FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

A Profitable Online Poker Agent

João Campos

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Master in Informatics and Computing Engineering

Supervisor: Henrique Lopes Cardoso (Doctoral Degree)

Co-Supervisor: Luís Teófilo (Master's Degree)

June of 2013

"*You have to understand the rules of the game. And then you have to play better than anyone else*" Albert Einstein

A Profitable Online Poker Agent

João Campos

Master in Informatics and Computing Engineering

June 2013

Abstract

Games of incomplete information, such as poker, are a continuous source of research and study in the area of artificial intelligence. Poker presents challenging problems such as opponent modeling, risk management and bluff[1] detection. The development of agents capable of probabilistic calculations considering those problems is considered to be difficult to achieve, since dynamic adaption is required in order to create a robust computer poker player. This thesis focuses on the development of a poker agent able to play against human players and aiming to achieve the dynamic adaptation needed to beat some human players online. This will be achieved by using some sets of information about each player the agent plays against. Using Holdem Manager, a tool that registers the hands played in an online poker room; it is possible to obtain statistics about every player the agent is playing against. The agent is able to explore some of these statistics so that it can better decide on which action to take. Some factors like how aggressive an opponent is, the position held at the table, how many players are involved, how much money is involved, and the hand dealt to the agent are a few portions of the information sets used to compute the agent's behavior. This agent was developed based on a short-stack[2] strategy, and through the use of the sets of information provided by the Holdem Manager. For the first time in the Computer Poker literature, results on online Poker agent games versus human players in a controlled environment are presented, and without the players being aware their opponent was a computer agent. The agent is able to play live online poker versus human players, and presents a small profit in the No-Limit Texas Hold'em poker game at micro stakes[3], namely 0.02 and 0.01 cents.

[1] The act of deceiving other players by betting strong and making believe our own hand is strong when it is not
[2] A strategy that implies entering a poker table with the minimum possible money, and mostly playing pre-flop
[3] Term used in poker to refer to the maximum amount of money played at a certain table

Resumo

Jogos de informação incompleta tais como poker são uma fonte contínua de estudo e pesquisa no âmbito da inteligência artificial. No poker problemas como: modelação de oponentes; gestão de riscos e detecção de bluffs[4] representam um desafio. O desenvolvimento de agentes capazes de considerar esses problemas e realizar cálculos probabilísticos é considerado como uma tarefa árdua de se realizar, uma vez que é exigida uma adaptação dinâmica para que seja criado um agente de poker robusto. Esta tese irá focar-se no desenvolvimento de um agente de poker capaz de jogar contra jogadores humanos e alcançar a adaptação dinâmica necessária para superar alguns jogadores humanos de poker online. Algo que será possível usando um conjunto de informações sobre cada jogador que o agente enfrenta. Utilizando como auxílio o Holdem Manager, uma ferramenta que regista mãos jogadas em salas de poker online, é possível obter estatísticas sobre todos os jogadores que o agente enfrenta nas mesas. O agente é capaz de explorar algumas destas estatísticas de maneira que possa decidir melhor sobre a acção a tomar. Alguns factores como quão agressivo é um adversário, a posição ocupada na mesa, quantos jogadores estão envolvidos, quanto dinheiro está em causa, e o par de cartas que o agente recebe são uma pequena porção do conjunto de informações utilizadas na determinação do comportamento do agente. Este agente foi desenvolvido baseando-se numa estratégia "*short stack*"[5], e modelando adversários com o auxílio do conjunto de informações reunido através do Holdem Manager. Pela primeira vez na literatura do *Computer Poker*, são apresentados resultados de jogos de poker online, num ambiente controlado, contra jogadores humanos sem estes saberem que estão em jogo contra um agente. O agente é capaz de jogar poker online ao vivo contra jogadores humanos, e apresenta um pequeno lucro na vertente Texas Hold'em em micro limites[6] de apostas, nomeadamente 0.01 e 0.02 cêntimos.

[4] Acção de iludir adversários ao apostar uma mão que não é forte de modo a induzir desistência
[5] Estratégia que implica entrar numa mesa de poker com o mínimo valor possível e jogar maioritariamente pre-flop
[6] Termo utilizado no poker referente à máxima quantidade de dinheiro jogada numa determinada mesa

Acknowledgements

I would like to thank Michel Dattani, a professional poker player, for helping with valuable information and knowledge about the game of poker.

Thanks to Luís Teófilo, my co-supervisor, a researcher and enthusiast on the development of poker agents. His research and code developments were very helpful for the agent's strategy definition.

