National University of Computer & Emerging Sciences
Karachi Campus

Snake and Ladder Game

Project Proposal
Artificial Intelligence
Section: BCY-6A
Instructor name: Ms. Mehak Mazhar
Date: 8-3-2025

Group Members:
(22K-4746) - Jahanzeb Khairi
(22K-4754) - Murtaza Rizvi
(22K-4690) - M. Yahya Khan

Project Overview

Project Topic

This project focuses on an innovative variant of the classic Snake and Ladder game. The modifications include:

- **Power-Ups and Penalties** to introduce an element of strategy.
- **AI-based Opponent** using Minimax and Reinforcement Learning for decision-making in multi-player mode.

Project Objective

The main goal of this project is to develop an AI-driven Snake and Ladder game with increased complexity and strategic depth by introducing power-Ups, Penalties and AI-controlled opponents. The AI will evaluate risk, decide optimal moves, and adapt to board changes using heuristic-based decision-making.

Game Description

Original Game Background

Snake and Ladder is a classic board game played on a grid of 1 to 100 numbered squares. Players roll a die to move forward, climbing ladders when landed upon and sliding down snakes when encountered. The first player to reach the final square wins.

Innovations Introduced

Innovations Introduced:

- **Power-ups and Penalties**:
  - "Shield" prevents sliding down a snake.
  - "Boost" allows an extra dice roll.
  - "Trap" forces an opponent to skip a turn.
- **AI Opponent**:
  - Uses **Minimax Algorithm** to assess dice roll impact.
  - **Reinforcement Learning** to adapt strategies over multiple games.
- **Alternative Winning Conditions**: Instead of just reaching 100, new conditions may include collecting points based on ladder climbs and avoiding penalties.

AI Approach and Methodology

AI Techniques to be Used

- **Minimax Algorithm**: Determines the best dice roll outcome for AI opponents.
- **Alpha-Beta Pruning**: Optimizes decision-making to reduce unnecessary computations.

Heuristic Design

- **Positional Advantage**: AI assigns weights to squares based on risk (snake nearby) and rewards (ladder).
- **Probability Calculation**: AI evaluates possible rolls to maximize positive outcomes and avoid penalties.

Complexity Analysis

- Minimax time complexity: $O(b^d)$, where **b** is the branching factor (6 dice outcomes) and **d** is the game depth.
- RL Training complexity depends on the number of **simulated games** used for model optimization.

Game Rules and Mechanics

Modified Rules

1) Players can collect and use power-ups.
2) AI-controlled players use strategy instead of random dice rolls.

Winning Conditions

- **Classic mode**: Reach the final square first.
- **Strategy mode**: Win by accumulating the most points.
- **AI Challenge mode**: Compete against AI for highest efficiency.

Turn Sequence

1) Roll the dice.
2) Move forward unless AI decides to use a power-up.
3) If landing on a snake or ladder, apply board effects.
4) AI evaluates game state and adjusts strategy.

Implementation Plan

Programming Language

Python

Libraries and Tools

- **Tkinter** (for GUI development)
- **PIL (Pillow)** (for image handling and board rendering)
- **NumPy** (for AI computations)
- **Scikit-learn** (for decision heuristics)

Milestones and Timeline

- **Week 1-2**: Define new rules, board dynamics, and AI framework.
- **Week 3-4**: Implement game logic, dice rolling, and power-ups.
- **Week 5-6**: Develop and integrate AI decision-making strategies.
- **Week 7**: AI self-play testing and fine-tuning.
- **Week 8**: Final testing and report preparation.

References

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.
[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.
[3] Python Software Foundation, "Tkinter documentation," Available: https://docs.python.org/3/library/tkinter.html.
[4] A. Sharma, "Enhancing Board Games with AI: A Case Study on Minimax and RL," *International Journal of Game Theory and AI*, vol. 12, no. 3, pp. 45-60, 2021.
[5] Python Imaging Library (PIL) documentation, Available: https://pillow.readthedocs.io/.