**Lab 2 Report: Hassan Naveed and Abdul Qadeer Rehan**

**Abstract**: The goal is to make an AI that plays chess. The approach used is to have a heuristic function that calculates the likelihood of winning at a state that is not the terminal state. Experimental results show that the AI works really well in finding great moves in the near future.

**Introduction:** Good performance in chess is often viewed as an indicator of intellect. Many show off their talents in chess competitions, however, computers have been able to dominate the game for over a decade. It is worthwhile to develop an AI for this since that gives more competition to humans and they are able to practice on their own with an opponent whos difficulty they can set by themselves. There is the problem of there being too many possible moves at any given time, and each on branching out further. The terminal depth is also very far away, so there are too many possible states to explore each of them. This makes it a good experimental exercise to learn important AI concepts.

**Formulation:** Since it is impossible to look at all the terminal states quickly enough to play in real time, several approaches need to be used. In this problem we use H-minimax with alpha beta pruning.

* **Initial State:** The initial state is a board with pieces arranged in some valid fashion. The king of both players must be on the board, otherwise the termination condition will be reached.
* **Cut-off Test:** A heuristic function is used in H-minimax to evaluate the position of the player with the given sequence of moves. Here the cutoff test is when a depth of 4 has been reached. This translates to 2 moves for each player. (Using a higher depth significantly increased runtime).
* **Transition Model:** One chosen piece is moved to a valid position. All other pieces remain unchanged.
* **Evaluation Function:** The evaluation function here calculates the standing of the player using Renfield values assigned to each piece. A very large number was assigned for a king, since the game terminates without the king. (P = 1, N = 3, B = 3, R = 5,Q = 9, K = 100000). There is a small penalty for depth as well. The evaluation function is:

(sum of players Renfield values) - (sum of opponents Renfield values) – depth\*0.01

* **Exploration policies:** In our implementation, The first exploration policy randomizes the pieces and then the moves. An alternative exploration policy would be to start with moves from pieces of the highest importance first. For example, look at the moves of the king, the queen, then rook, and so on. This is beneficial since more important pieces are likely to give better moves.

**Experiment and Analysis**