**Project Report: AI-Powered UNO Card Game**

**Course**: **Artificial Intelligence**

**Group Members**: Muhammad Ahmed Haque (22K-4232), Syed Muhammad Irtiza(22K-4638), Muzammil Soomro(22K-4598)

**Abstract**

Is report mein hum apne semester project, ek AI-powered UNO card game, ke bare mein discuss karenge. Ye game Python mein Pygame library ke sath develop kiya gaya hai, aur is mein ek heuristic-based AI opponent shamil hai jo player ke against khelta hai. Hum ne game ko modular structure mein banaya, with separate modules for card logic, game mechanics, aur UI, to ensure maintainability aur reusability. The project demonstrates AI concepts through decision-making, alongside software engineering principles, making it a valuable learning experience for our AI course.

**1. Introduction**

UNO ek popular card game hai jahan players apne cards ko discard karte hain by matching colors ya numbers, with special action cards like Skip aur Wild adding strategic depth. Hamara project objective tha to create a single-player UNO game jahan AI opponent intelligently khel sake, using AI techniques taught in our course. The goals were:

* Implement full UNO rules with a user-friendly interface.
* Develop a smart AI opponent using heuristic-based decision-making.
* Use modular programming to organize code effectively.
* Ensure the game is engaging aur visually appealing with an enhanced UI.

Ye project AI ke concepts jaise decision-making under uncertainty ko explore karta hai, sath hi game development aur software design ke practical aspects ko bhi cover karta hai.

**2. Methodology**

**2.1 Project Scope**

Hum ne ek single-player UNO game banaya jahan player AI ke against khelta hai. The game follows standard UNO rules:

* Players discard cards matching the top card’s color or value.
* Action cards (Skip, Reverse, DrawTwo) aur Wild cards (Wild, WildDrawFour) affect gameplay.
* The first player to discard all cards wins.

**2.2 AI Approach**

Hum ne **heuristic-based decision-making** use kiya for the AI opponent, jo har playable card ko score assign karta hai based on strategic factors:

* **Color Matching**: +5 points if the card matches the current color, to limit player’s options.
* **Color Prevalence**: +3 points if the card is of the AI’s most common color, to maintain playability.
* **Action Cards**: +10 points for cards like Skip or DrawTwo, to disrupt the player.
* **Wild Cards**: +8 points, with +2 extra for WildDrawFour, due to their flexibility and impact.

AI highest-scoring card ko select karta hai ya draws a card if no playable cards exist. This approach is simple, computationally efficient, aur UNO ke liye effective, aligning with our course’s focus on practical AI techniques.

**2.3 Modular Design**

To make the code organized aur reusable, hum ne project ko multiple modules mein divide kiya:

* **constants.py**: Stores game constants (e.g., card size, colors).
* **card.py**: Defines the Card class for card representation aur playability logic.
* **game\_logic.py**: Manages core mechanics (deck creation, AI decisions, action handling).
* **ui.py**: Handles rendering (cards, buttons, text) aur user input using Pygame.
* **main.py**: Orchestrates the game loop, connecting logic aur UI.

This modular structure improves maintainability aur allows future extensions, jaise multiplayer support ya different AI strategies.

**2.4 UI Enhancements**

Hum ne UI ko improve kiya to enhance user experience:

* Cards are centered with dynamic spacing to prevent overlap.
* Text is rendered with a black background for readability against any card color.
* Hover effects highlight playable cards, aur buttons (Draw, color selection) have rounded corners aur hover feedback.
* A gradient background replaces the plain green table for visual appeal.

**3. Implementation**

**3.1 Tools and Technologies**

* **Python 3.12**: Core programming language.
* **Pygame 2.6.1**: For graphical interface aur event handling.
* **NumPy**: For generating click sound effects.
* **Pyodide**: Ensures compatibility for browser-based execution.

**3.2 Key Features**

* **Full UNO Rules**: Supports all standard cards aur actions, with proper handling of deck reshuffling when empty.
* **AI Opponent**: Uses heuristics to make smart decisions, challenging the player.
* **Interactive UI**: Clickable cards, a Draw button for no playable cards, aur color selection for wild cards.
* **Modular Code**: Separated into five files for clarity aur reusability.
* **Polished Design**: Centered hands, readable text, hover effects, aur a gradient background.

**3.3 Development Process**

Hum ne project ko iteratively develop kiya:

1. Implemented basic UNO rules aur a simple UI.
2. Added the heuristic-based AI, testing it for balance.
3. Enhanced the UI based on feedback (e.g., fixing text overlap, adding hover effects).
4. Modularized the code to improve structure.
5. Tested extensively to ensure no bugs, especially in AI decisions aur UI rendering.

**4. Results and Discussion**

The UNO game successfully meets all objectives:

* **AI Performance**: The heuristic-based AI plays strategically, often using action cards to disrupt the player aur choosing wild card colors based on its hand. It provides a challenging yet fair opponent.
* **UI Quality**: The interface is intuitive, with no card overlap, clear text, aur engaging visuals. Hover effects aur smooth button interactions enhance the experience.
* **Modularity**: The code is well-organized, making it easy to maintain aur extend. For example, adding a multiplayer mode would only require changes to game\_logic.py aur main.py.
* **Educational Value**: The project taught us AI concepts (utility functions, decision-making) aur practical skills in game development aur software engineering.

**Challenges**:

* Balancing AI difficulty was tricky; we adjusted heuristic scores to avoid making the AI too aggressive.
* Ensuring UI readability required adding text backgrounds, as some card colors obscured text.
* Modularizing the code required careful management of imports to avoid errors, jaise missing Card class in ui.py.

**Limitations**:

* The AI is greedy, focusing on the current turn without long-term planning. Advanced techniques like minimax could improve it but were beyond scope.
* The game is single-player; multiplayer would require network programming.

**5. Conclusion**

Hamara UNO card game project ek successful implementation hai of AI and game development concepts. The heuristic-based AI demonstrates intelligent decision-making, the modular code showcases good software design, aur the enhanced UI ensures an engaging user experience. Ye project ne hame sikhaya how to apply AI techniques practically aur manage a complex software project. In the future, we could extend it with multiplayer support, advanced AI (e.g., reinforcement learning), ya card images for a more realistic look.

**6. Acknowledgments**

We thank our course instructor for guidance aur the Pygame community for resources. Special thanks to [Group Member’s Name] for collaboration aur testing.

**7. References**

* Pygame Documentation: https://www.pygame.org/docs/
* UNO Official Rules: https://www.unogame.com/
* AI Concepts: [Course Textbook or Lecture Notes, if applicable]