**Project Report**

Project Title: Smart Mini Go: AI Powered Strategy Game

Submitted By: Fatima Farhan, Najaah Noor

Course: AI

Instructor: Abdullah Yaqoob

Submission Date: 11/05/2025

# 1. Executive Summary

* **Project Overview**
  + This project presents a new version of the traditional board game Go, called Smart Mini Go. The game introduces new rules that limit where players can place their pieces and simplify how the game is won. The goal of this project was to build an AI agent that plays this version of the game using classical search techniques. We used the minimax algorithm with Alpha Beta pruning to help the AI make good decisions. The game was also developed with a user-friendly interface to allow human players to play against the AI or each other.

# 2. Introduction

* **Background**
  + Go is a well-known strategy board game where players take turns placing black or white stones on a grid, trying to surround territory and capture the opponent’s stones. We chose to base our project on Go because of its rich strategic depth and the challenge it presents for AI. However, to make the game easier to implement and analyze, we created a simplified version. In Smart Mini Go, the rules are more focused, and the conditions for winning are clearer and quicker to reach.
* Objectives of the Project
* To build an AI agent that can play Smart Mini Go by making smart decisions.
* To design and test an evaluation method that helps the AI assess different game positions.
* To create a working version of the game with a graphical interface.
* To measure how well the AI performs in real games.

# 3. Game Description

* Original Game Rules
* In the classic Go game, players place stones on a grid trying to surround and capture areas. Stones that are fully surrounded are removed from the board. The game ends when both players agree there are no more useful moves, and the player who controls more territory wins.
* Innovations and Modifications

We changed the rules of the game in the following ways:

1. Players can place a stone only on a space that is part of the opponent’s liberties, unless they need to protect their own group.
2. Moves that would immediately result in the player’s own stone group being captured are not allowed.
3. The game ends when one player captures a group or when the opponent has no valid moves remaining.
4. Black always plays first and the first move must be at the center of the board.

# 4. AI Approach and Methodology

**AI Techniques Used**

We used the Minimax algorithm with Alpha Beta pruning. This helped the AI consider different possible future moves and ignore those that were clearly not helpful. It made the decision process faster and more efficient.

**Algorithm and Heuristic Design**

The AI uses a custom-designed evaluation method that looks at the current board. It checks how many liberties are available, how close the player is to capturing the opponent’s stones, and how much control each player has over the board. These factors are combined to give each position a score, and the AI uses this to choose its move.

**AI Performance Evaluation**

To test the AI, we observed how it played in different situations. We looked at how often it won, how quickly it made decisions, and whether it avoided mistakes. The addition of Alpha Beta pruning made the AI much quicker without reducing the quality of its moves.

# 5. Game Mechanics and Rules

**Modified Game Rules**

Stones can only be placed on opponent liberties, unless protecting your own group.

Moves that lead to an immediate capture of the player’s own group are not allowed.

Capturing a group or forcing the opponent into a position with no valid moves ends the game.

**Turn-based Mechanics**

Players take turns one after the other. Black always begins the game, and the first move is placed at the center of the board.

**Winning Conditions**

A player wins if they capture the opponent’s group or if the opponent cannot make any valid move

# 6. Implementation and Development

**Development Process**

The project started by finalizing the game rules and creating a basic AI design. The Minimax algorithm was implemented first, followed by improvements using Alpha Beta pruning. Once the AI was working, we created a graphical user interface using Pygame so that users could play the game easily. We tested the game regularly and made changes based on how the AI performed.

**Programming Languages and Tools**

* Language: Python
* Libraries: Pygame for graphics and NumPy for managing game data
* Tools: All code was written and tested using local development environments

**Challenges Encountered**

Some challenges included making sure the AI did not choose invalid moves, designing a good evaluation method, and keeping the game interface responsive while the AI was thinking. We solved these by writing clear rules for move validation and improving how the AI calculates its next move.

# 7. Team Contributions

* Fatima: Developed the AI algorithms and worked on designing the evaluation method.
* Najaah: Focused on defining the game rules, creating the user interface, and connecting it with the AI.

# 8. Results and Discussion

**AI Performance**

The AI performed well in test games. It was able to make decisions quickly and won a majority of the games against test players. The evaluation method helped it choose strong moves, especially in situations where it had to defend or go for a capture. Adding Alpha Beta pruning made the response time fast enough to allow smooth and competitive play.

# 9. References

* Go (game) - Wikipedia: https://en.wikipedia.org/wiki/Go\_(game)
* Go - Britannica: <https://www.britannica.com/topic/go-game>
* https://online-go.com/learn-to-play-go