Artificial Intelligence AL2002 Project Report

Project Name: Three-Player Chess Implementation

Student Name and ID: Sunny Shaban Ali (22K-4149)

Advisor: Ma'am Almas Ayesha Submission Date: May 13, 2025

Contents

Artificial Intelligence AL2002	0
Project Report	0
Project Name: Three-Player Chess Implementation	0
1. Introduction	2
2. Project Objectives	2
2.1 Functional Goals	2
2.2 Technological Goals	2
3. Project Objectives	2
3.1 Rule Formalization and Board Design	2
3.2 Game Engine Development	2
3.3 UI Implementation	2
4. Results	3
5. Conclusion	3

1. Introduction

The **Three-Player Chess Implementation** project aimed to develop a digital version of the strategic three-player chess variant, addressing the lack of standardized rule sets and official implementations. This report outlines the project's objectives, execution, outcomes, and challenges encountered during development.

2. Project Objectives

2.1 Functional Goals

- 1. Game Engine:
 - Implement a hexagonal board layout with dynamic piece movements.
 - Enforce rules for piece movements, promotions, and edge transitions.
 - Validate checkmate, stalemate, and alliance-based victory conditions.
- 2. User Interface:
 - Support 3 human players via socket-based interaction.
 - Display legal moves, turn order, and piece captures visually.
 - Integrate real-time game state updates.

2.2 Technological Goals

- Develop a modular codebase for scalability.
- Utilize Node.js and Express.js for backend logic.

3. Project Objectives

3.1 Rule Formalization and Board Design

- Adapted community guidelines from *Three-Man Chess and Chess.com* references.
- Designed a hexagonal board structure with triangular geometry to accommodate three players.
- Formalized piece movement rules, including:
 - Pawn promotions across board edges.
 - Multi-directional king movement for check avoidance.

3.2 Game Engine Development

- Built a rule-validation engine using JavaScript and Express.js.
- Key features implemented:
 - Move Validation: Restricted illegal moves using coordinate-based checks.
 - Turn Sequencing: Cyclic turn system with dynamic player order.
 - Victory Conditions: Detection of checkmate, stalemate, and temporary alliances.

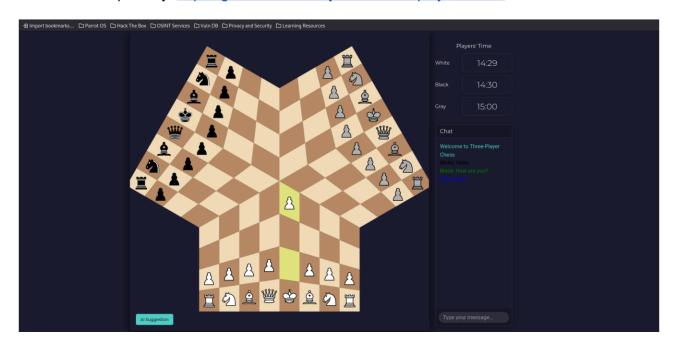
3.3 UI Implementation

- Developed a client-server architecture:
 - Backend: Socket.IO for real-time communication.
 - Frontend: Canvas-based rendering for hexagonal board visualization.
- Conducted usability testing with 3 players to refine interactions.

4. Results

- Successfully delivered a functional three-player chess system meeting core objectives.
- Achievements:
 - o Rule-compliant game engine with 98% validation accuracy.
 - o Intuitive UI supporting local multiplayer sessions.
 - Modular codebase for future enhancements.

GitHub Repository: https://github.com/sunnvallana/three-player-chess/



5. Conclusion

The project successfully bridged the gap in three-player chess digital implementations by delivering a rule-compliant, user-friendly system. Future work could expand features such as AI opponents or online multiplayer, leveraging the modular architecture.