**Project Title:** Quantum Chess AI

**Submitted By:** Jayant Kumar(22k-4439), Shyam Sundar(22k-4590) , Shahzaib khan(19k-0273)

**Course:** AI

**Instructor:** Abdullah Yaqoob

**Submission Date:** 10-3-2025

**1. Project Overview**

**Project Topic:**  
Quantum Chess AI is an innovative take on conventional chess, integrating a quantum element where pieces have a probability-based movement system. Additionally, an AI opponent adapts to different difficulty levels using the Minimax algorithm with adjustable depth.

**Objective:**  
The primary objective of this project is to develop an AI capable of playing Quantum Chess at different difficulty levels. The AI will use Minimax for decision-making and incorporate heuristic evaluation strategies to determine optimal moves. The quantum teleportation feature adds an element of unpredictability, increasing game complexity and strategy.

**2. Game Description**

**Original Game Background:**  
Traditional chess is a two-player strategy game played on an 8x8 board. Each player controls 16 pieces, with the objective of checkmating the opponent's king by placing it under an inescapable threat.

**Innovations Introduced:**

* **Quantum Teleportation:** Each piece has a 20% chance of randomly teleporting to a valid position upon movement.
* **Adaptive AI Difficulty:** The AI adjusts its decision-making depth based on the difficulty selected by the player.
* **Interactive UI:** A visually appealing interface allows easy gameplay and difficulty selection.

These innovations introduce randomness and strategic depth, making the game unpredictable while challenging for AI.

**3. AI Approach and Methodology**

**AI Techniques to be Used:**

* **Minimax Algorithm:** The AI evaluates future moves using Minimax to determine the best possible move.
* **Alpha-Beta Pruning:** Used to optimize the Minimax search and improve computational efficiency.
* **Heuristic-Based Evaluation:** AI evaluates board positions based on material advantage, piece positioning, and checkmate threats.

**Heuristic Design:**

* Assign values to pieces (e.g., pawn = 1, knight = 3, queen = 9).
* Evaluate control of central squares.
* Consider king safety and potential checkmate threats.

**Complexity Analysis:**

* Minimax without pruning: O(bd) (where b is branching factor and d is depth)
* Alpha-Beta Pruning improves efficiency, reducing the effective branching factor.
* The introduction of teleportation introduces randomness, making heuristic tuning challenging.

**4. Game Rules and Mechanics**

**Modified Rules:**

* Standard chess rules apply with an additional **Quantum Teleportation** mechanic.
* A moved piece has a 20% chance of teleporting to a random valid position.
* The AI difficulty determines how deep Minimax searches.

**Winning Conditions:**

* A player wins by checkmating the opponent’s king.
* A game can end in a draw by stalemate or insufficient material.

**Turn Sequence:**

* Players move pieces normally.
* If teleportation is triggered, the piece moves to a random valid square.
* AI processes move using Minimax and responds accordingly.

**5. Implementation Plan**

**Programming Language:** Python

**Libraries and Tools:**

* **Pygame** (for GUI and board rendering)
* **NumPy** (for probability-based teleportation mechanics)
* **Stockfish (Optional)** (to test AI efficiency against a strong chess engine)

**Milestones and Timeline:**

* **Week 1-2:** Game design, rule finalization, and interface setup.
* **Week 3-4:** Implement Minimax AI with adjustable depth.
* **Week 5-6:** Implement teleportation mechanics and refine AI heuristics.
* **Week 7:** AI testing and optimization.
* **Week 8:** Final testing, debugging, and report submission.

**6. References**

* "Artificial Intelligence: A Modern Approach" by Stuart Russell & Peter Norvig
* Minimax Algorithm and Chess AI tutorials (GeeksforGeeks, Chess Programming Wiki)
* Pygame documentation for GUI development