Project Title: Snakes & Ladders - Dungeon Escape

Submitted By: [Amna Emaan, Hania Fahim, Wali Yar Khan]
Course: AI
Instructor: [Ms. Mehek Mazhar, Ms Saeeda Kanwal]
Submission Date: [8/3/2025]

---

1. Project Overview

Project Topic:

Snakes & Ladders - Dungeon Escape is an AI-driven twist on the classic board game. In this version, players navigate through a dungeon where **snakes are monsters** and **ladders are secret tunnels**. The AI dynamically places obstacles to prevent the player from reaching the final destination (Tile 100). Players must strategize using power-ups like shields, teleportation, and snake repellents while racing against time to escape.

Objective:

The goal is to create a **challenging AI opponent** that dynamically adjusts difficulty by placing obstacles (snakes) in response to the player’s movements. The project will focus on **game AI, heuristic-based decision-making, and real-time strategy adaptation.**

---

2. Game Description

Original Game Background:

The classic **Snakes & Ladders** game consists of a **100-tile board** where players select moves (1-6) to move forward. Landing on a ladder advances the player, while landing on a snake pushes them back. The objective is to reach Tile 100 first within time constraint.

Innovations Introduced:

- **AI as Dungeon Master:** AI dynamically places snakes (monsters) based on the player's movement pattern, making the game more unpredictable.
- **Power-Ups for Player:** Players can collect shields, teleportation scrolls, snake repellents, and AI slow-down items to increase their chances of winning. But these power ups will be hidden at first and be only visible when reaching that tile (luck -based)

- **Dungeon Theme:** The board represents a dungeon where ladders are **hidden tunnels**, and players can collect **keys, weapons, and potions** to fight monsters.
- **Time Constraint:** The player must **escape before time runs out**, adding urgency to the gameplay.
- **Final Boss Mechanic:** AI will attempt a **final challenge** by sending a powerful monster when the player is near escape (Tile 90+), making the endgame more intense.

These changes make the game **strategic**, **AI-driven**, and **interactive**, rather than just luck-based.

---

3. AI Approach and Methodology

AI Techniques to be Used:

- **Rule-Based AI**: AI analyzes the player's movement pattern and places snakes dynamically.
- **Heuristic-Based Decision Making**: AI predicts high-risk tiles and places obstacles accordingly.
- **Path Optimization**: AI calculates the best positions to slow down the player strategically.
- **Monte Carlo Simulation (Optional)**: AI can test different obstacle placements and choose the best.
- **Minimax Algorithm**: AI decision-making for snake placement based on player movement prediction.
- **Alpha-Beta Pruning (if applicable)**: Optimizing Minimax to reduce computational complexity.
- **Scikit-learn (Optional)**: AI training for adaptive difficulty using reinforcement learning techniques.

Heuristic Design:

- AI evaluates the board to determine **where the player is likely to move** and places snakes accordingly.
- AI adjusts difficulty based on **player progress and power-up usage**.

Complexity Analysis:

- **AI Decision Making:** O(n) complexity for board evaluation and snake placement.
- **Player Movement Simulation:** O(1) per move.

- **Path Prediction (if implemented):** O(n log n) for predicting the best/worst paths for the player.

---

4. Game Rules and Mechanics

Modified Rules:

- **Player** selects (1-6) steps to move forward.
- **AI dynamically places new obstacles (snakes) predicting what step user can take**
- **Power-ups randomly appear on tiles**:
  - 🛡 **Shield:** Blocks the next snake.
  - ⚡ **Teleport:** Instantly moves the player forward by 10 tiles.
  - **Snake Repellent:** Neutralizes the nearest snake.
  - 🌀 **Slow Down AI:** AI cannot place new obstacles for 2 turns.
- **Keys & Weapons:** Some tunnels require a **key** to use, and weapons allow players to **fight monsters** instead of falling down.

Winning Conditions:

- **Player wins by reaching Tile 100 within the time limit.**
- **AI wins if time runs out or the player is forced back too many times.**

Turn Sequence:

1. **Player selects step move and moves forward.**
2. **Player encounters power-ups, tunnels, or monsters.**
3. **AI reacts after every turn by placing new obstacles.**
4. **Game continues until the player reaches 100 or time expires.**

---

5. Implementation Plan

Programming Language: Python

Libraries and Tools:

Tool	Purpose
Python	Main programming language
Pygame	For graphical user interface (GUI)
NumPy	Dice roll probability calculations
Scikit-learn (Optional)	AI training for adaptive difficulty
Minimax Algorithm	AI decision-making for snake placement based on player movement prediction
Monte Carlo Simulation	Player move prediction
Alpha-Beta Pruning (if applicable)	Optimizing Minimax for better performance
Heuristics	Smart placement of snakes to increase game difficulty

Milestones and Timeline:

- **Week 1-2:** Finalize game design, rules, and AI approach.
- **Week 3-4:** Implement board, movement mechanics, and AI snake placement logic.
- **Week 5-6:** Add power-ups, weapons, and dungeon mechanics.
- **Week 7:** Integrate AI strategy and optimize difficulty scaling.
- **Week 8:** Final testing, bug fixes, and documentation.

---

6. References

- Game development tutorials using **Pygame**
- AI strategy design for **board games and adversarial play**