**AI Project Report**

**Project Title:** Strategic Snakes & Ladders with Heuristic-Based AI  
**Submitted By:** Shaheer Ahmed Qidwai

**Group members:**Shaheer Ahmed 22k-4541  
Atif Arif 22k-4358  
Abdul Rafay 22k-4517  
  
**Course:** AI  
**Instructor:** Ms. Alishba Subhani  
**Submission Date:** 10th, May 2025

### **1. Executive Summary**

**Project Overview:**This project presents a modified version of the traditional Snakes & Ladders game, enhanced with strategic gameplay mechanics and an AI opponent. The key innovations include the addition of multiple paths (main, safe, and risky), trap and power-up tiles, and a heuristic-based AI that makes decisions on path selection, trap placement, and power-up usage. The project aims to create a more engaging game experience while showcasing decision-making AI.

### **2. Introduction**

**Background:**Snakes & Ladders is a classic board game played on a 10x10 grid, where players move based on dice rolls. Traditionally, the game relies solely on chance. In this project, the game is modernized by adding new strategic elements: forked paths, trap placements, and power-up decisions. These additions introduce a layer of strategy and make the game suitable for AI-driven decision-making.

**Objectives of the Project:**

* Modify the Snakes & Ladders game with strategic elements.
* Implement a playable Human vs AI mode.
* Develop a heuristic-based AI that makes intelligent gameplay decisions.
* Integrate UI elements for trap placement and power-up selection.

### **3. Game Description**

**Original Game Rules:**Players roll a die and move their token forward. Landing on a ladder moves the player up, and landing on a snake moves the player down. The first player to reach the tile number 100 wins.

**Innovations and Modifications:**

* **Forked Paths:** Safe and risky paths diverge from tile 20.
* **Traps:** Each player can place two traps on the board that cause the opponent to lose a turn, once the opponent lands on a trap their next turn is skipped and the trap is finished.
* **Power-ups:** Landing on a power-up tile allows a player to choose between disarming the opponent’s traps or gaining an extra dice roll.
* **Visual Feedback:** Dice animations, token rendering, and UI buttons enhance the user experience.

### **4. AI Approach and Methodology**

**AI Techniques Used:**

* A heuristic-based AI is used instead of full Minimax or Expectimax due to the high branching factor and probabilistic dice rolls.
* For the sake of fast and efficient decision making the heuristic-based model is used rather than Minimax which would overcomplicate the game.

**Algorithm and Heuristic Design:**

* **Path Choice Heuristic:** AI prefers risky path when behind and safe path when ahead, also considering the traps placed on either paths as well.
* **Trap Placement Heuristic:** AI places traps when the human is ahead and likely to land on the trap within 3–6 tiles. The traps are efficiently used when the human is within reach and by placing a trap the AI can likely catch up or move ahead of the human.
* **Power-up Strategy:** AI chooses to disarm traps if any exist; otherwise, it selects an extra roll.

**AI Performance Evaluation:**

* AI performs competitively and adapts its strategy based on the human player's position.
* The heuristic-based decision-making proves effective in dynamic scenarios.

### **5. Game Mechanics and Rules**

**Modified Game Rules:**

* Players can place up to 2 traps.
* Players choose between safe or risky paths at fork tiles.
* Power-ups offer strategic choices when landed on.

**Turn-based Mechanics:**

* Human and AI alternate turns.
* Players click to roll dice (human) or auto-roll (AI).
* Players affected by traps skip one turn.

**Winning Conditions:**

* First player to reach tile 100 wins.
* Overshooting tile 100 is allowed for win.

### **6. Implementation and Development**

**Development Process:**

* Iteratively built the game using Pygame.
* Gradually added board layout, player logic, traps, powerups, and AI behavior.

**Programming Languages and Tools:**

* Language: Python
* Libraries: Pygame, Math
* Tools: GitHub for version control

**Challenges Encountered:**

* Managing turn-based flow during UI interactions.
* Making the overall board design for a multipath scenario.
* Ensuring traps and power ups didn’t conflict with snakes/ladders.
* Preventing AI from making poor trap placements.

### **7. Team Contributions**

**Shaheer Ahmed:**

* Designed and implemented the full game logic in Python and pygame.
* Developed AI logic using heuristics.

**Atif Arif:**

* Created the overall UI components using pygame.
* Integrated traps and powerups.

**Abdul Rafay:**

* Made the logic and design for the multipath board design in pygame.
* Conducted the performance testing of the AI based logic and evaluation of the AI’s decisions.

### **8. Results and Discussion**

**AI Performance:**

* The AI demonstrates solid decision-making and competitiveness.
* Successfully placed effective traps and chose appropriate paths, not overusing the available traps within the start of the game.
* Maintains win/loss balance in human-AI games with reasonable outcomes

### **9. References**

* Python Pygame Documentation
* AI heuristics and game tree tutorials
* Stack Overflow discussions
* Classic Snakes and Ladders rules