Project Title: *AI-Based Monopoly Game Variant*
Submitted By: *Hafsa Nauman (22k4274), Hafsa Fatima (22k4653), Haneesh Ali (22k4240)*
Course: *AI*
Instructor: *Abdullah Yaqoob*
Submission Date: *9th March 2025*

1. Project Overview

Project Topic:

We are developing a digital version of Monopoly with AI-driven opponents and enhanced game mechanics. This variant introduces new decision-making strategies for AI players, advanced financial management, and modified game rules to increase complexity and engagement.

Objective:

The goal of this project is to create a more strategic and immersive Monopoly experience by integrating AI-driven decision-making, a more interactive game board, and new rule modifications. The AI will use advanced algorithms to evaluate financial decisions, engage in trades, and participate in auctions. The game will also include a save/load feature to allow for seamless gameplay continuity.

2. Game Description

Original Game Background:

Monopoly is a popular board game where players compete to buy, trade, and develop properties. The objective is to dominate the board by accumulating wealth and bankrupting opponents. Players roll dice to move around the board, purchasing properties, collecting rent, and navigating random events via Chance and Community Chest cards.

Innovations Introduced:

- ***Enhanced AI Decision-Making:*** *AI evaluates property purchases, trade negotiations, and cash management more effectively.*
- ***New Chance & Community Chest Cards:*** *Additional effects such as double rent, turn-skipping, and property tax refunds.*
- ***Property Auctions:*** *Unsold properties are auctioned, allowing AI and human players to bid dynamically.*

● ***Improved Game Board UI:*** *Hover effects, enhanced player stats display, and color-coded property ownership.*
● ***Save & Load Feature:*** *Game state can be saved and resumed later.* ● ***AI Trade System:*** *AI can propose and evaluate trade offers based on property values and cash reserves.*

3. AI Approach and Methodology

AI Techniques to be Used:

● ***Minimax Algorithm:*** *To optimize AI decision-making in property purchases and trades.*
● ***Alpha-Beta Pruning:*** *To reduce computation time for Minimax-based decision trees.* ● ***Reinforcement Learning:*** *AI adapts its strategy based on game progression.* ● ***Monte Carlo Search Tree:*** *AI evaluates possible future states and selects the most advantageous moves.*

Heuristic Design:

● ***Property Valuation:*** *AI evaluates property worth based on rent potential and board positioning.*
● ***Cash Flow Management:*** *AI makes strategic decisions on when to mortgage properties or hold cash.*
● ***Risk-Reward Analysis:*** *AI assesses the profitability of trades and auctions.*

Complexity Analysis:

● *Minimax with multiple players increases the branching factor exponentially.* ● *Dynamic events introduce probabilistic elements, making exact evaluation more complex.*
● *AI must balance short-term cash needs with long-term property investments.*

4. Game Rules and Mechanics

Modified Rules:

● ***New Chance & Community Chest Cards*** *with strategic gameplay effects.*
● ***Auction System for Unsold Properties.***
● ***AI Trade System*** *where AI evaluates and engages in trades with players.*
● ***New property valuation mechanics*** *to improve AI decision-making.*

Winning Conditions:

- *The game ends when all but one player go bankrupt.*
- *Alternatively, the winner can be determined by the highest net worth after a set number of turns.*

Turn Sequence:

1. Roll Dice → Move Token → Perform Action (Buy, Rent, Draw Card, etc.).
2. AI/Player decides on purchases, trades, or mortgages.
3. AI evaluates board state and adjusts its strategy.
4. Repeat until win condition is met.

5. Implementation Plan

Programming Language:

- ***Python***

Libraries and Tools:

- ***Pygame*** *(for GUI, if applicable)*
- ***NumPy*** *(for data handling)*
- ***TensorFlow/PyTorch*** *(for reinforcement learning)*
- ***networkX*** *(for modeling property connections and graphs)*

Milestones and Timeline:

- ***Week 1-2:*** *Finalize game design and rule modifications.*
- ***Week 3-4:*** *Implement core game mechanics.*
- ***Week 5-6:*** *Coding and testing game mechanics.*
- ***Week 7:*** *AI integration and testing.*
- ***Week 8:*** *Final testing and report preparation.*

6. References

NetworkX — NetworkX documentation
Monopoly Simulation: Hack a board game in Python and learn MatPlotLib — Raspberry Pi Official Magazine