**Project Title:** AI-Enhanced Chess with Dynamic Piece Movements  
**Submitted By:** Muhammad Anas (22k-4717), Junaid Masood (22k-4735), Muhammad Hassan(22k-4697)  
**Course:** AI  
**Instructor:** Mehak Mazhar   
**Submission Date:** 08-03-2025

**1. Project Overview**

**Project Topic:**

This project aims to develop an AI-enhanced chess game with **dynamic piece movements** that evolve based on board conditions. The game will introduce **adaptive rules** where certain pieces gain additional movement options under specific circumstances, adding complexity and new strategic depth.

**Objective:**

The primary goal of this project is to **develop an AI-powered chess engine** capable of handling the dynamic rule set using advanced search and decision-making algorithms. The AI will utilize the **Minimax algorithm with Alpha-Beta pruning** and heuristic-based decision-making to evaluate board positions efficiently.

**2. Game Description**

**Original Game Background:**

Chess is a two-player strategy game played on an 8x8 board with six types of pieces: **pawns, knights, bishops, rooks, queens, and kings.** The objective is to checkmate the opponent’s king by putting it in an inescapable threat.

**Innovations Introduced:**

* **Dynamic Piece Movements:** Specific pieces (such as knights and bishops) can **gain additional movement abilities** after a set number of turns or when capturing an opponent's piece.
* **Power-Up Tiles:** Special board tiles provide temporary buffs (e.g., increased movement range, additional attack options).
* **AI-Driven Gameplay:** The AI will **adjust strategies** dynamically based on board conditions.

These modifications will create a **more unpredictable and strategic version of chess**, requiring players (and AI) to think beyond traditional openings and endgames.

**3. AI Approach and Methodology**

**AI Techniques to be Used:**

* **Minimax Algorithm** (Optimized for two-player turn-based strategy)
* **Alpha-Beta Pruning** (To enhance decision-making efficiency)
* **Custom Heuristics** (Evaluating board states beyond material advantage)
* **Machine Learning (Optional Future Extension)** (Using reinforcement learning for AI self-improvement)

**Heuristic Design:**

* **Piece Mobility Score** (Increased for pieces on power-up tiles)
* **Positional Advantage Calculation** (Evaluating board control and future threats)
* **Threat and Defense Analysis** (Weighing offensive and defensive positions)

**Complexity Analysis:**

* The Minimax algorithm operates at **O(b^d)** complexity (where b = branching factor, d = depth), which will be optimized with **Alpha-Beta pruning** to reduce unnecessary computations.

**4. Game Rules and Mechanics**

**Modified Rules:**

* Knights gain the ability to **jump two pieces instead of one** after making three moves.
* Bishops can **move like a queen** for one turn after capturing an opponent’s piece.
* Special tiles provide temporary **bonus moves or defensive boosts**.

**Winning Conditions:**

* Standard chess rules apply: **Checkmate, Stalemate, or Draw** conditions.
* Players can win by eliminating all opponent’s power-up options.

**Turn Sequence:**

* Players take turns as in normal chess, but **adaptability is required** due to dynamic rule changes.

**5. Implementation Plan**

**Programming Language: Python**

**Libraries and Tools:**

* **Pygame** (for GUI implementation)
* **NumPy** (for AI computations)
* **Chess.py** (for standard chess rules implementation)

**Milestones and Timeline:**

* **Week 8:** Define dynamic rule changes and finalize game design.
* **Week 9-10:** Implement board logic and standard chess mechanics.
* **Week 11-12:** Develop AI logic using **Minimax & Alpha-Beta Pruning**.
* **Week 13-14:** Integrate AI with the chessboard and test against different strategies.
* **Week 15:** Final testing, bug fixes, and documentation preparation.

**6. References**

1. **"Artificial Intelligence: A Guide for Thinking Humans" – Melanie Mitchell**
   * **Covers AI techniques, including game-playing strategies.**
2. **"Programming Game AI by Example" – Mat Buckland**
   * **Useful for implementing Minimax, heuristics, and AI-driven game logic.**
3. **"Computer Chess Compendium" – David Levy**
   * **Discusses AI-driven chess engines and strategic decision-making.**
4. **"Algorithms in a Nutshell" – George T. Heineman et al.**
   * **Contains detailed explanations of Minimax and Alpha-Beta pruning.**
5. **Research Paper: "Design and Implementation of Artificially Intelligent Microcontroller based Chess Opponent "**
   * [**https://www.iaeng.org/publication/WCE2010/WCE2010\_pp410-414.pdf**](https://www.iaeng.org/publication/WCE2010/WCE2010_pp410-414.pdf)