National University of Computer and Emerging Sciences
(FAST-NUCES)
PROJECT PROPOSAL

1v1 Chess using SFML

Instructor: Talha Shahid

Group Members:

- Muhammad Adil Saeed – 24K-0705
- Zaid Faraz – 24K-0636
- Muhammad Haseem Samo – 24K-0666

1. Introduction

Background:

Chess is a strategic board game that has stood the test of time. Developing a digital version of chess using **Object-Oriented Programming (OOP)** in **C++** helps students understand concepts like inheritance, encapsulation, and polymorphism. By integrating **SFML** for multimedia, the game also explores modern tools for graphics, audio, and timing.

Problem Statement:

The project tackles the challenge of building a modular and reusable codebase for a chess game using OOP. It also involves integrating multimedia components and file management, offering a hands-on way to apply several ICT and programming principles.

Objectives:

- Demonstrate practical application of OOP in a game development context.
- Use SFML to create a rich user experience with visuals and sounds.
- Implement per-turn timers with alerts.
- Enable saving of user data using file handling.

---

2. Scope of the Project

Inclusions:

- Full two-player chess game with standard rules
- Real-time timers
- Player data
- Normal chess rules with time
- Includes castling, en passant, pawn promotion etc
- Undo Moves: Allow players to take back their moves (uncertain)

Exclusions:

- Networked or online multiplayer support
- AI-based opponent
- Advanced chess features like move suggestions or analysis

---

3. Project Description

Overview:

This C++ project models a playable chess game using OOP principles. Each component (e.g., `Piece`, `Board`, `Player`) will be encapsulated into its own class. The use of **SFML** allows for an engaging interface and responsive user experience. Sound and time-based alerts enhance gameplay, and users can access their records using the file system.

Technical Requirements:

- **Programming Language:** C++
- **Graphics & Timing Library:** SFML
- **IDE:** Visual Studio Code
- **File Handling:** Standard C++ I/O streams

Project Phases:

1. **Research & Planning:** Learn chess logic and SFML basics.
2. **Design:** Draw up class diagrams and define responsibilities.
3. **Implementation:**
    - Build core game logic and structure.
    - Integrate visuals, audio, and file handling.
4. **Testing & Debugging:** Ensure all features work correctly and debug as needed.

---

4. Methodology

Approach:

The team will use a modular and iterative approach. Each member is responsible for their own area, with Zaid Faraz coordinating efforts and integrating all parts into the final product.

Team Responsibilities:

- **Muhammad Adil Saeed:** Timer logic using SFML
- **Haseem Samo:** File handling for saving user data and responsible for
- report and proposal
- **Zaid Faraz:** Project leader; overseeing full game logic, class structure, and final integration

---

5. Expected Outcomes

Deliverables:

- Functional chess game executable
- Game code demonstrating OOP principles
- User interface with timing and sound alerts
- Save individual game record/history
- Short report with screenshots and user guide

Relevance to ICT Topics:

- Demonstrates object-oriented design and class structuring
- Explores multimedia integration in software applications

---

6. Resources Needed

Software:

- Visual Studio Code / Visual Studio
- SFML (Simple and Fast Multimedia Library)
- C++ Compiler (e.g., MinGW)

Other Resources:

- C++ Documentation - cppreference.com
- GeeksforGeeks - OOP and Chess Implementation
- SFML Documentation - Audio, Time, and Graphics
- YouTube - Hopzbie