

# Chess Game Visualized in C++ Using SFML

Muhammad Raza Mustafa

Abdullah Razzaq

Muhib Ahmed Khan

## 1 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 integrate a basic graphical user interface using SFML.

## 2 Scope of the Project

### Inclusions

- Core chess rules (movement, turns, basic captures).

- C++ classes (e.g., `Piece`, `Board`, `Game`).
- Graphical User Interface (GUI) using SFML.
- Clean and modular codebase.

## Exclusions

- Advanced chess rules (e.g., castling, en passant, checkmate detection).
- Artificial intelligence for computer play.
- Web-based interface.

## 3 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. The game will use SFML to create an interactive graphical interface, replacing the initial plan for a text-based or web-based interface.

### Technical Requirements

- C++17 or later.
- Microsoft Visual Studio or Visual Studio Code.
- SFML 2.5+ for GUI development.

### Project Phases

1. Research: Understand chess rules and C++ class design.
2. Development: Code the core components (board, pieces, moves).
3. Testing: Verify correctness of moves and game state transitions.
4. GUI Implementation: Develop a basic 2D GUI using SFML.

## 4 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. Once the logic is validated via the console, the SFML-based GUI will be implemented to visualize the board, pieces, and player moves.

### Team Responsibilities

- Abdullah Razzaq: Class design and game logic.
- Muhammad Raza Mustafa: Input/output handling and move processing.
- Muhib Ahmed Khan: Finalizing of game and documentation.

## 5 Expected Outcomes

### Deliverables

- Fully functional GUI-based chess game using SFML.
- Well-documented C++ code using OOP design.
- Project report and usage instructions.

### 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. The use of SFML integrates modern graphics programming concepts.

## 6 Resources Needed

### Software

- Visual Studio / Visual Studio Code.
- C++ compiler (e.g., g++, make).
- Git (for version control).

## **Libraries and Tools**

- SFML (Simple and Fast Multimedia Library).
- Online tutorials and chess programming references.
- Instructor support and feedback.