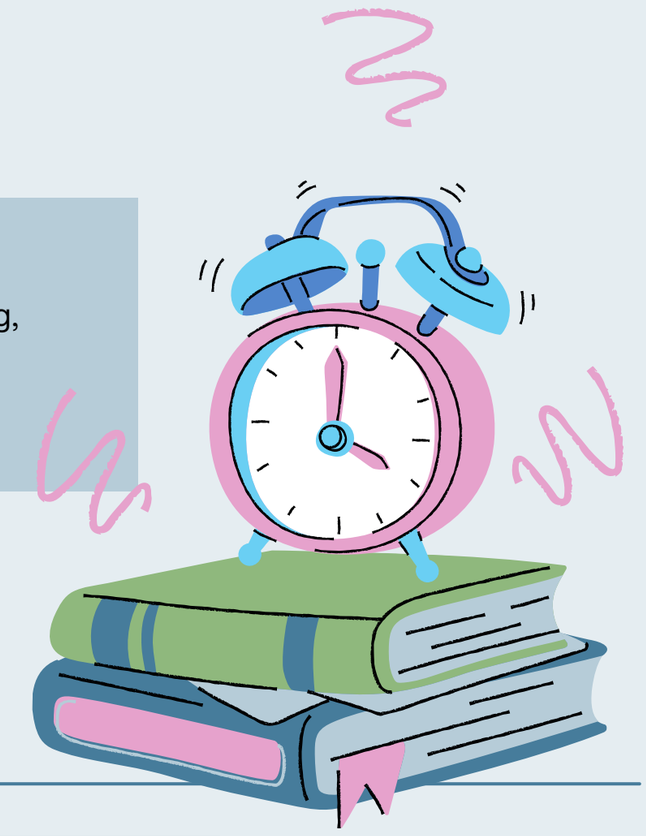
PLAYING NINE MEN'S MORRIS: AN AI AGENT USING ADVERSARIAL SEARCH

Nine Men's Morris (Mill) is a classic two-player strategy game divided into three phases: placing, moving, and flying. The goal is to form mills three aligned pieces to capture the opponent's pieces. This project explores the design of an intelligent AI agent that plays Nine Men's Morris using adversarial search.



Author
Dushime Mudahera Richard
Masters in Data Science
Faculty of Mathematics, Natural Sciences, and Information Technologies
University of Primorska (Famnit)

INTRODUCTION

- Nine Men's Morris: classic two-player strategy board game
- Three phases: placing, moving, flying
- Objective: Create 'mills' (three in-a-row) to capture opponent pieces

AIM

- Develop an AI agent using adversarial search
- Implement multiple difficulty levels
- Evaluate performance and playability

METHODOLOGY

Project Objectives
- Design intelligent game agent
- Ensure real-time playability
- Analyze strategic strength

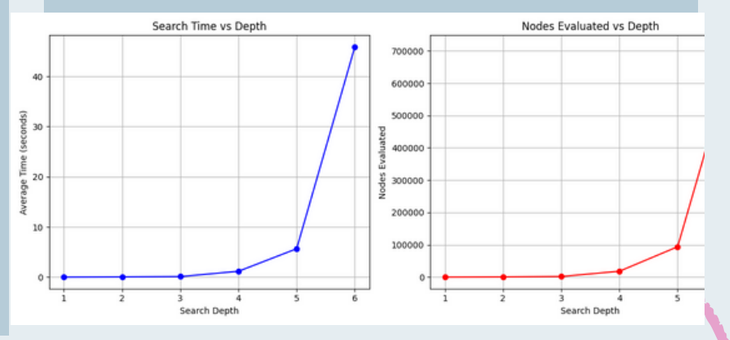
Utility Function
- Piece Count, Mills, Mobility, Threats, Phase Bonus
- Final Score: Weighted sum (details in visual box)

Algorithm
- Minimax: Game tree exploration for optimal moves
- Alpha-Beta Pruning: Efficiently trims search branches
- Depth Limiting: Maintains fast, responsive gameplay

RESULTS

Depth Analysis
- Search depth beyond 6 = infeasible for real-time
- Alpha-beta pruning: ~60-80% reduction in explored nodes
- Optimal depth: 5–6 for <5s move time

