Treasure Hunt Tactics

Submitted by:
22K-4456 Syed Muhammad Shuja Ur Rahman
22I-2127 Abdul Qadir Tareen
22K-4546 Fiza Farooq

Course: Artificial Intelligence

Instructor: Ms. Alina Arshad, Ms Ravia Ejaz

Submission Date: 15/May/2025

1. Introduction

The *Treasure Hunt Tactics* project is a turn-based strategy board game that integrates Artificial Intelligence (AI) concepts such as Reinforcement Learning (RL) and adversarial decision-making using Minimax. Players navigate a grid-based board to reach a treasure while facing dynamic traps placed by an intelligent AI opponent. The project aims to demonstrate how AI can adapt its strategies to hinder human or agent players in real-time gameplay.

2. Objectives

- To design an engaging and interactive board game with AI elements.
- To implement trap placement using **Minimax** to increase game difficulty strategically.
- To use **Reinforcement Learning** to guide player decision-making and adapt over time.
- To evaluate and visualize AI behavior in a game environment.

3. Technologies Used

- **Programming Language:** Python
- **Libraries/Modules:**
  - pygame – for graphical interface and game loop
  - random, math, copy – for decision logic
  - Custom utility functions for evaluation and AI strategies

- **AI Techniques:**
  - Reinforcement Learning (RL) and Minimax Algorithm with Alpha-Beta Pruning

4. Game Description

Treasure Hunt Tactics is a 2D grid-based adventure game developed using Pygame, where the player navigates a 6x6 board to collect hidden treasures while avoiding traps intelligently placed by AI.

4.1. Gameplay Mechanics:

- The player starts at the top-left tile and moves using arrow keys.
- The goal is to collect all treasures within a time limit and limited lives.
- Colliding with a trap reduces the player's lives and resets their position.
- Visual cues highlight nearby traps with red tile borders.

4.2. Levels & Features:

4.2.1. Level 1: Solo Exploration

- The player plays alone.
- Traps are placed dynamically by a Reinforcement Learning **agent** trained using an MLPRegressor.
- The agent adapts over time by rewarding or penalizing trap placements based on the player’s actions.

4.2.2 Level 2: Competitive Challenge

- A Rival AI is introduced, controlled by a Minimax algorithm with alpha-beta pruning**.**
- The rival competes with the player to collect treasures first.
- The AI evaluates future board states to block the player or reach treasures faster.

5. AI Implementation

5.1. Minimax Trap Placement

The AI simulates future player moves and places traps where they are most likely to be effective. The evaluation function rewards placements that maximize the chance of the player stepping on a trap.

5.2. Reinforcement Learning for the Player

The player uses a simple Q-learning-like approach to update its strategy based on previous successes and failures. Over multiple games, the player learns to avoid traps and find the optimal path to the treasure.

6. Conclusion

The *Treasure Hunt Tactics* game successfully integrates key AI strategies to build a challenging, adaptive environment. The project demonstrates the potential of combining **Reinforcement Learning** with **Adversarial Search** for intelligent behavior in games. This approach is applicable to other AI scenarios requiring strategic planning under uncertainty.

7. References

- Russell, S. J., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*.
- Pygame Documentation: https://www.pygame.org/docs/