Noobs Poker

A poker game simulator using Expictiminimax algorithm

Project Team

Dewan Fatin Hasnat(1921227642)

Ashfaq Hassan Shifat(2012889042)

Yeamin Mahmud(2011372042)

Group 11

Supervised by

M. Shifat-E-Rabbi

Department of Electrical and Computer Engineering

North South University
Bashundhara, Dhaka, Bangladesh

December, 2024

Introduction

Background and Motivation

Poker is one of the most popular card games worldwide, known for its blend of strategy, probability, and psychology. Among the various types of poker, **Texas Hold'em** is particularly prominent due to its widespread adoption in tournaments and online platforms.

Simulating poker provides a controlled environment for studying game theory, AI decision-making, and human behavior under uncertainty. This has made poker a common focus in fields such as artificial intelligence, machine learning, and computational game theory.

Python, with its simplicity and vast ecosystem of libraries, is an ideal language for building poker simulators. Its readability, combined with powerful libraries for simulation, statistical analysis, and machine learning, makes Python well-suited for developing robust poker AI systems.

2.1. Educational Purpose

- **Game Theory and Decision Making**: A poker simulator serves as an excellent tool for teaching concepts like **Nash equilibrium**, **bluffing strategies**, and **expected utility**.
- **Probability and Statistics**: Poker involves continuous probability assessments, such as calculating **pot odds** and **hand strength**, making it an engaging way to learn these principles.

2.2. AI and Machine Learning Research

- **Reinforcement Learning**: Poker is a benchmark problem for reinforcement learning algorithms, offering a challenging environment for training AI to adapt to uncertain and evolving scenarios.
- **Opponent Modeling**: Developing AI that can predict and adapt to human or AI opponents' behavior is a key aspect of poker simulation research, useful in broader AI applications like **autonomous decision systems** and **negotiation agents**.

2.3. Real-World Applications

- **Online Poker Platforms**: Simulators can be used to develop bots for online poker platforms, ensuring fair play by detecting anomalies or for training players.
- **Financial and Risk Management**: Poker strategies involve risk assessment and resource management, skills that are transferable to fields like **finance** and **project management**.

2.4. Entertainment and Gaming Industry

- **Game Development**: Poker simulators can be integrated into video games, mobile apps, or virtual reality experiences.
- **Player Training**: Advanced simulators offer an environment for players to hone their skills, test strategies, and analyze their performance without the risk of losing real money.

Platforms and Diagrams

System Design

The system design for a Poker Game Simulator using the ExpectiMiniMax algorithm is structured to handle the complexities of poker decision-making by combining modular components for efficient game management and intelligent AI behavior. At the core of the system is the Game Engine, which manages the state of the game, including player actions, betting rounds, and phase transitions (Pre-flop, Flop, Turn, and River). It interfaces with a User Interface Layer, which can be implemented using graphical libraries such as PyQt, Tkinter, or PyGame, or via a command-line interface (CLI) for simplified interaction. This layer allows players to make decisions and visualize the current game state, including card hands, community cards, and pot size.

The heart of the simulator lies in the AI Decision Engine, where the ExpectiMiniMax algorithm is implemented. This engine constructs a game tree representing all possible game states, considering both player decisions (Max nodes), opponent responses (Min nodes), and probabilistic events like card draws (Chance nodes). The ExpectiMiniMax algorithm recursively evaluates this tree to determine the optimal action by balancing expected rewards and minimizing potential risks. The AI also incorporates an Opponent Modeling module, enabling it to adapt dynamically to player behaviors by analyzing historical actions, thereby refining its strategy over time.

