Project Report

Project Title: AI-Based Quoridor Game
Submitted By:

- Waniya Badar 22k-4526
- Alishba Hassan 22k-4333
- Nimil Zubair 22k-4617

Course: AI
Instructor: Muhammad Shafique Rehman
Submission Date: 11th May, 2025

---

1. Executive Summary

- **Project Overview:**
  This project aimed to develop an enhanced version of the classical Quoridor board game, integrating Artificial Intelligence to create a challenging and interactive experience for players. The main objective was to implement a playable game that supports both single-player (against AI) and multiplayer modes, with options for 2 and 4 participants. The original Quoridor game mechanics were retained but extended to include dynamic AI behavior, GUI enhancements, and scalability. The AI strategies implemented include Breadth-First Search (BFS) for easy difficulty and a combination of BFS and A* (A-Star) search algorithm for hard difficulty, enabling the AI to make strategic moves and wall placements based on real-time game states.

2. Introduction

- **Background:**
  Quoridor is a strategic board game where players aim to reach the opposite side of the board while obstructing opponents using walls. Traditionally designed for 2 players, the game poses unique challenges in AI development due to its dual aspects of movement and obstruction. This project was selected to explore the application of AI techniques in a game that requires both offensive and defensive strategies. The innovation introduced in this project includes support for up to 4 players, difficulty-based AI, and a refined user interface. The project also features enhancements such as animated movement, sound integration, and refined wall placement logic.

- **Objectives of the Project:**

  The primary objectives of this project were to design and implement a complete, fully functional version of the Quoridor game, create AI opponents that can play competitively using pathfinding algorithms, test the AI's performance against human players, and ensure the game is both challenging and user-friendly. The goal was to make the game more accessible and intelligent while providing players with a realistic challenge across different difficulty levels.

3. Game Description

- **Original Game Rules:**

  In the original Quoridor game, each player controls one pawn and begins on one side of a 9x9 grid. The goal is to reach the opposite side before any other player. On their turn, a player can either move their pawn to an adjacent square or place a wall to block their opponents. Walls are placed between squares and cannot completely block a player's access to their goal. The game continues until one player reaches the target row.

- **Innovations and Modifications:**

  In this modified version, several new features and enhancements were introduced. The game was redesigned to support 2 and 4 players, either human or AI-controlled. The AI logic varies based on difficulty: a simpler BFS approach is used for easy AI, while hard AI employs a more strategic blend of BFS and the A* algorithm. Additionally, a graphical interface was built to allow drag-and-drop wall placement and smooth pawn movements. Sound effects were also included to enrich the gameplay experience.

4. AI Approach and Methodology

- **AI Techniques Used:**

  The AI component of the game was developed using two primary techniques: Breadth-First Search (BFS) and A* Search. For easy difficulty, the AI relies on BFS to determine the shortest path to its goal row, ensuring that it always advances efficiently. In hard mode, the AI uses a combination of BFS and the A* algorithm, incorporating heuristics such as Manhattan Distance to evaluate possible moves more strategically. This allows the AI not only to progress toward its goal but also to hinder opponents by placing walls in optimal positions.

- **Algorithm and Heuristic Design:**

  The A* algorithm incorporated heuristics such as Manhattan Distance to estimate the distance to the goal. The AI's decision-making process involves calculating all possible legal moves and evaluating them based on which move optimally advances the player while potentially obstructing others. The wall placement logic is designed

to analyze the opponent's shortest path and identify positions where a wall would have the maximum disruptive effect, without violating the rule that all players must have a valid path to their goals.

- **AI Performance Evaluation:**

  The AI's performance was evaluated by playing multiple matches against human players. Metrics such as win rate, move decision time, and effectiveness in blocking opponents were tracked. The easy mode AI demonstrated competent pathfinding but lacked strategic wall placement. In contrast, the hard mode AI, leveraging the A* algorithm, showed significant improvements in its ability to win, achieving a win rate of around 65%.

5. Game Mechanics and Rules

- **Modified Game Rules:**
  While retaining the fundamental Quoridor rules, we added multi-player support (up to 4 players), AI difficulty levels (easy-BFS and hard- BFS + A*), and validation mechanisms to ensure fairness and legal wall placements. Walls cannot completely block a path to the goal, and every move undergoes legality checks before execution.

- **Turn-based Mechanics:**
  Gameplay follows a turn-based system. Each player, on their turn, can either move their pawn or place a wall. The turn then shifts to the next player. The interface displays visual cues to indicate whose turn it is and enforces sequential execution of moves.

- **Winning Conditions:**
  The first player to reach their designated goal row is declared the winner. In single-player mode against AI, the user plays until they win or the AI reaches its target. In multiplayer games, the match ends immediately upon a player's victory, preserving the competitive nature of the game.

6. Implementation and Development

- **Development Process:**
  The development process began with setting up the core game logic and rule engine, followed by implementation of the AI using BFS and A* algorithms. Once the AI behavior was validated in simulations, the user interface was developed using Python's GUI libraries. This included handling drag-and-drop wall placements, pawn movements, and animations. Sound integration was added for move confirmation and game events, enhancing the overall user experience.

- **Programming Languages and Tools:**
    - Programming Language: Python
    - Libraries: Tkinter/Pygame, heap, math, random
    - Tools: GitHub

- **Challenges Encountered:**
  Several challenges were faced, particularly in wall placement logic, where ensuring that no player was completely blocked required the use of graph traversal validation. Balancing AI performance and response time also proved complex, especially when scaling from two and four players. These were resolved through extensive testing, recursive validation functions, and heuristic optimization in A*.

7. Team Contributions

- **Team Members and Responsibilities:**
    - **Waniya Badar:** Developed the AI algorithms, implementing both BFS and the A* heuristic.
    - **Alishba Hassan:** Modified and enforced the game rules, ensuring all legal constraints were met.
    - **Nimil Zubair:** Conducted thorough testing, measuring AI performance and resolving bugs related to pathfinding and multi-player turns.

8. Results and Discussion

- **AI Performance:**
  The final product successfully met all project objectives. The AI was able to make informed decisions and offered a challenging experience, particularly in hard mode. Against human players, the easy AI won about 40% of the time, primarily due to its limited wall strategy, whereas the hard AI achieved a 65% win rate due to its more advanced heuristics.

9. References

1. Original Python Codebase Developed by Team
2. Amit's A* Pages – Understanding Pathfinding: http://theory.stanford.edu/~amitp/GameProgramming/
3. Python Documentation: https://docs.python.org/3/
4. Tkinter GUI Reference: https://docs.python.org/3/library/tkinter.html