

Project Report

Project Title:

AI-Powered Quoridor with Modified Rules and Alpha-Beta Search

Submitted By:

- Muhammad Umer Ahmed Abbasi (22K-4599) – Group Leader
- Jawwad Ahmed (22K-4648)
- Muhammad Talha Asim (22K-4589)

Course: AI

Section: BCS-6C

Instructor: Sir Abdullah Yaqoob

Submission Date: 11 May 2025

1. Executive Summary

This project aimed to develop an intelligent 4-player version of the board game Quoridor using AI. The original 2-player Quoridor was modified with a 13x13 board, 4-player support, and new interactive mechanics such as wall placement previews, AI move visualization, and dynamic GUI. We integrated the Minimax algorithm with Alpha-Beta Pruning to build decision-making agents that intelligently navigate the board and place walls to delay opponents strategically.

2. Introduction

Background

Quoridor is a pathfinding strategy game where each player must reach the opposite side of a grid while placing walls to hinder opponents. The original game is for 2 players on a 9x9 grid. We selected Quoridor due to its rich strategic possibilities and need for search-based AI algorithms, making it an ideal project for exploring AI planning in a competitive environment.

Objectives of the Project

- Extend Quoridor to support 4 players
- Implement Alpha-Beta based AI that can play competitively
- Provide a GUI for human interaction
- Visualize AI decisions and wall logic

- Compare human vs. AI performance

3. Game Description

Original Game Rules

- 2 players compete on a 9x9 grid
- Each player moves one space per turn (forward, backward, left, right)
- Walls can be placed to block paths (limited number of walls for each player)
- Players cannot be fully trapped
- First to reach opposite side wins

How to play

Game Objective

Each player must move their pawn from their starting position to the opposite side of the board.

- Player 1: Start bottom center → Goal top row
- Player 2: Start top center → Goal bottom row
- Player 3: Start left center → Goal rightmost column
- Player 4: Start right center → Goal leftmost column

The first player to reach their goal side wins.

Turn Structure

On a player's turn, they must choose one of the following actions:

1. Move the Pawn

- Pawns move one square in any of the 4 directions (up, down, left, right).
- Movement is blocked by walls.
- You cannot move diagonally unless jumping.

2. Jumping Over a Player

- If an opponent is directly adjacent, and the cell beyond them is unblocked and empty, you may jump over.
- If the square behind the opponent is blocked by a wall or pawn, you may move diagonally left/right around the opponent.

In 4-player mode, diagonal moves and jump-over logic must handle multiple opponents correctly.

3. Place a Wall

- You can place a wall to block or delay opponents.
- A wall:
 - Is either vertical (|) or horizontal (—)
 - Must align along the grid lines (not inside a cell)
 - Blocks two adjacent paths
- Walls cannot completely block a player's only path to their goal.
- Walls cannot overlap existing walls.

Wall Placement Rules

- Use keyboard:
 - Press H for horizontal wall mode
 - Press V for vertical wall mode
- Ghost walls show preview of placement.
- Click to place a wall along the boundary between cells.
- Walls are visually brown and block 2 cell connections.
- You cannot place a wall:
 - That crosses another wall improperly
 - If you have no walls left
 - That blocks all paths to goal (checked using BFS)

Winning the Game

- First player to reach the opposite side wins.
- Movement into the final goal row/column completes the game.
- If no legal moves are possible (rare), the player closest to their goal wins (fallback rule).

Innovations and Modifications

- Expanded to support 4 players
- Grid size increased to 13x13
- Wall previews and ghost mode for placement
- Wall counters, timers, and turn indicators
- Enhanced wall validation to ensure fair paths
- “My turn” label and hover-based visual cues
- Wall placement logic improved for tight intersections

4. AI Approach and Methodology

AI Techniques Used

- **Minimax Algorithm** with **Alpha-Beta Pruning** for optimal decision-making
- **Breadth-First Search (BFS)** used for pathfinding and shortest path evaluation

Algorithm and Heuristic Design

- The evaluation function minimizes the player’s shortest path to their goal while maximizing distance for opponents.
- AI evaluates potential wall placements and movements using simulated game states.
- Wall decisions are made probabilistically to prevent overly aggressive blocking.

AI Performance Evaluation

- AI decisions complete within 1–2 seconds
- Strategic wall placement occurs ~30% of the time
- AI shows 60–70% win rate vs. human players at depth=1

5. Game Mechanics and Rules

Modified Game Rules

- 13x13 grid for expanded strategy
- Up to 4 players with wall limits adjusted (10 walls each)
- Wall placements are validated not to block all paths

Turn-based Mechanics

- Each player moves or places a wall per turn
- Ghost preview shown before wall is placed
- Human players use keyboard ('H', 'V') + click to place walls

Winning Conditions

- First to reach their opposite edge wins
- If no moves are possible, the closest to their goal wins

6. Implementation and Development

Development Process

- Implemented Quoridor rules and movement logic
- Created quoridor_game.py for AI, rules, validation, alpha-beta pruning with minimax algo, BFS
- Designed GUI with Tkinter (quoridor_gui.py)
- Integrated Alpha-Beta AI decision engine
- Added visual enhancements (timers, ghost walls, “My Turn”, counters)

Programming Languages and Tools

- **Python**
- **Tkinter** (GUI)
- **BFS** logic built from scratch for movement

Challenges Encountered

- Handling wall placement without path blockage
- Resolving diagonal jump logic and wall crossovers
- GUI alignment for wall preview rendering
- Performance optimization of AI decision time

7. Team Contributions

- **Muhammad Umer Ahmed Abbasi** (22K-4599):
AI implementation, game logic, pathfinding algorithms
- **Jawwad Ahmed** (22K-4648):
Game rule modifications, player interactions, wall validation
- **Muhammad Talha Asim** (22K-4589):
GUI development, event handling, ghost walls, visual effects

8. Results and Discussion

AI Performance

- AI wins 65% of games in single-player mode
- Wall usage varies based on board state and proximity to players
- Alpha-Beta pruning significantly improved decision time

Summary

The AI adapts to opponents' strategies, places walls to stall progress, and balances movement vs. interference intelligently. The game achieves a playable and competitive experience for 1–4 players with smooth GUI interaction.

9. References

- Quoridor Official Rules: <https://boardgamegeek.com/boardgame/624/quoridor>
- AI in Games – Russell & Norvig, “Artificial Intelligence: A Modern Approach”
- BFS/Minimax Tutorials: GeeksForGeeks, Stack Overflow
- Python Tkinter Documentation
- Original Quoridor Python Implementations and GitHub Projects