Project Report Template

Project Title: ***Advanced Chess Variant with AI Implementation***
Submitted By: *Ashhal, Omer, Hassan [22K-4306, 22K-4266, 22K-4539]*
Course: AI
Instructor: Ms. Ravia Ejaz/ Ms. Alina Arshad
Submission Date: before *16th May 2025.*

---

1. Executive Summary

- **Project Overview:**
  This project implements a modified chess variant with an advanced AI opponent using the Minimax algorithm with Alpha-Beta pruning. The key innovation is allowing pawns to make double moves from any rank (once per pawn), introducing new strategic elements while maintaining core chess principles. The implementation features a modern GUI using customtkinter and sophisticated AI evaluation techniques.

2. Introduction

- **Background:**
  Chess is a classic strategy game traditionally played between two players. This project modifies conventional chess by introducing a special pawn movement rule while implementing an AI opponent using advanced evaluation techniques.
  Objectives
- *Implement a chess variant with modified pawn movement rules*
- *Develop a strong AI opponent using Minimax and Alpha-Beta pruning*
- *Create a user-friendly GUI interface*
- *Incorporate advanced position evaluation techniques*

3. Game Description

- **Original Game Rules:**
  Standard chess rules apply, with traditional piece movements and objectives.
- **Innovations and Modifications:**
- Pawns can make double moves from any rank (not just the starting rank)
- Each pawn can use this special move only once per game
- Enhanced AI evaluation considering multiple strategic factors

4. AI Approach and Methodology

- **AI Techniques Used:**

1. ***Core Algorithm:***
    - *Minimax with Alpha-Beta pruning*
    - *Search depth of 3 moves*
    - *Transposition table for optimization*
2. ***Position Evaluation:***
    - *Material counting*
    - *Piece positioning (using piece-square tables)*
    - *Pawn structure analysis*
    - *King safety evaluation*
    - *Center control scoring*
    - *Mobility assessment*

- **Algorithm and Heuristic Design:**

ai.py:

```
def evaluate_position(self, board: chess.Board) -> float:
    # Material evaluation
    # Positional evaluation using piece-square tables
    # Mobility scoring
    # Center control evaluation
    # Pawn structure analysis
    # King safety assessment
```

5. Game Mechanics and Rules

- **Modified Game Rules:**

Modified Game Rules

- *Standard chess rules apply*
- *Special pawn double-move rule from any rank*
- *One-time use per pawn for special moves*
- *Traditional checkmate and stalemate conditions*

Turn-based Mechanics

1. *Player (White) moves first*
2. *AI (Black) responds automatically*
3. *Game continues until checkmate/stalemate/draw*

6. Implementation and Development

Development Process:

- Used Python with python-chess library
- Implemented custom board class for variant rules
- Developed sophisticated AI evaluation
- Created modern GUI with customtkinter

Technologies Used

- **Language:** Python 3.11
- **Libraries:**
  - python-chess
  - customtkinter
  - typing

● **Challenges Encountered:**

- *Implementing variant rules while maintaining chess integrity*
- *Optimizing AI performance*
- *Creating responsive GUI*
- *Managing game state effectively*

Team Contributions

- **Team Members and Responsibilities:**
- ***Ashhal:*** *AI algorithm development, evaluation functions, project coordination*
- ***Omer:*** *GUI implementation, testing*
- ***Hassan:*** *Game rules implementation, testing*

Results and Discussion:

AI Performance

- *Successfully implements deep position evaluation*
- *Considers multiple strategic factors*
- *Efficient move selection through Alpha-Beta pruning*
- *Quick response time through transposition table usage*

GUI Features

- *Modern dark theme interface*
- *Clear game status display*
- *Move history tracking*
- *Special notifications for game events*

- *Key components:*

1. *game.py: Core game logic and variant rules*
2. *ai.py: AI implementation with evaluation*
3. *main.py: Game management and control*
4. *chess_gui.py: User interface*

References

Python-chess documentation

Customtkinter documentation

Chess programming concepts and algorithms

Alpha-Beta pruning optimization techniques

Chatgpt, Github