Title- Pichu.py

Assumptions-

- This program assumes an initial board that takes a string as input and converts it into a list of lists and uses 8*8 board as the format of the input board.
- The first two rows of the board assumes to have all the white pieces and the last two rows of the board assumes to have all the black pieces.
- All the uppercase letters are white pieces and all the lowercase letters are black pieces.
- w=white , b= black

Abstraction-

- **State space** – The state space is the initial configuration of the board that evolves due to the successor function of a code. It includes all the different boards that can come from all the possible moves of each piece on the board.
- **Initial State-** It is the configuration of the initial board. For example-
  `'R.B..BKRPPP...PP..N........QP.....b.............pppp.ppprnb.k..r'`

- **Successor Function-** The successor function gives all the possible moves for the given initial state of the board along with its heuristic or the evaluation value of that particular board in the form of the tuple based on what the player is . If the player is white (w) , then the successor function will give all the moves possible for the white pieces. If the player is black(b), then the successor function will give all the possible moves for the black pieces.
- **Terminal State-** Terminal State is the state when any one of the following happens-
  1. The depth (d) becomes 0
  2. The king of any of the player is not there in the board.

  If any of the above conditions become true, the Terminal State is true and hence return the evaluation value of that particular state of the board and thus start backing up the values to the top node.

- **Evaluation Function** – The evaluation function returns the evaluation value of the given board based on the following chess heuristic value , where each piece has a defined weight and then we use the following equation to define that- e(board)= 1(no. of white parakeets -number of black parakeets)+3(no. of white Nighthawks – number of black nighthawks)+3(No. of white Bluejays – No of black bluejays)+5(no. of white Robins – no. of black robins)+9(No. of white kingfisher – no of black Kingfisher)+200(No. of white quetzals – No. of black quetzals)
  This evaluation function can be reversed for when the player is black.

- **Approach**- First the initial board is taken as an input and passed as an input board.
  Alpha-beta decision calls the successor function based on the passed player(white or black) . The successors are then passed into the Max_value function which then calls the min_value function recursively. Max function takes the maximum value among all the evaluation values returned by the successors of min .And Min function takes the minimum value among all the evaluation values returned by the successors of max. Max and min are the white and black players according to the given condition.
  Also a while loop has been set up for the depth to determine within the given number of seconds (time), how much depth can be traversed for the tree.
- **Output-** We will get the best board depicting the best move according to the given player, state of the board and time.