Poker Project AI

Agent ID - 15

Shankaranarayanan B R

Sachin Bharadwaj B A

Abstract:

The agent which was implement is a reflex agent where in the agent would investigate opponent's player action and react accordingly. For the opening action we have taken the initial bet of the player from the condition and if the Bet value of less than or minimum chips left in the pot it will perform do the **CHECK** action, when the bet amount of greater than the minimum chip left the player is pushed to bet all the money in the pot(**ALLIN**), all this based on the Card rank and the score which was assigned to them.

For determining the call or raise or action we are considering the balance chips present in the players pocket and the player's aggressivity style and by assigning score to each of the Hands which is present with the player and the cards thrown.

Introduction:

Poker is generally a card game, with 52 deck cards, where players use their card combination or hand to bet some money, by placing their bet in the central pot. Each hand has a value and the winner is declared with the hand of highest value according to the given hand ranking hierarchy, according to the game rules. The player also wins the current bet when the rest have folded. Here in our project, we play AI poker game where we implement a python code for the agent to play the game. We need a server and two clients to play the game. The server allows the clients to connect to it and when the clients are connected the game starts. The server controls the game. The server also sends query to the client when a decision must be made. A maximum of 5 clients can be connected and played against each other

Rules:

The game starts by forcing each player set an Ante (a forced bet). Once the bet has been set, each player will receive fire cards from the server. When it is a player's turn to act, the first action he takes binds him to his choice of action and changing his action after seeing how other players react. Once each player has received five cards the round can start. The first player can then choose to make one of the following specific actions:

Open: This is where the player makes the first voluntary bet in the betting round. It can also be referred to as opening the pot.

Call: This is referred to a bet or a "raise".

Check: This is when a player may pass or check, when no one has opened the betting round yet. This is equivalent to calling a current bet or a zero. When checking, a player declines to make a bet; this indicates that he does not wish to open but does wish to keep his cards and retain the right to call or raise later in the same round if an opponent opens.

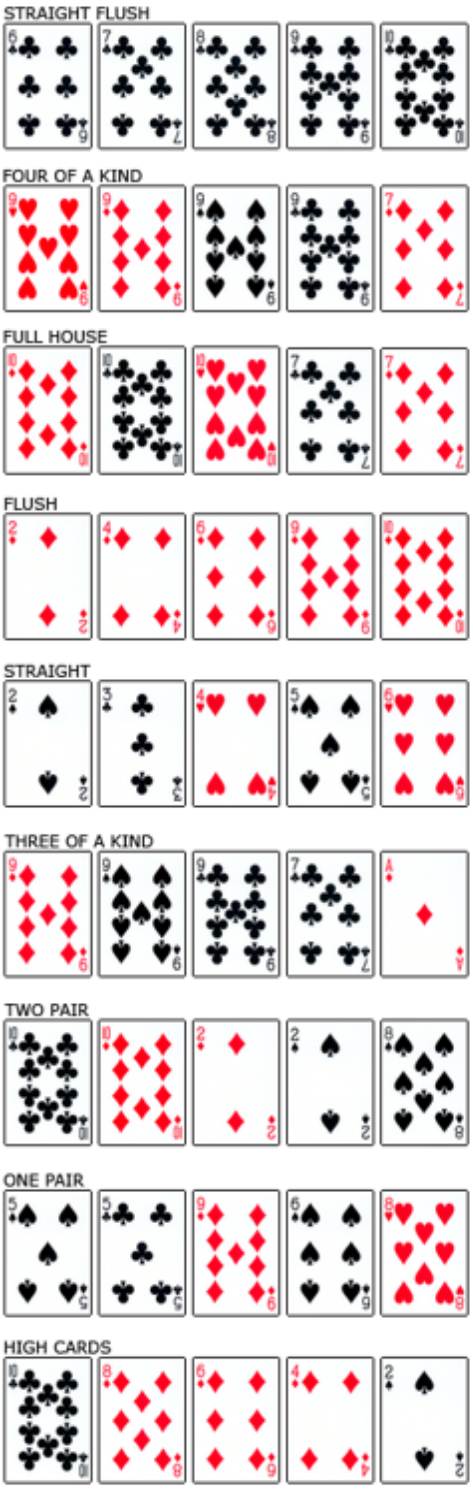
Raise: This means increasing the bet in the pot. This forces players to call in new amount. It can be considered as opening bet if the current bet is nothing. If the player makes a second raise in the betting round, it is called as re-raise. If, due to an open or raise action, a bet has been placed that the player in-turn cannot match, then unless that player chooses to go all-in, he must fold.

Fold: Fold means the player discards his current hand and ignores the current pot. The player cannot bet further, and he cannot win.

Ante: This is also called forced bet. Every player plays equal amount into the pot before the dealing begins.

Hands:

In poker game, there exists hands of five cards according to the prespecified game rules. Using standard ranking system, the hands are compared and the player with the highest hand is declared the winner. The strength of a hand is increased by having multiple cards of the same rank, all the cards being from the same suit, or having all the cards with consecutive values.