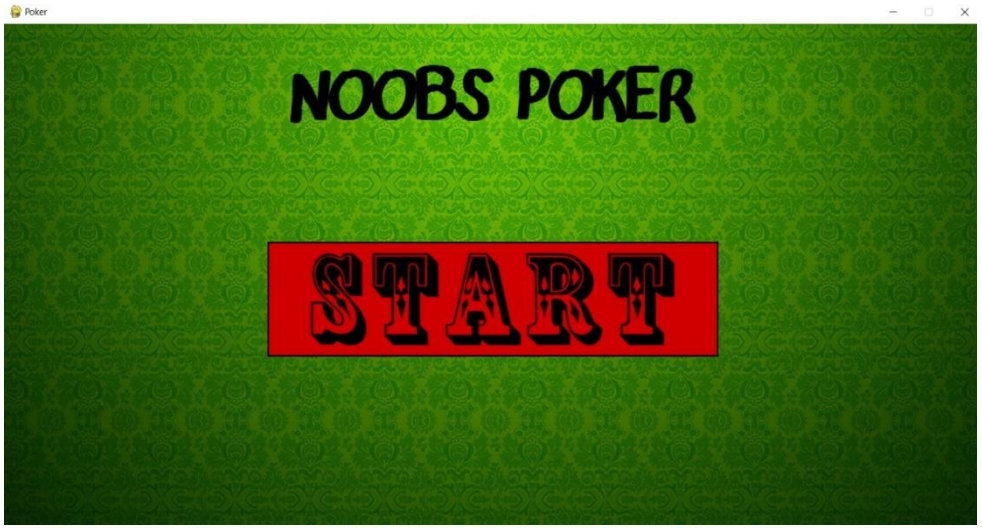
Supporting this system are several Core Modules, including the Card Handling Module for shuffling and dealing cards, a Hand Evaluator for ranking poker hands, and a Probability Calculator to compute pot odds and hand improvement probabilities. All gameplay data, such as player profiles and decision logs, are stored in the Data Layer, which can be used for post-game analysis, training, and refinement of AI strategies. This modular design ensures the simulator is scalable, allowing for future enhancements like multiplayer functionality, additional poker variants, and more sophisticated AI models, making it a versatile tool for both educational and research purposes.

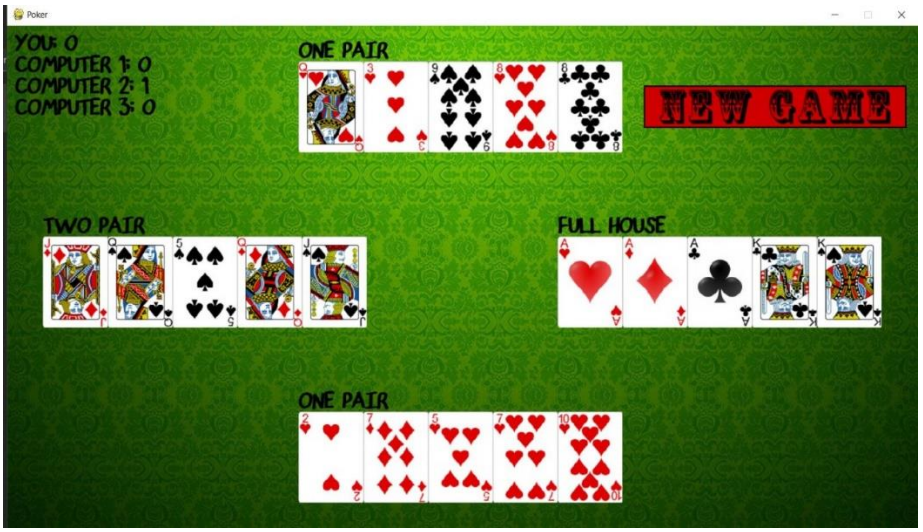
Expectiminax algorithm: The **expectiminimax algorithm** is an extension of the minimax algorithm, commonly used in decision-making for games involving both deterministic and stochastic elements. While the minimax algorithm assumes two players with opposing goals, expectiminimax incorporates a third type of node, known as chance nodes, which represent random events such as dice rolls or card draws. The algorithm operates by traversing a game tree consisting of decision nodes (representing a player's move), chance nodes (representing random outcomes), and terminal nodes (representing the end of the game with a utility value). At decision nodes, the algorithm selects the move that maximizes or minimizes the player's expected utility, depending on whether it is a maximizing or minimizing player's turn. At chance nodes, it calculates the expected value by averaging the outcomes weighted by their probabilities. This allows the algorithm to handle games where luck plays a role, making it ideal for games like backgammon or poker. However, expectiminimax is computationally expensive due to its need to evaluate both strategic choices and probabilistic outcomes.