Thanks to Henrique Lopes Cardoso, my supervisor, for the support and availability shown throughout this work.

I would like to thank my family, friends and girlfriend for all the support given.

João Campos

Table of Contents

1.Introduction 21

1.1 Context 21
1.2 Motivation 22
1.3 Objectives 22
1.4 Thesis structure 23

2.State of the art 25

2.1 Opponent Modeling 25
2.1.1 Collecting Opponents' Information 27
2.2 Simulation Software 29
2.3 Game State Recognition 31
2.3.1 Image Processing Software 31
2.4 Agents Developed 33
2.4.1 Experiments 34
2.5 Lokibot 34
2.5.1 Experiments 35
2.6 Tools 36
2.6.1 Holdem Manager Software 36
2.7 Conclusions 38

3.Background 39

3.1 Poker Game 39
3.1.1 Table Positioning 39
3.1.2 Rules 40
3.1.3 Hand Ranks 41
3.1.4 Advanced Poker Domain Knowledge 42
3.2 Summary and Conclusions 45

4.Agent Implementation 47

4.1 Programming Language 47
4.2 Application Struture 47
4.3 Agent Development 49
4.3.1 Strategy Definition 50

4.3.2 Game State Recognition 51
4.4 Database Communication 53
4.4.1 SQL Queries 54
4.5 Conclusions 55

5.Experiments and Results 56
5.1 Experiments 56
5.2 Results 57
5.2.1 All time results....57
5.2.2 Agent’s Statistical Information 59
5.2.3 Results from Each Position at Table 61
5.2.4 Results from Attempting to Steal Blinds....61
5.2.5 Results from Attempting to Defend Blinds....62
5.2.6 Results for Top 10 Most Profitable and Unprofitable Players....64

6.Conclusions and Future Work 68
6.1 Conclusions 68
6.2 Future work 68

References....70

Appendix A....72

List of Figures

Figure 1 – Player's classification 25
Figure 2 – Player's nicknames by play style 27
Figure 3 - HoldemML FrameWork 29
Figure 4 - LIACC Poker Simulator Architecture 30
Figure 5 - OpenHold'em 31
Figure 6 – Letter T 32
Figure 7 – Player actions 32
Figure 8 – Output text sample 33
Figure 9 – Average Bankroll History 35
Figure 10 – Holdem Manager software 37
Figure 11 – Player HUD statistics 37
Figure 12 – Table positions 40
Figure 13 – Betting rounds 41
Figure 14 – Hand strengths 41
Figure 15 – Follow chart 48
Figure 16 – Small Blind color detection 51
Figure 17 – Small blind color detection 52
Figure 18 – Copying game console chat into clipboard 52
Figure 19 - Agent's Architecture 54
Figure 20 – All time results 57
Figure 21 – General statistics and expected values 59
Figure 22 – Pre-flop hand filter 60
Figure 23 – Statistical information with filtering from figure 17 60
Figure 24 – Winnings from steal attempts 62
Figure 25 – Winnings from attempting to defend the blinds 63

List of Tables

Table 1 – Vexbot agains other bots (small bets per game) 34

Table 2 – Results by position 61

Table 3 – Results from steal attempts 62

Table 4 – Results from attempting to defend the blinds 63

Table 5 – Top 10 most profitable human players played against 64

Table 6 – Top 10 most unprofitable human players played against 64

Table 7 – Agent versus Player1 65

Table 8 – Agent versus Player2 65

Table 9 – Agent versus Player3 66

Table 10 – Agent versus Player4 66

Table 11 – Agent versus Player5 66

Abbreviations and Symbols

ALL-IN	Action of betting all money
BB	Big blind position
BTN	Button position
CO	Cut-off position
EP	Early position
MP	Middle position
NL	No limit
PFR	Pre-flop raise
SB	Small blind position
VPIP	Voluntarily money put in pot

Chapter 1

Introduction

Computing strategies for games of incomplete information is still a challenge for artificial intelligence research. Poker as a multiplayer stochastic game represents one of games where the strategy definition turns into a most challenging problem. Poker has an enormous variation of strategy types, and since these strategies don't rely only on the game state but on incomplete game states, because of hidden cards, the best strategy is to adapt, recognizing weaknesses, and exploiting them. The amount of information to be taken into account is a challenging computational problem because the hidden information creates a very large decision space, and there is no more than a few seconds to make a decision while playing online poker.

