Joseph Chavez
Leonardo Medrano
Matthew Jun

BlackJack with AI

Problem Description

We aim to develop an enhanced Blackjack game that incorporates AI to simulate a more challenging and realistic dealer and possibly other virtual players. The primary goal is to improve the traditional online Blackjack experience by introducing an AI opponent capable of adaptive strategy, rather than a predictable, static algorithm. This AI would analyze player behavior and adjust its strategy accordingly, providing a dynamic gaming experience that mirrors playing against a skilled human opponent.

Programming Language

We will use Python for this project. Python offers extensive libraries for AI and machine learning, such as TensorFlow, PyTorch, and scikit-learn, making it an ideal choice for developing complex AI models. Additionally, Python's simplicity and readability will facilitate rapid development and collaboration within our group.

Datasets

For training our AI, we might not require extensive external datasets since Blackjack's dynamics can be simulated internally to generate necessary training data. However, if we decide to incorporate machine learning models that require pre-existing game outcomes for learning, we could generate synthetic data by simulating thousands of Blackjack games or use publicly available datasets from online gambling sites.

Existing Code

We will start with existing open-source Blackjack game code, our primary extension will focus on integrating an AI module. This AI module will replace the traditional dealer logic with an intelligent system that learns from each game. Additionally, we may introduce an AI-driven virtual player option for single-player modes, enhancing the training and entertainment value of the game.

Algorithm/Approach

Our approach will involve reinforcement learning, specifically Q-learning or deep reinforcement learning, for developing the AI's decision-making capabilities. These methods are well-suited for environments like Blackjack, where decisions must be made with incomplete information, and the AI must learn from the outcomes of its actions.

Timeline

Week 1 Project Initialization: Set up the development environment, tools, and finalize the project plan. Begin developing the basic structure of the Blackjack game, including simple player interactions and game rules.

Week 2 Basic Game Framework: Complete the basic Blackjack game framework. Ensure that a simple version of the game can be played with a basic user interface.

Week 3 AI Module Development: Start with the development of the AI module. Implement a basic version of the reinforcement learning model that can make decisions based on the current game state.

Week 4 Integration: Integrate the AI module into the game, replacing the traditional dealer logic. Begin internal testing to identify initial issues.

Week 5 AI Training: Focus on training the AI model with simulated game data to improve its decision-making capabilities. Make necessary adjustments based on initial test results.

Week 6 Initial Game Testing: Continue AI training and start extensive game testing, focusing on AI performance and user experience. Begin iterating based on feedback.

Week 7 Final Testing: Conduct final testing, including beta testing with potential users if possible. Collect feedback and make final adjustments to the AI logic and game interface.

Week 8 Adjustments and Documentation: Finalize the project with any last-minute tweaks and complete the project documentation. Prepare for project presentation and potential release.

Computing Platform

We plan to develop and test the game on standard computing hardware (laptops/desktops with sufficient processing power). For training more complex deep learning models, we might utilize cloud-based GPUs (e.g., Google Colab or AWS EC2 instances) to accelerate the training process.

Roles and Responsibilities

Project Manager (Joseph Chavez): Oversees project progress, ensures milestones are met.

AI Specialist (Leonardo Medrano): Leads the development and training of the AI models.

Game Developer (Matthew Jun): Focuses on game mechanics, user interface, and integrating the AI model into the game.

Quality Assurance Tester (Joseph Chavez): Responsible for testing the game for bugs, user experience, and AI performance.

Documentation Specialist (Joseph Chavez): Manages project documentation, including the final report and user manuals.