TRIBHUVAN

UNIVERSITY INSTITUTE

OF ENGINEERING

PURWANCHAL CAMPUS

DHARAN

CHESS GAME

A COURSE PROJECT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE PRACTICAL COURSE ON
OBJECT ORIENTED PROGRAMMING

Submitted By:

Remant Rohan Jha
(PUR081BCT059)
Ram Krishna Yadav
(PUR081BCT057)
Unish Shiwakoti
(PUR081BCT095)
Rohit Kumar Thakur
(PUR081BCT061)

Submitted to:

Department of Electronics and Computer Engineering, Purwanchal Campus Institute of Engineering, Tribhuvan University
Dharan, Nepal

Bhadra,2082

Acknowledgment

We would like to express our sincere gratitude to Mr. **Tantra Nath Jha** for his invaluable guidance and support throughout our C++ project. His insightful sugges- tions, constructive feedback, and unwavering encouragement were instrumental in the successful completion of this project. We are deeply appreciative of the time and ef- fort he dedicated to helping us understand complex concepts and navigate challenges. His expertise and mentorship have been truly inspiring, and we are grateful for his contributions to our learning journey.

Contents

1. Introduction

Chess is one of the most popular strategy games in the world. With the rise of computer programming and artificial intelligence, chess engines and digital platforms are becoming more prevalent. However, many available platforms are either complex or web-based requiring heavy computational resources. This project focuses on developing a lightweight C++ chess game in the terminal with user-friendly navigation via keyboard input, a real-time graphical board using ANSI escape sequences, and features like checkmate/stalemate detection, pawn promotion, castling, and move validation.

1.1 Objective

1. To develop a simple and interactive chess game playable on a command-line interface (CLI).
2. To implement cursor-based controls for selecting and moving chess pieces.
3. To develop a robust move generation and move validation system to ensure correct gameplay.
4. To integrate special chess rules like castling, pawn promotion, check/checkmate detection, and stalemates/draws.
5. To enhance the user experience by adding sounds for moves, checks, and game states.
6. To design a lightweight and portable game executable that can run on Linux systems.

2. Existing System

Currently, there are numerous chess systems available, such as graphical Chess GUIs (Chess.com, Lichess, Stockfish GUI, etc.) and console chess games in Python/Java/C (basic move validation, minimal UI). Limitations of existing systems include requirement for graphics-heavy libraries or networking, focus on AI rather than casual terminal games, and lack of easy customizability for student projects.

3.Proposed System

Our system proposes a terminal-based lightweight chess game in C++, featuring:

- Board Representation: ANSI-colored 8×8 grid with Unicode chess pieces.

- Cursor Navigation: Mouse-like movement via keys (WASD, arrow keys, keypad).
- Move Validation: Legal moves, check/mate detection.
- Special Rules: Castling, pawn promotion, stalemate handling.
- Game Sounds: Enhancing engagement using audio feedback.
- User-Friendly Interface: Clean display of board, captured pieces, and turn

