DSA Final Project Proposal

Session 2023-2027

Submitted By:

Mohammad Bilal   2023-CS-168

Supervised By:

Mr. Nazeef Ul Haq

Mr. Waseem

Department of Computer Science

University of Engineering and Technology, Lahore

1 Project Overview

The goal of this project is to design and implement a fully functional multiplayer chess game in Python using Pyglet/PyGame Library. The chess game will feature a GUI-based interface, a rules engine for move validation, game analysis, and maybe an artificial intelligence (AI) opponent. The project will utilize fundamental data structures and algorithms to efficiently manage the chessboard, pieces, game rules, and AI decision-making.

2 Objectives

- Implement a chess game with standard rules, including move validation, check, checkmate, and stalemate detection.
- Develop a data structure to represent the chessboard and manage piece movements.
- Optimize performance with efficient data structures and algorithms.

3 Features and Scope

1. **Game Features**
    - Pick/Drop functionality for piece movement.
    - Real-time move validation.
    - Detection of check, checkmate, and stalemate conditions.
2. **Data Visualization**
    - Graphical representation of the chessboard.
    - Attack graph visualization for learning and analysis.

4 Data Structures

The project will utilize the following data structures:

- **2D Array**
    - Represent the chessboard as an 8x8 grid.
    - Store pieces and their positions.
- **Linked List**
    - Track captured pieces for each player.
    - Manage history of moves for undo functionality.
- **Stack**
    - Store game states for undo and redo functionality.

- **Graph**
  - Represent possible moves for each piece.
  - Use graph traversal to detect check and checkmate conditions.
- **Tree**
  - Model the game state as a tree for AI decision-making.
  - Use Minimax and Alpha-Beta pruning to explore the game tree (Optional).

5 Algorithms

The project will incorporate the following algorithms:

- **Breadth-First Search (BFS)**:
  - Compute the shortest path for pieces like knights.
- **Minimax Algorithm (Optional)**
  - Evaluate all possible moves and their outcomes for AI decision-making.
- Cycle Detection in Graphs:
  - Detect check and checkmate conditions by finding cycles in attack graphs.

6 Expected Outcomes

- A fully functional chess game with rule validation and AI functionality.
- Efficient management of the game state using appropriate data structures.

7 Conclusion

This project will demonstrate the practical application of data structures and algorithms in solving real-world problems. By integrating arrays, linked lists, stacks, graphs, and trees, along with advanced algorithms like BFS, the chess game will showcase computational thinking and problem-solving skills.