Chess Game Visualized in C++

Muhammad Raza Mustafa
Abdullah Razzaq
Muhib Ahmed Khan

Introduction

Background

This project focuses on the development of a basic chess game using C++. Chess is a well-known two-player strategy game, and its implementation provides an excellent opportunity to apply Object-Oriented Programming (OOP) concepts such as classes, encapsulation, and modular design. The main focus is on creating a functional and maintainable chess engine with clean code structure.

Problem Statement

Implementing a chess game requires organizing complex rules and interactions. The challenge lies in structuring the game logic in a clear and manageable way using OOP, allowing features to be added or modified easily.

Objectives

- To develop a simple and functional chess game in C++
- To apply core OOP concepts through the use of classes and objects
- To build a modular structure that could be extended in the future
- To implement a graphical user interface (GUI) using SFML

Scope of the Project

Inclusions

- Core chess rules (movement, turns, basic captures)
- C++ classes (e.g., Piece, Board, Game)

- Command-line and graphical user interface using SFML
- Clean and modular codebase

Exclusions

- Advanced chess rules (e.g., castling, en passant, checkmate detection)
- Artificial intelligence for computer play (initially)
- Web server or web-based interface (initial phase)

Project Description

Overview

The project involves designing a basic chess game using C++, showcasing the power of OOP. Each chess component will be represented by a class, and the game will initially run in a text-based interface, followed by GUI development using SFML. Future expansions include integration of a chess-playing AI bot and developing a web interface to interact with it.

Technical Requirements

- C++17 or later
- Microsoft Visual Studio or Visual Studio Code
- SFML library for GUI

Project Phases

1. Research: Understand chess rules and C++ class design
2. Development: Code the core components (board, pieces, moves)
3. GUI: Implement graphical interface using SFML
4. Testing: Verify correctness of moves and game state transitions
5. Future Plans:
    - Integrate an AI bot using our own chess engine
    - Create a web server to support remote access and play

Methodology

Approach

The project will be developed in incremental stages, starting with the core game loop and gradually adding features. The focus will remain on correctness and code clarity. GUI implementation using SFML will follow the text-based version. Future plans include AI integration and a supporting web server.

Team Responsibilities

- Abdullah Razzaq: Class design and game logic
- Muhammad Raza Mustafa: Input/output handling and move processing
- Muhib Ahmed Khan: GUI development, finalizing game, and documentation

Expected Outcomes

Deliverables

- Fully functional chess game with command-line and GUI (SFML) interface
- Well-documented C++ code using OOP design
- Project report and brief usage instructions
- Future scope of AI bot and web server integration

Relevance

The project demonstrates practical application of object-oriented programming in C++, and it aligns with ICT topics such as software design, data handling, and algorithm implementation. GUI implementation showcases C++ graphics handling with SFML, and future integration of AI and web technologies connects the project with current trends in game development.

Resources Needed

Software

- Visual Studio / Visual Studio Code
- C++ compiler (g++, make)
- SFML graphics library
- Git (for version control)

Other Resources

- Online tutorials and chess programming references
- Instructor support and feedback