**Project Title:** Quantum Chess: A Strategic Board Game with Superposition Mechanics

**Submitted By:**

Arham 21K-3604

Hayyan 21K-3634

Bilal 21K-3551  
**Course:** AI  
**Instructor:** Miss Mehak  
**Submission Date:** 8/3/2025

**1. Project Overview**

**Project Topic:**

Quantum Chess is an innovative board game that introduces quantum mechanics-inspired rules into traditional chess gameplay. Pieces can exist in a superposition of multiple states, meaning a move might not be finalized until observed by an opponent. This introduces an element of uncertainty and strategic depth beyond classical chess.

**Objective:**

The project aims to develop a strategic AI for Quantum Chess using Minimax with modifications for probabilistic decision-making, incorporating Alpha-Beta Pruning to optimize move selection. The AI will evaluate potential superposition states and apply heuristic techniques for strategic positioning.

**2. Game Description**

**Original Game Background:**

Chess is a two-player strategy board game played on an 8x8 grid, where each player controls 16 pieces. The goal is to checkmate the opponent's king by placing it under inescapable threat. Each piece has specific movement rules, and the game requires strategic planning and foresight.

**Innovations Introduced:**

* **Superposition Mechanics:** Pieces can exist in multiple positions simultaneously, resolved only when an opponent attempts to interact with them.
* **Quantum Moves:** Instead of definitive moves, players can make probabilistic moves, introducing uncertainty and deeper strategic planning.
* **Entanglement:** Certain pieces may be linked, meaning moving one affects the other in a unique way.
* **Modified Board Layout:** An experimental 10x10 grid with two additional pieces for added complexity.

These innovations increase the game's strategic complexity by incorporating probability, forcing players to rethink traditional chess heuristics.

**3. AI Approach and Methodology**

**AI Techniques to be Used:**

* **Minimax Algorithm:** Modified for probabilistic decision-making in a multi-state environment.
* **Alpha-Beta Pruning:** To optimize decision trees and reduce computational overhead.
* **Monte Carlo Tree Search (MCTS):** To handle uncertainty introduced by quantum moves.
* **Reinforcement Learning:** (Optional) If self-play training is used to enhance AI performance.

**Heuristic Design:**

* Evaluation function considering material value, positional advantage, and superposition strength.
* Risk assessment for entangled pieces and quantum moves.

**Complexity Analysis:**

* Traditional chess has a branching factor of ~35; Quantum Chess significantly increases this due to probabilistic states.
* Computational challenges include handling superposition collapse and managing entangled piece behaviors efficiently.

**4. Game Rules and Mechanics**

**Modified Rules:**

* Pieces can move in a quantum state, occupying multiple squares until observed.
* Special 'Quantum Capture' rules apply when attacking a piece in superposition.
* The addition of two new pieces with unique movement patterns.

**Winning Conditions:**

* Traditional checkmate rules apply, but additional strategies emerge from quantum mechanics.

**Turn Sequence:**

* Players take turns making either a classical or quantum move, followed by an optional 'Observation Phase' where a piece's state is revealed.

**5. Implementation Plan**

**Programming Language:**

* Python

**Libraries and Tools:**

* **Pygame:** For graphical user interface (GUI).
* **NumPy:** For probabilistic calculations.
* **SciPy:** For advanced mathematical operations.
* **AI Libraries:** TensorFlow/PyTorch (if reinforcement learning is applied).

**Milestones and Timeline:**

* **Week 1-2:** Game mechanics and rule finalization.
* **Week 3-4:** AI strategy development (Minimax, MCTS, and heuristics).
* **Week 5-6:** Coding and testing game mechanics.
* **Week 7:** AI integration and testing.
* **Week 8:** Final testing and report preparation.

**6. References**

* Research papers on AI in board games.
* Quantum Chess (Experimental versions).
* Minimax and Monte Carlo Tree Search implementations.
* Reinforcement learning techniques in strategic gameplay.