**Project Proposal**

**Project Title:** AI Powered Chess Game

**Submitted By:**

* Muhammad Asfand Aamir (22k4720)
* Daniyal Ali Tahir (22k4772)
* Osaid (22k4763)

**Course:** AI  
**Instructor:** Mehak Mazhar  
**Submission Date:** 08/03/2025

**1. Project Overview**

**Project Topic:**

We are developing **Warped Chess**, an innovative variant of traditional chess that introduces unique mechanics, such as warping piece abilities, gravity zones, phantom captures, and time rifts. This enhances the complexity and strategic depth of the game beyond conventional chess.

**Objective:**

The goal of this project is to create a **challenging and dynamic chess variant** that increases strategic diversity and unpredictability. Our focus is on designing and implementing the new mechanics while ensuring balanced gameplay. The game will be developed using **Python (Pygame)**, and an AI opponent will be implemented using **Minimax with Alpha-Beta Pruning** for intelligent decision-making.

**2. Game Description**

**Original Game Background:**

Chess is a two-player strategy game played on an 8x8 board where each player controls 16 pieces. The goal is to checkmate the opponent’s king using standard moves for different pieces: pawns, rooks, knights, bishops, queens, and kings. Each piece has predefined movement patterns, and players take turns moving one piece at a time.

**Innovations Introduced:**

* **Warping Pieces:** Each piece gains an alternate movement every **third turn** (e.g., knights can move diagonally, bishops can move like rooks, etc.).
* **Phantom Captures:** Captured pieces remain on the board as “ghosts” that block movement. They can be revived if a player reaches the opponent’s back rank.
* **King’s Wrath Mode:** If a king is the last remaining piece, it temporarily moves **like a queen for three turns** before resuming normal movement.

These innovations make the game more dynamic, requiring players to adapt their strategies frequently.

**3. AI Approach and Methodology**

**AI Techniques to be Used:**

* **Minimax Algorithm** – To evaluate the best move at each step.
* **Alpha-Beta Pruning** – To optimize decision-making by eliminating unnecessary evaluations.

**Complexity Analysis:**

* Standard chess has a branching factor of ~35. Warped Chess will have a higher branching factor due to additional mechanics, increasing computational complexity.
* Optimized evaluation functions and move-ordering heuristics will be necessary for efficient AI decision-making.

**4. Game Rules and Mechanics**

**Modified Rules:**

* **Warping Abilities:** Every third turn, a piece gains additional movement capabilities.
* **Phantom Captures:** Captured pieces remain on the board and can be revived.
* **King’s Wrath:** A lone king temporarily moves like a queen.

**Winning Conditions:**

* Standard checkmate rules apply.

**Turn Sequence:**

* Players take turns moving one piece at a time.
* Every third turn, a warping effect activates.
* Players can trigger a Time Rift at any point but only twice per game.

**5. Implementation Plan**

**Programming Language:**

* Python

**Libraries and Tools:**

* **Pygame** – For game graphics and user interface.
* **NumPy** – For efficient board state representation.
* **Custom AI Algorithm (Minimax + Alpha-Beta Pruning)** – For intelligent opponent behavior.

**Milestones and Timeline:**

* **Week 1-2:** Game design and rule finalization.
* **Week 3-4:** Basic game mechanics (standard chess movement and GUI).
* **Week 5-6:** Implement warping mechanics and AI strategy.
* **Week 7:** Testing and refining AI decision-making.
* **Week 8:** Final testing, debugging, and documentation.

**6. References**

* [1] Chess programming techniques (<https://www.chessprogramming.org/Main_Page>)
* [2] Minimax and Alpha-Beta Pruning (<https://en.wikipedia.org/wiki/Minimax>)
* [3] Python Pygame Documentation (<https://www.pygame.org/docs/>)