Project Report

Project Title:
AI-Based Quoridor Game Using Minimax and Alpha-Beta Pruning

Submitted By: Maisum Abbas
Course: AI
Instructor: Ms Ramsha Jatt
Submission Date: 16 May, 2025

1. Executive Summary

This project focuses on developing a playable version of the Quoridor board game enhanced with an AI opponent. The AI uses the Minimax algorithm with Alpha-Beta pruning to evaluate potential moves and wall placements. The goal is to create an intelligent and competitive opponent that can challenge human players in strategic decision-making.

2. Introduction

- **Background:**
  - Quoridor is a 2-player strategy game where each player starts at opposite ends of a 9×9 grid. Players move one tile per turn or place walls to hinder the opponent. The first to reach the opposite side wins. It was chosen for its simple rules but deep strategy potential, making it ideal for AI implementation.
- **Objectives:**
  - Build a functional digital version of Quoridor.
  - Integrate an AI using Minimax with Alpha-Beta pruning.
  - Evaluate AI performance and behavior against human moves.

3. Game Description

- **Original Rules:**
  - Players alternate turns to either move their pawn one step or place a wall. Walls cannot fully block a path to the goal.
- **Innovations:**
  - Built with Pygame for UI and interactions.
  - AI uses shortest-path analysis and wall placement strategies.
  - Enhanced visuals, start screen, and win popup.

4. AI Approach and Methodology

- **Techniques Used:**
    - The Minimax algorithm with Alpha-Beta pruning was used to simulate future moves and select the most advantageous action.
- **Heuristics:**
    - Evaluates states using path length difference between player and AI.
    - Considers remaining wall count to influence strategic advantage.
- **Performance:**
    - AI makes decisions in under 1 second. It effectively blocks paths and finds optimal routes, providing challenging gameplay.

5. Game Mechanics and Rules

- **Game Rules:**
    - 9×9 board grid.
    - Each player starts with 10 walls.
    - Walls cannot fully trap players.
- **Turn-based:**
    - Move or place a wall.
- **Winning Condition:**
    - First to reach the opposite side wins.

6. Implementation and Development

- **Process:**
    - Designed UI and gameplay logic.
    - Implemented Minimax with pruning.
    - Developed wall validation and shortest path logic.
- **Tools Used:**
    - Language: Python
    - Libraries: Pygame, collections
    - Tools: VS Code, Github
- **Challenges:**
    - Ensuring wall placement doesn’t block all paths.
    - Balancing AI decision quality with execution speed.
    - Integrating user interface with gameplay.

7. Team Contributions

[Maisum Abbas]: Creating the game from scratch and BFS

[Irteza]: Minmax with pruning and UI Design

8. Results and Discussion

The AI plays intelligently, using both offensive and defensive strategies. In testing, the AI won a majority of games against casual players. Its decision-making is fast and adapts well to different board scenarios, demonstrating successful implementation of strategic reasoning.

9. References

- Pygame Documentation. (n.d.). *Pygame*. Retrieved from https://www.pygame.org/docs/
- Wikipedia contributors. (n.d.). *Quoridor*. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Quoridor
- GeeksforGeeks. (n.d.). *Minimax Algorithm in Game Theory*. Retrieved from https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/