There are 311,875,200 ways ("permutations") of being dealt five cards from a 52-card deck, but since the order of cards does not matter there are 2 598 960 possible distinct hands ("combinations").

There are 40 possible straight flushes, including the four Royal Flushes. The probability of being dealt a straight flush is 40 / 2 598 960 = 0.0015%.

There are 624 possible hands including four of a kind; the probability of being dealt one is 0.024%.

There are 3,744 possible full houses; the probability of being dealt one in a five-card hand is 0.14%.

There are 5,148 possible flushes, of which 40 are also straight flushes; the probability of being dealt a flush in a five-card hand is 0.20%.

There are 10,240 possible straights, of which 40 are also straight flushes; the probability of being dealt a straight in a five-card hand is 0.39%.

There are 54,912 possible three of a kind hands which are not also full houses; the probability of being dealt one in a five-card hand is 2.1%.

There are 123,552 possible two pair hands that are not also full houses; the probability of being dealt one in a five-card hand is 4.75%.

There are 1,098,240 possible one pair hands; the probability of being dealt one in a five-card hand is 42.26%.

Of the 2,598,960 possible hands, 1,302,540 do not contain any pairs and are neither straights nor flushes. As such, the probability of being dealt "no pair" in a five-card hand is 50.12%.

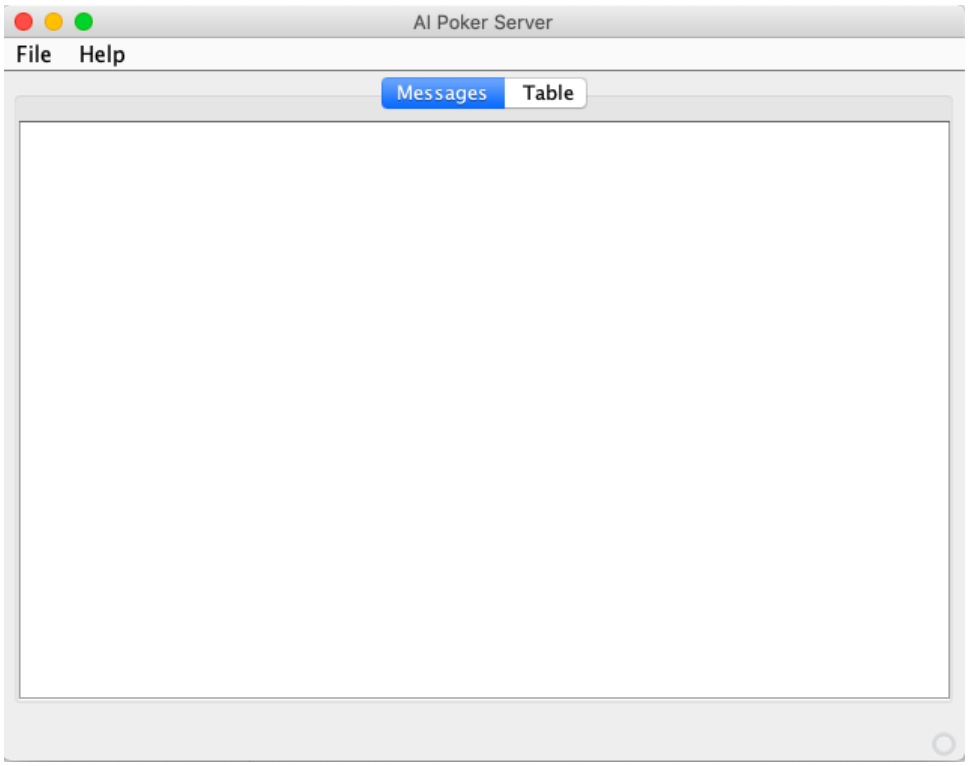
http://en.wikipedia.org/wiki/List_of_poker_hands

Settings:

In order to run the Poker application,we must start the Poker server jar which is present in the project folder. The jar file is present in the PokerServerGUI folder. It can be started by using the following command in command line/terminal

