Project Report

Project Title: AI-Powered Battleship Game with Monte Carlo Strategy

Submitted By:

Muhammad Safdar (22K-4304)

Abdul Rafiu (22K-4162)

Syed Mohammed Rayyan Imam (22K-4153)

Course: AI

Instructor: Shafique Rehman

Submission Date: 11 May 2025

1. Executive Summary

Project Overview:
This project modifies the traditional Battleship game to introduce a competitive player-vs-AI experience. Enhancements include dynamic difficulty levels and power-up features. The AI opponent uses an advanced Monte Carlo simulation approach for decision-making, especially in "Hard" mode, allowing for intelligent target prediction and adaptive strategy.

2. Introduction

Background:
Battleship is a two-player guessing game where each player places ships on a grid and takes turns attempting to hit the opponent's ships. The game ends when all of one player's ships are sunk. This version enhances the traditional gameplay by integrating AI decision-making, and additional power-ups to increase strategic depth.

Objectives of the Project:

- To develop an AI capable of playing Battleship effectively.

- To compare AI difficulty modes.
- To integrate Monte Carlo simulations for optimal move predictions.
- To create an interactive GUI using tkinter.

3. Game Description

Original Game Rules:
Players place ships of various sizes on a 10x10 grid. Each turn, a player chooses a cell to attack. If a ship occupies that cell, it is a hit; otherwise, it is a miss. The game ends when one player's fleet is entirely sunk.

Innovations and Modifications:

- A GUI interface for user interaction.
- Two AI difficulty levels: Easy (random targeting with memory) and Hard (enhanced Monte Carlo strategy).
- Power-ups: Missile (3x3 area strike), Destroyer (entire row attack), and Intel (random scouting).

4. AI Approach and Methodology

AI Techniques Used:

- Rule-based strategy for Easy mode.
- Enhanced Monte Carlo simulations with heatmap analysis for Hard mode.

Algorithm and Heuristic Design:

- The Monte Carlo approach simulates thousands of possible ship placements based on known hits and misses.
- A heatmap is built to prioritize attack cells with the highest likelihood of a hit.
- Heuristics include line detection for multiple hits and avoidance of already attacked cells.

AI Performance Evaluation:

- Evaluated through simulated games across both modes.
- Key metrics: win rate (vs. human and self-play), decision speed, and effective usage of power-ups.

5. Game Mechanics and Rules

Modified Game Rules:

- Added power-ups with limited uses.
- Monte Carlo decision-making for the AI in hard mode.
- Optional difficulty selection at game start.

Turn-based Mechanics:

- Players alternate turns.
- The AI adapts its strategy dynamically depending on mode and board state.

Winning Conditions:

- A player wins when all enemy ships are hit and sunk.

6. Implementation and Development

Development Process:
The project was built using Python with tkinter for the interface. Game logic and AI behavior were iteratively developed and tested. GUI design, AI refinement, and power-up integration were implemented in phases.

Programming Languages and Tools:

- **Language:** Python
- **Libraries:** tkinter, random
- **Tools:** GitHub (version control), local IDE

Challenges Encountered:

- Managing AI complexity and runtime in Monte Carlo simulations.
- Handling overlapping AI strategies with power-ups.
- Balancing difficulty between modes.
-

7. Team Contributions

- **Safdar:** Designed and implemented the enhanced Monte Carlo AI.
- **Abdul Rafiu:** Created the GUI using tkinter and integrated user interaction.
- **Rayyan:** Developed and tested power-up functionalities, evaluated and compared AI performance across difficulty modes.

8. Results and Discussion

AI Performance:

- Hard mode AI won approximately 70–80% of games against novice players.
- Decision-making time averaged 1–2 seconds per move with 1000 simulations.
- Easy mode provided a more casual experience, relying on random but informed choices.

9. References

- Python documentation: https://docs.python.org/3/
- Tkinter GUI guide: https://tkdocs.com/
- Monte Carlo method concepts: Wikipedia, academic articles
- Battleship rules: https://en.wikipedia.org/wiki/Battleship_(game)