**Blackjack AI: Enhancements & Improvements Report**

Final Project Enhancements Report

**1. Introduction**

*Project Overview*

This report summarizes the key enhancements and improvements made to the Blackjack AI project, focusing on new agent implementations, statistical analysis, and improved evaluation and visualization tools.

**2. Development Process & Prompt History**

*Initial Prompt*

Please help me revise my code to: 1. Implement a proper DQN approach by: - Exploring alternative network architectures (comparing different layers, activation functions) - Testing various training methods (epsilon-greedy exploration with different configuration parameters) - Including both deterministic and stochastic policy implementations 2. Add comprehensive baseline comparisons: - Implement a DQN baseline implementation - Create a fixed policy (deterministic) implementation - Add a table-based Q-learning approach (with approximately 730 state-action combinations) - Compare these approaches effectively 3. Enhance visualization and reporting: - Generate training curves showing episode steps vs. reward - Clearly define and implement reward function (r) and value function (Q) - Present Q as a DQN figure with appropriate visualization - Include comparative performance metrics across all implementations 4. Clean up code organization: - Modularize implementation components - Add comprehensive documentation - Include clear parameter descriptions - Create a results section with analysis of the different approaches When reviewing my code, please suggest specific implementations and optimizations that align with current best practices in DQN implementation.

*Iterative Enhancements*

The following improvements were made through iterative prompting and development:

* Deep Q-Network (DQN) agent implementation for advanced learning.
* Statistical analyses: deterministic baselines, Bernoulli trials, p-value calculations.
* RandomAgent for baseline comparison.
* Improved reward visualization: scatter plot for raw rewards, moving average line.
* Training curves and moving averages displayed in Jupyter notebook.
* Double down reward handling for fixed policy agent.

**3. Enhancements Made**

*Deep Q-Networks (DQN)*

A DQN agent was integrated, using a neural network to approximate Q-values and enable more complex decision-making.

*Statistical Analyses*

Deterministic baselines, Bernoulli trials, and p-value calculations were implemented to rigorously evaluate agent performance.

*RandomAgent for Baseline Comparison*

A RandomAgent was added to provide a baseline for comparison, choosing actions randomly.

*Improved Reward Visualization*

Raw rewards are now shown as a scatter plot, with a moving average line for trend clarity.

*Training Curves in Jupyter Notebook*

Training curves, including moving averages, are now visualized in the notebook.

*Double Down Rewards*

The fixed policy agent can now receive +2 or -2 rewards in specific double down scenarios, allowing for more nuanced analysis.

**4. Conclusion**

These enhancements have significantly improved the functionality and performance of the Blackjack AI project, providing a more comprehensive understanding of the agents' learning and decision-making processes.