Hardware and/or Software Components

- Software used: Python, VScode
- Libraries used: Pygame, Os, Sys, Random, Operator

Overview




Poker
YOU: 0
COMPUTER 1: 0
COMPUTER 2: 1
COMPUTER 3: 0
ONE PAIR
NEW GAME
TWO PAIR
FULL HOUSE
ONE PAIR


Poker
YOU: 0
COMPUTER 1: 1
COMPUTER 2: 1
COMPUTER 3: 0
ONE PAIR
NEW GAME
FULL HOUSE
THREE OF A KIND
HIGH CARD

Impacts of the Project

A **poker game simulator** can have several impactful outcomes, both in terms of learning and practical applications. Here are some of the key impacts:

1. Enhancement of AI and Machine Learning Models

A poker simulator can serve as a valuable tool for developing and testing AI models. It allows researchers to experiment with different strategies, algorithms (like **expectiminimax** or reinforcement learning), and decision-making processes in a controlled environment. This can lead to advancements in AI's ability to handle complex scenarios involving both strategic thinking and probabilistic elements.

2. Educational Tool for Learning Poker Strategies

A poker simulator provides an interactive way for users to learn the game, improve their strategies, and practice decision-making in various poker variants (e.g., Texas Hold'em, Omaha). It can simulate multiple opponents, different styles of play, and teach players how to manage risk, bluff, and calculate odds in a safe environment without financial stakes.

3. Improvement of Human-Computer Interaction

By integrating advanced AI into the simulator, users can experience more realistic poker opponents. This helps improve the design and testing of AI systems in human-computer interaction (HCI), offering insights into making AI behavior more human-like, challenging, and engaging for users.

4. Application in Gambling and Risk Management

In the context of online gambling or casino management, poker simulators can be used to analyze player behavior, predict outcomes, and design systems that ensure fair play and responsible gaming. It can also assist in the development of more effective gambling analytics, such as detecting patterns of cheating or optimizing game strategies.

5. Practical Application in Decision-Making

The strategic and probabilistic decision-making involved in poker simulates real-world situations like stock trading, resource management, and other areas where decisions must account for both known factors and uncertainty. This can make the simulator a valuable tool for training professionals in these fields, helping them understand and manage risk in dynamic environments.

Conclusions

Summary

A poker game simulator is a software application designed to simulate the experience of playing poker, either against AI opponents or other players. It provides a controlled environment where users can practice poker strategies, learn the rules, and improve decision-making skills without financial stakes. The simulator can mimic various poker variants, such as Texas Hold'em or Omaha, and includes features like customizable gameplay, multiple difficulty levels, and the ability to simulate different types of opponents (AI or human).

Key benefits of a poker game simulator include:

- **Educational Value**: Helps users learn poker rules, strategies, and probability calculations in a risk-free environment.
- **AI and Strategy Development**: Useful for testing and developing poker AI, including decision-making algorithms like **expectiminimax** or **reinforcement learning**.
- **Realistic Gameplay**: Provides a realistic and engaging experience for players, enhancing both single-player and multiplayer experiences.
- **Application in Research**: Assists researchers and developers in studying AI behavior, decision-making under uncertainty, and human-computer interaction.
- **Practical Use in Gambling**: In gambling and risk management contexts, simulators can optimize game design, promote responsible gaming, and improve fairness in poker games.

