpha and beta to prune branches. If some legal moves result in better score than beta, then opposite player will avoid this branch and we don’t need to look that sub-tree. And if some legal moves result in a better score than the current best score than replace it with that one.

In Othello, the branching factor of the game tree is bounded by the number of empty squares on the board which goes down as the end approaches. Using alpha beta pruning definitely improved branching factor which is close to the optimal one.