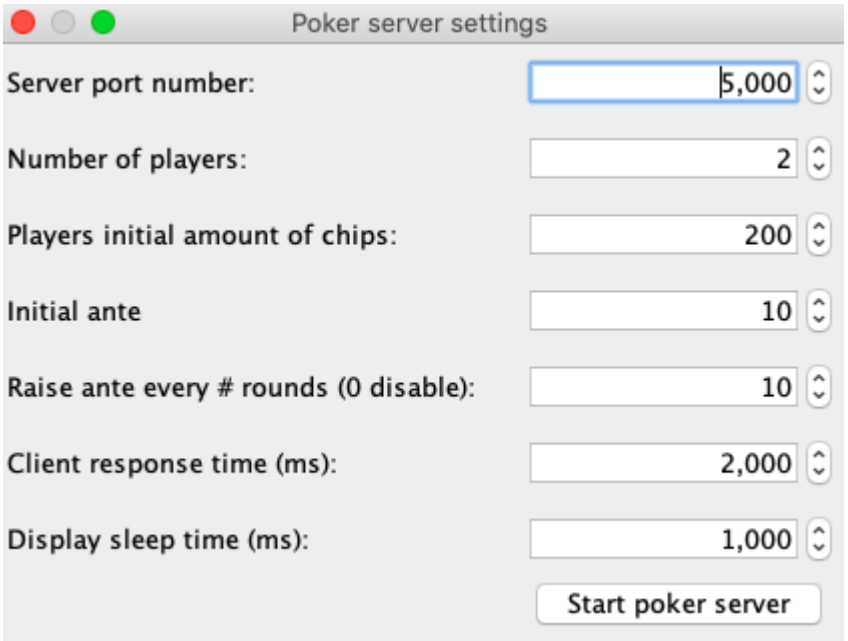
```
java -jar PokerServerGui.jar
```

Following which we will see the following window in the screen.



Follow the below instructions:

1. Click on the File option and select Start Poker Server
2. It will display the following popup



3. Click “Start poker server” button present in the popup.
4. In order to run client file in the server given we need to do run the following command

```
python ./PokerGame.py
```

Types of agents:

The following agents can be implemented in the poker game.

Rational agent: Where the player bids based on what it feels true at that particular moment.

Fixed agent: Where the player bids the fixed amount in each betting round irrespective of the hand he has in his possession.

Reflex agent: Where the player bids in a reflex manner i.e. bidding based on the hand strength. It is a significant agent compared to the above agents.

Learning agent: Where the player bids by the experience of his actions. Also called as memory agent. perform training in an online manner and can deal with streaming data.

Methods:

Several AI methods or strategies can be implemented for the poker game in order to maintain the gameplay efficiently. In our project we use significant AI methods to write and implement the code. The following are the methods used to build the game.

1. Opening Strategy.
2. Betting Strategy.
3. Final Showdown Phase.

These three strategies are mainly used to implement the gaming strategy in the project and writing the client file. In general, the opening strategy deals with the initial voluntary bet from a random player and this leads to opening of the game. The other players continue with the play after this. This is called betting strategy. This is the strategy where the type of agents come into play. The betting strategy is more efficient based on the type of agent we implement in the game. Better the agent, better is the strategy of betting. Finally, the showdown phase comes into play where the agent concludes the game with final showdown of his hand. It depends on the agent to make the action unless he has no amount left.

In order for the writing the opening, betting and showdown strategy methods we had written supporting classes and method in the project, such as Cardrank , Calcstyle and Player classes in the project folder. The Players file is used to write the all the actions which the player can perform, the Card Rank file has the rank of different hands.

PEAS Description:

PEAS stand for Performance, Environment, Actuators and Sensors. They can be collectively defined or differentiated based on the type of agent we use in the project.

Performance:

Performance is the output obtained by agent. All types of results yielded by the agent after processing comes under performance. In AI poker, result, winner declaration, final amount in the pot and number of dealings are some of the performance factors taken into consideration.

Environment:

Environment of an agent is the surrounding conditions and things of that agent. It is basically the scenario under which the agent works. In AI poker game, the server platform, type of opponent agent and type of dealer are some environment conditions under which our agent is implemented.

Actuators:

The device/criteria through which the agent performs any action or process any information to produce a specific result is generally called actuator. In AI poker, the agent performs any action based on the card rank/hand strength, betting strategies and card dealing in each round. Some agents like learning agent learns the strength of the opponent hand and performs the action based on that.

Sensors:

The devices/phenomena through which an agent observes and perceive its environment is generally called sensor. In AI poker, the game is controlled by the server. Hence the inputs from the server and server command GUI acts as sensor devices in the output game.

Expected Behaviour:

The general observation from the reflex agent (the agent which we implemented in our project) is that it performs action based on hand strength or card rank. Therefore, the common expected behaviour from the respective agent is playing smartly based on the hand and winning by a good margin. And, the expectation from our agent is high against random and fixed agents and with a greater number of wins. There is always one winner and one loser when two agents play against each other.

Experiment and Tournament result:

On playing with agent which was developed by us we can see that agent does provide a winner and loser in the game and we it does provide the number of Chips won by each player for each round. The result does match the expectations we are getting the result in all the rounds and the server declares the winner in all the rounds are over.

Since we did not attend the pre-tournament, I had setup a small tournament with agent 19 and our agent had lost with them. The agent lost due to the low ranking of the Cardrank and agent had gone Allin due to the unavailability of any bets.

Conclusion:

In this project, the exposure to AI poker game was done where different types of agents were studied and implemented in the source code. Reflex agent was implemented in this project where the agent takes into the account the hand strength/card rank and then performs the action. The actions were written into different class files and client file was executed to play the game. The output message was read on the server window and the winner was recorded and stored. The project efficiently gave an opportunity to observe how agents can be implemented in the game and executed later for better results.