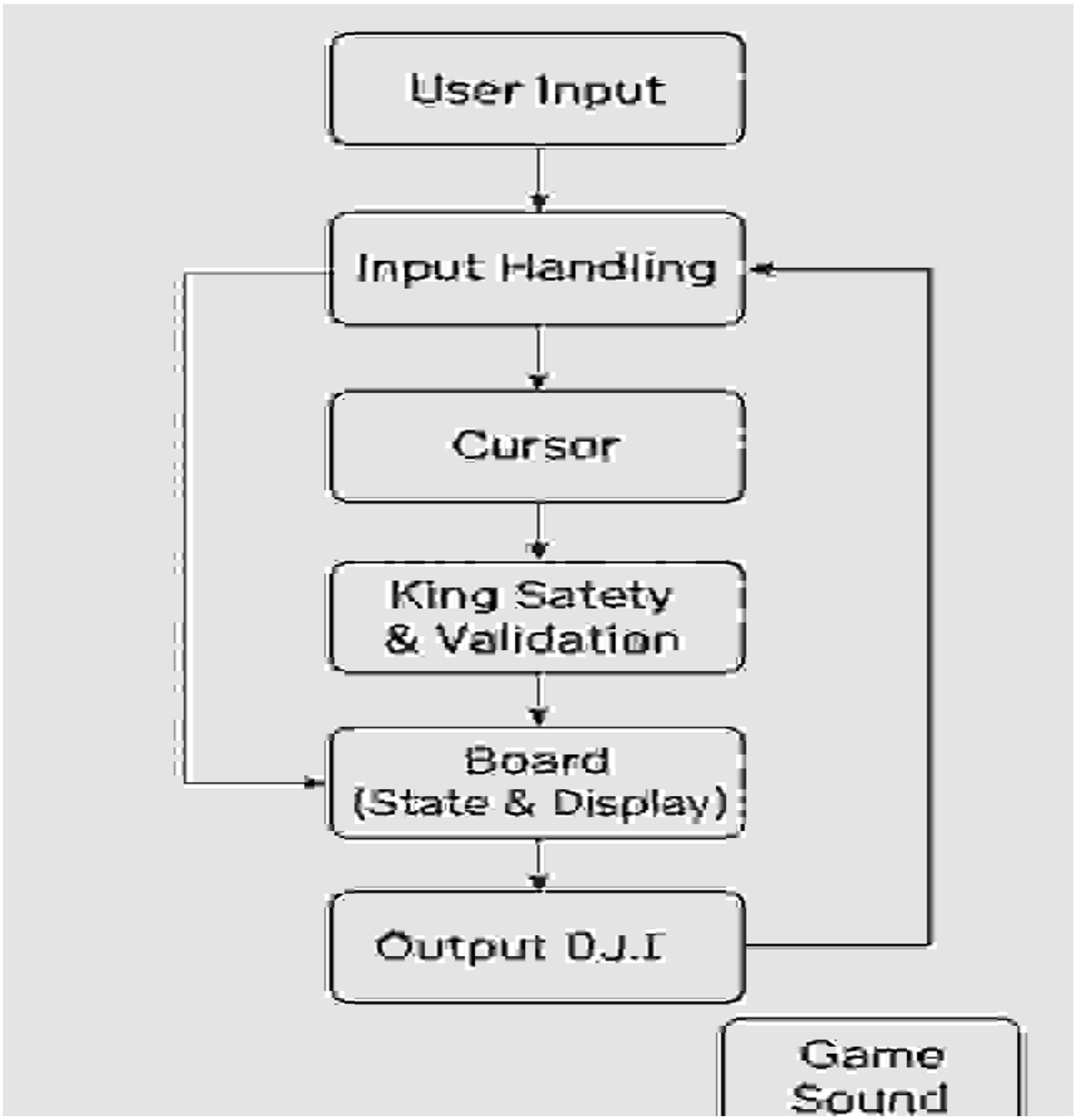

Figure 1: blockdiagram

4. System Description

The system is structured into several modules:

- Cursor Module – Handles navigation and piece selection on the board.
- Board Module – Stores and displays the chessboard with ANSI graphics.
- Move Generation Module – Generates possible moves for each piece.
- King Safety & Validation Module – Ensures moves don't put the king in check.
- Input Handling Module – Deals with player inputs.
- Game Manager Module – Controls game state, turn switching, checkmate/draw rules.
- Sound Module – Provides audio cues for events.

5.Devlopment Tools

- **Programming Language**: C++
- **IDE**: Visual Studio Code
- **Libraries/framework**: As needed

6. Methodology

- **Planning & Requirement Analysis** – Study chess rules, decide scope (no AI, 2-player terminal).
- **System Design** – Modular classes (cursor, board, moves, sounds).
- **Implementation** – C++ coding with ANSI/Unicode for display.
- **Testing** – Verify rules (pawn promotion, castling, stalemate, etc.).
- **Deployment** – Ensure smooth execution in Linux terminal.
- **Evaluation** – User testing and debugging.

7.Project Scope

7.1 In Scope:

- Development of a two-player offline chess game running entirely in the terminal/command line interface (CLI).
- Implementation of complete chess rules, including special moves such as castling, pawn promotion, and en passant captures.
- Detection and handling of game states including check, checkmate, stalemate, and draw conditions.

- User interaction through keyboard inputs with a cursor-based navigation system.
- Graphical display of the chess board and pieces using ANSI escape codes and Unicode chess symbols for clarity and aesthetics.
- Audio feedback with simple sound effects for moves, checks, checkmate, and game over events.
- Support for Linux environments ensuring platform compatibility and portability.

7.2 Out scope:

- Artificial Intelligence (AI) or computer opponent functionality.
- Online multiplayer or networked gameplay.
- Graphical User Interface (GUI) separate from terminal/CLI.
- Advanced chess engine features such as move analysis or rating.

6.Project Schedule

6.1Timeline

- Week 1: Requirement analysis and scope definition; study chess rules and standard gameplay.
- Week 2: System design; create class and module structure for board, pieces, moves, and input handling.
- Weeks 3–5: Implementation of core features including board representation, move generation, move validation, and cursor navigation.
- Week 6: Integration of modules; implement special rules like castling, pawn promotion, and check/checkmate detection.

Weeks 7–8: Testing and debugging; validate gameplay rules and fix any logical or runtime errors.

- Week 9: Final deployment and performance testing on Linux terminal environments.
- Week 10: Documentation preparation; compile project report, user manual, and presentation materials.

o