AI Project | Project Groups Details
Project Title: Mini Battleship
Submitted By: Muhammad Shahzad [22k-4565] (group members : umer razi [22k-4451] , huzaifa sohail [22k-4450] , taushar lekhraj[22k-4532])
Course: AI
Instructor: Talha Shahid
Submission Date: 23- march-2025

---

1. Project Overview

Project Topic:
Mini Battleship is a smaller, faster-paced version of the classic Battleship game. The game is played on a 5x5 grid, where players place ships and take turns guessing locations to sink each other's ships. The first player to sink all opponent's ships wins.

Objective:
The main goal of this project is to develop an AI opponent for Mini Battleship that uses a probability-based attack strategy. The AI will start with random guesses and adjust its strategy based on previous hits to improve its chances of winning.

---

2. Game Description

Original Game Background:
Battleship is a classic two-player guessing game where each player places ships on a hidden grid and takes turns calling out coordinates to attack the opponent's ships. The goal is to sink all of the opponent's ships before they sink yours.

Innovations Introduced:

- **Smaller Grid:** The game is played on a 5x5 grid instead of the traditional 10x10, making it faster and more accessible.
- **Simplified Ships:** Fewer ships are used, and their sizes are adjusted to fit the smaller grid.
- **Probability-Based AI:** The AI uses a probability-based strategy to make smarter guesses, improving its chances of winning.

Impact on Gameplay:

- The smaller grid increases the pace of the game, making it more engaging for quick matches.
- The AI's probability-based strategy adds a layer of complexity and challenge for human players.

---

3. AI Approach and Methodology

AI Techniques to be Used:

- **Probability-Based Attack Strategy:** The AI will calculate the probability of a ship being present at each cell based on previous hits and misses.
- **Random Initial Guesses:** The AI will start with random guesses to gather initial data.
- **Adaptive Strategy:** The AI will adjust its strategy based on hits, focusing on adjacent cells to sink ships.

Heuristic Design:

- **Hit Probability:** Cells adjacent to hits are given higher probability scores.
- **Ship Placement Rules:** The AI will consider the possible positions of remaining ships based on their sizes.

Complexity Analysis:

- **Time Complexity:** The probability-based strategy is efficient, with a time complexity of $O(n^2)$ for an n x n grid.
- **Challenges:** Ensuring the AI adapts quickly to new information and avoids redundant guesses.

---

4. Game Rules and Mechanics

Modified Rules:

- **Grid Size:** 5x5 grid for faster gameplay.
- **Ships:** Each player has 2-3 ships of varying sizes (e.g., 2-cell and 3-cell ships).
- **Guessing:** Players take turns calling out coordinates (e.g., A1, B3) to attack.

Winning Conditions:
The first player to sink all of the opponent's ships wins the game.

Turn Sequence:

- Players take turns guessing coordinates.
- If a guess hits a ship, the player gets another turn.
- If a guess misses, the turn passes to the opponent.

---

5. Implementation Plan

Programming Language: Python

Libraries and Tools:

- **Pygame:** For creating the game's graphical user interface (GUI).
- **NumPy:** For handling probability calculations and grid operations.
- **Random:** For generating initial random guesses.

Milestones and Timeline:

- **Week 1-2:** Design the game mechanics and finalize rules.
- **Week 3-4:** Implement the probability-based AI strategy.
- **Week 5-6:** Develop the game GUI using Pygame.
- **Week 7:** Integrate the AI with the game and test gameplay.
- **Week 8:** Final testing, debugging, and report preparation.