**Project Report Template**

Project Title:

**UNO Spinner Showdown: An AI-Powered Twist on Classic UNO**

**Submitted By:** Zain Abbas

**Course:** AI

**Instructor:** [Instructor’s Name]

**Submission Date:** [Date]

1. Executive Summary

**Project Overview:** This project reimagines the traditional UNO card game by introducing an AI opponent and a new spinner mechanic that injects random game-altering events. The AI makes strategic card selections based on a custom heuristic evaluation system, enhancing gameplay with competitive dynamics. The objective was to simulate intelligent play under unpredictable game circumstances and test the AI against human and random strategies.

2. Introduction

**Background:** UNO is a well-known turn-based card game where players aim to discard all cards by matching colors or values. The simplicity and popularity of UNO made it an excellent candidate for AI modeling. In this project, we innovate on the original by introducing a spinner that introduces random events such as drawing extra cards or swapping hands, increasing both complexity and excitement.

**Objectives of the Project:**

* Develop a rule-based AI agent to play UNO effectively.
* Introduce a random spinner mechanic and model AI adaptability.
* Evaluate AI’s performance against human players and a random-play bot.
* Enhance understanding of decision-making in games with partial information and chance elements.

3. Game Description

**Original Game Rules:** In traditional UNO:

* Players are dealt 7 cards.
* Turns proceed clockwise.
* Cards can be played by matching the top card on the discard pile by number or color.
* Special cards include Skip, Reverse, Draw Two, Wild, and Wild Draw Four.
* First to discard all cards wins.

**Innovations and Modifications:**

* **Spinner Mechanic**: Triggered every 3 rounds. Possible outcomes include:  
  + Draw 4 Cards
  + Skip Turn
  + Reverse Play
  + Swap Hands
* **AI Agent**: Makes choices using a heuristic evaluation that considers:  
  + Color frequency in hand
  + Priority of special cards
  + Game progression
* **Improved Wild Card Use**: AI dynamically selects the color most frequent in its hand.

4. AI Approach and Methodology

**AI Techniques Used:**

* **Heuristic-Based Decision Engine**: Due to hidden information and randomness, a full Minimax tree is not feasible. Instead, a utility score is computed for each playable move.
* **Optional RL Expansion**: Reinforcement Learning (Q-learning) considered for future enhancement.

**Algorithm and Heuristic Design:**

* Assign weights to special cards.
* Prefer actions that reduce hand size fastest.
* Avoid enabling opponent’s winning moves (if inferable).
* Spinner events are modeled probabilistically when planning ahead.

**AI Performance Evaluation:**

* **Win Rate**: Tracked over multiple games.
* **Decision Time**: Less than 1.5 seconds per move.
* **Adaptability**: Evaluated via playthroughs with different spinner frequencies and card distributions.

5. Game Mechanics and Rules

**Modified Game Rules:**

* Spinner triggers every 3 turns.
* Spinner has 4 possible outcomes that influence game flow.
* Each player (human or bot) must adapt to spinner effects immediately.

**Turn-Based Mechanics:**

* Standard clockwise turns unless reversed.
* After every 3 turns, the spinner activates and executes its effect.
* AI responds to spinner events in real-time.

**Winning Conditions:**

* First player to discard all cards wins.
* If no player can finish after a certain number of rounds, lowest card count determines the winner.

6. Implementation and Development

**Development Process:**

* Designed game flow and spinner logic on paper.
* Implemented game mechanics in Python.
* Built AI agent using rule-based heuristics.
* Tested spinner randomness with seeded inputs.
* Conducted match simulations for evaluation.

**Programming Languages and Tools:**

* **Language**: Python
* **Libraries**:  
  + colorama – For colored console output
  + random – For card shuffling and spinner events
  + (Optional) pygame – Future use for GUI
* **Tools**:  
  + GitHub (for version control)
  + Jupyter Notebook (for AI testing)

**Challenges Encountered:**

* Modeling partial observability of other players’ hands.
* Designing an AI that balances aggression with caution.
* Spinner events causing unexpected state transitions required robust exception handling.

7. Team Contributions

**Team Member:** Zain Abbas

* Designed game modifications and spinner logic
* Developed AI heuristic engine
* Implemented game logic and testing framework
* Conducted AI performance evaluations and improvements

(If this was a team project, list other contributors here with their roles.)

8. Results and Discussion

**AI Performance:**

* **Win Rate**: ~65% against random player; ~45% against human players
* **Avg. Decision Time**: 1.3 seconds
* **Spinner Impact**: Added unpredictability, but AI adapted well by holding Wilds and Draw Twos for late-game
* AI was particularly effective at selecting wild card colors based on current hand composition

9. References

* UNO Official Rules – Mattel Games
* Sutton & Barto – Reinforcement Learning: An Introduction
* https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory/
* Python Documentation – random, collections, itertools
* https://realpython.com/tutorials/
* <https://www.pygame.org/>