Treasure Hunt Tactics

Submitted By:
22k-4456 Syed Muhammad Shuja Ur Rahman
22i-2127 Abdul Qadir Tareen
22k-4546 Fiza Farooq

Course: AI
Instructor: Syeda Ravia Ejaz
Submission Date: 22/03/2025

1. Project Overview

Project Topic:

Treasure Hunt Tactics is a turn-based board game where players move across a board collecting hidden treasures while avoiding AI-controlled traps. The AI adapts to player strategies using Reinforcement Learning and Minimax to place obstacles intelligently.

Objective:

The goal of this project is to develop a strategic AI capable of dynamically placing traps and predicting player movements using Minimax and Reinforcement Learning. The game will provide an engaging experience by adapting to different player strategies.

2. Game Description

Original Game Background:

This game draws inspiration from classic treasure hunt games where players explore a map to find hidden rewards. However, in this version, AI dynamically influences the environment by placing obstacles and traps to challenge players.

Innovations Introduced:

- **AI-Controlled Traps:** The AI will strategically place obstacles based on player movements and past behaviors.
- **Adaptive AI Strategies:** Reinforcement Learning will allow AI to learn optimal placements over multiple rounds.
- **Limited Hints:** Players receive a restricted number of hints regarding treasure locations, increasing challenge and strategic depth.
- **Dynamic Gameplay:** Every game session presents a unique board layout due to AI adaptability.

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm:** AI will use Minimax with heuristic evaluation to predict and counteract player moves.
- **Alpha-Beta Pruning:** To optimize Minimax efficiency, reducing unnecessary calculations.
- **Reinforcement Learning:** The AI will improve its trap placement strategies over multiple games by learning from player behavior.

Heuristic Design:

- **Trap Efficiency Score:** Measures how effectively an obstacle hinders player progress.
- **Player Movement Prediction:** AI will estimate probable player moves based on previous actions.
- **Treasure Accessibility Factor:** Ensures a balanced challenge by placing traps without making treasures impossible to reach.

Complexity Analysis:

- **Minimax Algorithm Complexity:** O(b^d), where **b** is the branching factor and **d** is the search depth.
- **Reinforcement Learning Complexity:** Depends on the learning rate and policy updates, increasing complexity as the AI adapts over multiple games.
- **Challenges:** Balancing difficulty so the AI does not become unbeatable while maintaining engagement for players.

4. Game Rules and Mechanics

Modified Rules:

- Players can move in four directions per turn.
- AI places traps dynamically based on player movement patterns.
- Players get limited hints about treasure locations.
- Traps slow down players or send them back to previous positions.

Winning Conditions:

- The player who collects the most treasure within a fixed number of turns wins.
- If a player finds all treasures before the turn limit, they win automatically.

Turn Sequence:

1. **Player Move:** Players choose a direction to move.
2. **AI Decision:** AI places or modifies traps based on its learned strategy.
3. **Hint Usage:** If a player has hints left, they can use one to reveal partial treasure locations.
4. **Game Progresses Until Win Condition is Met.**

5. Implementation Plan

Programming Language:

Python

Libraries and Tools:

- **Pygame:** For board game GUI and interaction.
- **NumPy:** For data handling and AI calculations.
- **Scikit-learn/TensorFlow:** For Reinforcement Learning implementation.

Milestones and Timeline:

- **Week 1-2:** Game design and rule finalization.
- **Week 3-4:** AI strategy development (Minimax and heuristics).
- **Week 5-6:** Coding and testing the game mechanics.
- **Week 7:** AI integration and testing.
- **Week 8:** Final testing and report preparation.