**Chess-playing system**

In general, an AI system mainly consists of an *agent* and its *environment*. There are different types of models on which AI works of which today we will use the *PEAS* model. This model is used for grouping similar types of agents together. Unfolding the acronym, ‘P’ stands for a *Performance measure*, ‘E’ for *Environment*, ‘A’ for *Actuators*, and ‘S’ for *Sensors*.

The main objective while playing chess is to *checkmate* the opponent, this occurs when the opponent's king is in check, and there is no legal way to remove it from the attack. Also, it is illegal for a player to make a move that puts or leaves theplayer's own king in check[1]. Therefore, the performance measure of the system is to succeed in accomplishing the above-mentioned winning criteria. The *chessboard*(with figures and a clock)will be the system’s *environment,* after all, the agent will operate here along with the given conditions. The environment is *fully observable* as the agent can see everything in the surrounding via its sensors[2, para.2, p.42]. In chess, the opponent entity *B* is trying to maximize its performance measure, which, by the rules of chess, minimizes agent *A’s* performance measure. Thus, chess is a *competitive multiagent* environment[2, para.1, p.43]. The environment is *deterministic* because the opponent’s next move is fairly known and the agent can try to predict it. We can clearly see the dependencies of the moves, on that account, making our environment *sequential*. If the environment itself does not change with the passage of time but the agent’s performance score does, then we say the environment is semi-dynamic. Thus, Chess, when played with a clock, is *semi-dynamic*[2, para.2, p.44]. The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent. For example, the chess environment has a finite number of distinct states (excluding the clock), hence, Chess also has a *discrete* set of percepts and actions[2, para.3, p.44]. The environment is *known* since the agent has full knowledge of chess rules. In short, the chess(with a clock) environment is *fully observable*, *Multi-agents*, *Deterministic*, *sequential*, *semi-dynamic*, and *discrete*[2, fig2.6, p.45].

Actuators consist of the *screen*- will show the next move to the official for moving the figures, *clock*- will track the time of the players and switch it accordingly, and *printer*- will print the moves made by the machine. Lastly, the sensors of such an intelligent system are the *camera*- AI will keep an eye on the board by tracking the movement of the figures and *clock*- will decide the result(win/lose/draw) by obeying the rules of the game.

The *game tree* is a tree where the nodes are game states and the edges are moves. We use the term search tree for a tree that is superimposed on the full game tree and examines enough nodes to allow a player to determine what move to make. Let us assume our chess game is a tree and to play the game we can use any of the two:- Uninformed search - no sense of the problem domain or Informed search - Uses a heuristic function developed for the domain[3]. For chess there are over 1040 nodes, so the game tree is best thought of as a theoretical construct that we cannot realize in the physical world[2, para.2, p.163].

Let us start with the *Minimax* algorithm - We assume that both *black* and *white* are playing their best moves and we try to maximize the white’s score. We start by performing a DFS(depth-first search, uninformed search) and evaluate the leaf nodes. Choosing the child node with the *highest* value to move if it is the *white* and *lowest* value to move if it is *black*. In a normal chess position, the branching factor is 40. DFS always expands one of the nodes at the deepest level of the tree, and only when a dead end is hit, does the search go back to expand nodes at shallower levels. Advantages: Requires space only in storing the path traversed. For maximum depth ‘m’, requires b\*m nodes to be expanded, as opposed to bd for BFS. There are higher chances that it gets lucky and finds a solution right away. Disadvantages: It still has O(bm ) time complexity. Also, it may get stuck following an unfruitful path, or never find a solution if stuck in a loop. A simple variation can be limiting the depth that we can search (this is Depth-Limited DFS)[4]. It uses *exploration* as it searches the whole tree.

The Best-First Search methods also named “Possibly Best-First Search” by reason of if actually the so-called best move possible at every stage is made, it wouldn’t be searching at all. Instead, it’d just find a direct path to the goal. Most searches are based on heuristics, which makes them informed search techniques as opposed to the previous technique mentioned, which was uninformed. Heuristic *h(n)* = estimated cost of the cheapest path from the state at node ‘n’ to a goal state i.e. a guess as to the “goodness” of a particular state. We need to make a note that a heuristic only applies to a particular state. Unlike DFS or BFS, Best-First Search remembers all previous moves and takes the next move that looks best, according to the heuristic function[5]. Therefore it uses *exploration* as well as *exploitation* since it tries to make the best move.

To Summarizing, Chess has approximately 10120 game paths. These positions comprise the problem search space. Thus, AI problems will have a very large space, too large to search or enumerate exhaustively[6]. Therefore it is hard to comment over victory, if the framework realizes how to play chess relies upon what is implied by the inquiry. In the event that one thinks about realizing how to play as knowing the rules, indeed, the this AI system knows how to play. On the off chance that knowing how to play assumes having a superiority defined by having the option to utilize various methodologies just to win, at that point our system ends up being too feeble to even consider fulfilling this condition.

References -

1. “Chess.” *Wikipedia*, Wikimedia Foundation, 3 Mar.2021, <https://en.wikipedia.org/wiki/Chess#:~:text=The%20object%20of%20the%20game%20is%20to%20checkmate%20the%20opponent,player's%20own%20king%20in%20check>.
2. Russell Stuart, and Norvig Peter. *Artificial Intelligence - A Modern Approach*. Upper Saddle River, Pearson Education, 2003.
3. Dr. Brown. "Week4 lecture slides." Introduction to Artificial Intelligence, Innopolis University. 9 Feb. 2020. Lecture Slides.
4. Djurhuus Rune(*Chess Grandmaster*). “Chess Algorithms: Theory and Practice”. 20 Sept. 2017, Microsoft. Class lecture.
5. Weixiong Zhang, Richard E. Korf. “Depth-First vs. Best-First Search: New Results” 1993, AAAI.
6. “Problem and Search Spaces.”Department of Mathematics and Statiscs, *University of Alaska Achorage*, <http://www.math.uaa.alaska.edu/~afkjm/cs405/handouts/search.pdf>.