

PROJECT: BLACKJACK SIMULATION

By Shikan Chen 23202508

01 RESEARCH

Investigation into applying reinforcement learning to casino games, focusing on Blackjack to simulated decision-making processes and strategy development in AI models.



02 ANALYSIS

Examination of standard Blackjack strategies and potential AI enhancements, employing statistical and machine learning techniques to analyze performance and optimize decision-making.



03 OBJECTIVES

Develop a robust Blackjack simulation that: 1) employs an AI trained via Q-learning, 2) adheres to traditional Blackjack rules, 3) improves AI decision-making through continuous gameplay and feedback.



Why I Chose Q-Table for Blackjack Simulation
The Q-table approach in our Blackjack AI utilizes Q-learning, a model-free reinforcement learning algorithm, which is highly adaptable to the unpredictable nature of Blackjack. This method allows the AI to learn optimal strategies directly from gameplay without needing a predefined model of the environment. It provides a straightforward, efficient way to estimate and update the utility of each action in every game state, facilitating dynamic learning and continuous strategy improvement as more games are played. This choice ensures that the AI progressively refines its decision-making skills, making our simulation both robust and scalable.



```
1 Blackjack-Simulation/
2   ai_models/          # Trained AI models and data
3   ...
4   images/             # Images and figures used or produced by the project
5   ...
6   ...
7   results/            # Output results and datasets
8   ...
9   eval/               # Scripts for evaluation of simulation results
10  analysis_plot.py   # Script for plotting analysis results
11  simulate.py        # Script for running simulations
12
13
14  ...
15  __init__.py         # Python file containing the main application logic
16  agent.py           # Handles the AI agent
17  card.py             # Card class for handling playing cards
18  deck.py             # Deck class for handling decks of cards
19  game.py              # Core game logic
20  graphics.py         # Graphics interface for the game
21  hand.py             # Hand class for handling cards in a hand
22  main.py             # Entry point of the program
23  train.py            # Training routines for the AI
24
25  .gitignore          # Specifies intentionally untracked files to ignore
26  environment.yml     # Conda environment file
27  LICENSE              # License details
28  literature_review.pdf # Background literature review document
29  README.md            # Project overview and instructions
```

04 METHODOLOGY

Utilized Q-learning, a model-free reinforcement learning algorithm, to train an AI on thousands of simulated Blackjack games. The AI adjusts its strategy by updating a Q-table based on game outcomes, learning optimal moves in various game scenarios.



05 DATA

Generated data from over 100,000 simulated games, capturing detailed records of game decisions, outcomes, and AI adjustments. This dataset provides a foundation for analyzing trends, AI learning progress, and strategy efficacy.



06 RESULTS

AI performance analysis shows significant improvement in strategy and increased win rates after extended training periods, as well as detailed outcome metrics validate the AI's capability to adapt and optimize gameplay strategies over time.

