National University of Computer & Emerging Sciences
Karachi Campus

AI-Powered Battleship Game with Enhanced Monte Carlo Strategy

Project Proposal

Artificial Intelligence

Section: BCS-6B

Group Members:
22k-4304 Muhammad Safdar
22k-4162 Abdul Rafiu
22k-4153 Syed Mohammed Rayyan Imam

Project Proposal

- **Introduction**
  We propose to develop an AI-powered version of the classic Battleship game with enhanced intelligence and interactive features. The game will include a graphical user interface (GUI), two difficulty modes (Easy and Hard), and strategic power-ups to improve gameplay. The AI will use Monte Carlo simulations for decision-making to provide a challenging experience.

- **Existing System**
  Classic Battleship games exist as board games or simple digital implementations. These versions often use static or random AI that lacks adaptive strategy. Some digital games provide basic rule-based bots, but they fail to analyze prior hits effectively or simulate realistic opponent behavior.

- **Problem Statement**
  Most existing AI opponents in Battleship games rely on static or naive random behavior. They do not adapt based on prior game states (hits, misses, sunk ships), making gameplay predictable and unchallenging. Additionally, most lack interactive enhancements like power-ups or adjustable difficulty levels.

- **Proposed Solution**
  Our game introduces a Monte Carlo-based AI in Hard mode that runs thousands of simulations to identify the most probable ship locations. It considers historical hits and misses to prioritize moves intelligently. We also provide power-ups (Missile, Destroyer, Intel) that users and the AI can use to influence the game strategically. A difficulty selector will let users choose between casual and competitive gameplay.

- **Salient Features**
  - Interactive GUI using tkinter
  - Two AI difficulty levels: Easy (random targeting with memory) and Hard (Monte Carlo strategy)
  - AI logic considers unsunk hits for targeted strategies
  - Power-ups:
    - Missile: Hits a 3x3 area
    - Destroyer: Attacks an entire row
    - Intel: Scans five random positions
  - Ship placement and targeting validation
  - Visual feedback on hits, misses, and sunk ships
  - Balanced AI gameplay with strategic decision-making

- **Tools & Technologies**

  - **Programming Language:** Python
  - **Framework/Libraries:** tkinter (GUI), random
  - **Operating System:** Windows/Linux