Key Graphs:
- Search Time vs Depth
- Nodes Evaluated vs Depth

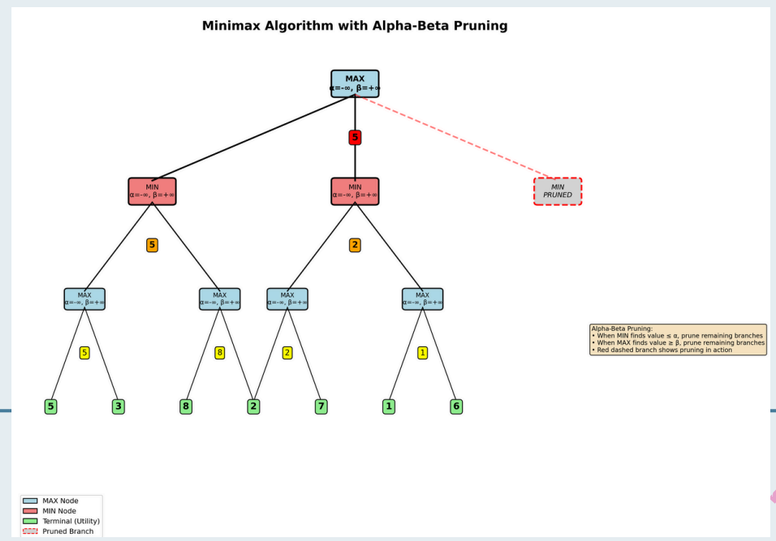


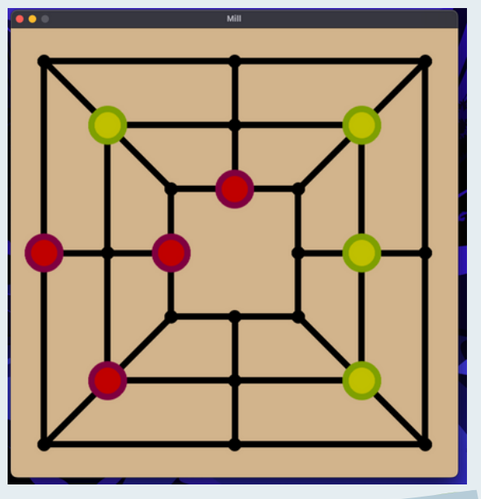
IMPLEMENTATION

Features
- Custom minimax & alpha-beta implementations
- Adaptable UI
- Tournament mode
- Code in Python; **famnit-gym** for environment

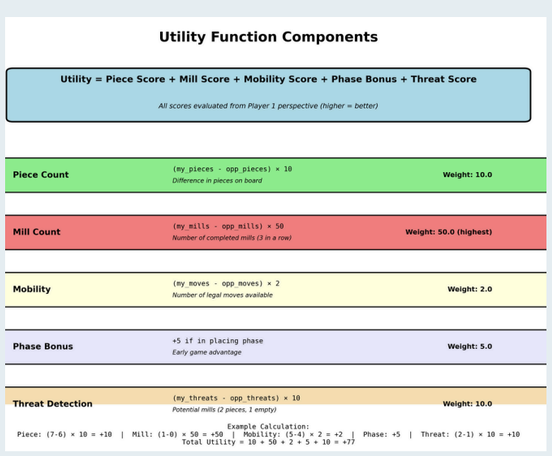
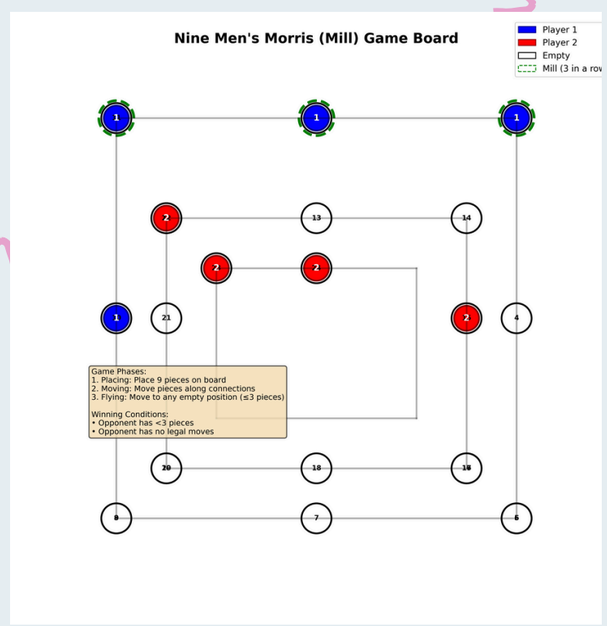
DIFFICULT LEVELS

Level: Easy	Medium	Hard
Depth: 2	4	6
Random: 30%	15%	0%
Desc: Randomized moves	Occasional errors	Full adversarial





CONCLUSION

- Alpha-beta pruning makes the AI fast enough for real-time play.
- Our utility function creates strong and adaptable gameplay.
- Difficulty levels ensure the game is challenging for all players.
- When two strong AIs play, most games end in a draw, showing solid strategy.

Github Repository
For More Visit the Repository here