Overall, a poker game simulator serves as both an educational tool and a practical application for refining poker strategies, developing AI models, and providing entertainment.

Limitations

While a **poker game simulator** offers many benefits, it also has some limitations that can affect its effectiveness and realism. Here are some of the key limitations:

1. Lack of True Human Behavior

- **AI Limitations**: While AI opponents can be programmed to mimic certain behaviors, they may still lack the unpredictability and subtle nuances of human behavior, such as emotional decision-making or bluffing tactics. This can make the simulator feel less authentic for players seeking a truly human-like experience.

- **Predictability**: AI-controlled opponents might follow deterministic strategies, making it easier for skilled players to predict their moves, which reduces the challenge.

2. Limited Randomness

- **Simplified Randomness Models**: Many simulators might use overly simplified randomness in dealing cards or determining outcomes. Real-world randomness, such as psychological factors or player tendencies, might not be fully captured, which can impact the simulator's realism.
- **Inaccurate Odds Representation**: The randomness of poker hands and betting patterns may not always be fully representative of real-life scenarios, which can limit its ability to accurately prepare players for actual poker games.

Future Improvement Plans

Improving a **poker game simulator** can enhance both its realism and usability, allowing users to get more out of their training and entertainment experience. Here are some potential areas for future improvements:

1. Advanced AI and Adaptive Behavior

- **Improved Opponent AI**: Future simulators could incorporate more advanced AI techniques, such as deep learning or reinforcement learning, to create opponents that better mimic human behavior. AI could learn from players' strategies, adapt to different styles of play, and become more unpredictable in terms of bluffing and decision-making.
- **Psychological Modeling**: AI could be designed to simulate more human-like traits, including bluffing, emotional decision-making, or reactions to stress, adding a layer of realism to the game. This could involve incorporating elements like tilt (emotional frustration) and psychological cues, which human players use to gain an advantage.

2. Better Realism and Randomness

- **Enhanced Randomness Models**: Simulators could improve their randomness models to better reflect real-world outcomes, accounting for factors like player tendencies, emotions, and social dynamics. This would increase the unpredictability of the game, making it feel more like a live poker game.
- **Dynamic Game Environments**: Simulators could evolve to include more varied environments that mimic real-world poker settings—different casinos, online poker rooms, and even home games—with specific rules, player profiles, and styles.

Bibliography

H. Berliner, 1979. “The B* Tree Search Algorithm: A Best First proof Procedure”, Artificial Intelligence, vol. 12, no. 1, pp. 23-40.

H. Berliner, G. Goetsch, M. Campbell and C. Ebeling, 1990. “Measuring the Performance Potential of Chess Programs”, Artificial Intelligence, vol. 43. no. 1, pp. 7-20.

D. Billings, D. Papp, J. Schaeffer and D. Szafron, 1998a. “Poker as a Testbed for Machine Intelligence Research”, in AI’98 Advances in Artificial Intelligence (R. Mercer and E. Neufeld, eds.), Springer Verlag, pp. 1-15.

D. Billings, D. Papp, J. Schaeffer and D. Szafron, 1998b. “Opponent Modeling in Poker”, AAAI, pp. 493-499.

D. Billings, D. Papp, L. Peña, J. Schaeffer and D. Szafron, 1999. “Using Selective-Sampling Simulations in Poker”, AAAI Spring Symposium.

R. Fung and K. Chang, 1989. “Weighting and Integrating Evidence for Stochastic Simulation in Bayesian Networks”, Uncertainty in Artificial Intelligence, Morgan Kaufmann.

M. Ginsberg, 1999. “GIB: Steps Towards an Expert Level Bridge-Playing Program”, IJCAI, to appear.

D. Koller and A. Pfeffer, 1997. “Representations and Solutions for Game-Theoretic Problems,” Artificial Intelligence, vol. 94, no. 1-2, pp. 167-215.