The Texas Hold'em Poker variant rose in popularity in a small period of time mostly because of the broadcasting of the World Series of Poker. People became more interested in the game, where a small investment could mean a huge increase in wealth. Information about the poker game became available in the internet for players that wanted to improve their strategies. Game strategies and videos of professional players were easily found and helped the growth of the game.

Several methodologies already exist and were developed for agents to play Poker, but none has been consistently tested against human players in an uncontrolled environment. This thesis describes the development of a computer Poker player that aims to win against human players in real money tables in a live environment, an online poker room. Regarding the bot usage at online poker rooms, there are some rooms that allow the use of programs to aid the players, and some state that it is an unfair advantage and thus do not allow such programs. All poker rooms stand united when stating that accessing encrypted information traded between the room and its server is not allowed, since it would lead to opponents' cards being known, and turn all hands profitable. In the legal terms, since there is not a law that regulates online poker yet, the only drawback from developing an agent able to play online poker would be having its earnings confiscated.

1.1 Context

Researching games of incomplete information brings a challenging problem in the field of artificial intelligence. As a game of incomplete information poker became target for multiple researches, since the outcome of each play cannot be predicted accurately. Although some probabilistic measures can be made for certain information sets, hand values and player strategies, and these probabilities will have a significant impact after a large amount of events. In the field of artificial intelligence systems like Bayes' theorem, Nash equilibrium or Monte Carlo samplings, which rely on probabilistic distribution, are the most common approaches in incomplete information games. These systems will be essential in determining the probable outcome of different actions while playing poker, by computing the large amount of information sets until they become a significant value to take into account. Finding the best computer poker player means developing the best adapting strategy, taking into account as many sets of information as possible. Besides all probabilistic calculations, behavioral changes occur among human players and they must also be taken into account as well. The most challenging aspect of the development of poker agents is giving them the ability to adjust to their opponents' strategies.

1.2 Motivation

The motivation for the development of this project is the challenge of developing an agent capable of playing online poker against human players, adapting its strategy according to the information available to it. The ability to recognize patterns, and exploit them, is a valuable asset and essential on a winning player. Developing an agent capable of adapting according to different sets of information is interesting and a matter to study for the artificial intelligence field.

The innovation factor in this project would be testing the agent in a real online environment where human players are experienced poker players. Playing with real money provides results much more accurate when evaluating the agent's performance against human players.

1.3 Objectives

The most obvious objective for this work is the possibility of this agent making a profit against human players, proving its superiority and ability to outplay them. The best way to achieve this would necessarily mean the implementation of a methodology which allows the development and selection of strategies accurately. A rather complex process that encompasses:

- Classify Poker incomplete game states;
- Obtain strategies through opponent modeling;
- Create a platform that allows testing Poker agents at online tables;
- Development of a strategy selection algorithm;

- Create an agent that uses a short-stack strategy to validate the platform.

1.4 Thesis structure

This thesis is divided in five chapters, ordered for a better understanding of the concepts and studies made.

The first chapter contains a brief introduction of the subject studied in this thesis as well as its context, motivation and objectives.

Following chapter one is chapter two, the state of the art. This chapter describes existent research performed on this thesis domain, algorithms developed, useful tools and software.

Chapter three contains some domain information, such as the poker game, rules and game play. There is also the differentiation of players and their categorization into different behaviors according to their play style. Finally there is also a highlight on expert poker terms and situations to be aware of.

Chapter four is all about implementation, and here it can be found how the project was built, the steps took to develop the agent, and its entire structure. Conclusions will state the difficulties encountered and what could have been made differently.

Chapter five presents the performed experiments as well as the obtained results. The results consist of interpreting the agent's performance using the hand histories and graphical representation of all hands and winnings of the agent. Finally, some conclusions are taken of the results.

Chapter six is the ending chapter containing the conclusion of this work and some suggestions for future work.

Chapter 2

State of the art

In this chapter it will be described the state of the art and related work will be presented in order to demonstrate what has been already developed and which are the main problems found. The main aspects to be developed and opportunities for improvements will also be stated here.

This chapter shall contain a technological review according to the main tools usable in the scope of the project, justifying future choices.

2.1 Opponent Modeling

Players can be classified by their playing style; this usually happens when using software that registers hands while playing online poker. Their behavior will be continuously registered by a database of hands, so it is easy to point out which players are playing a large amount of hands, in what position and how aggressively. This can be very useful in the matter of deciding how to play our own hand. Their classification can be narrowed to these four ranges (figure 1):

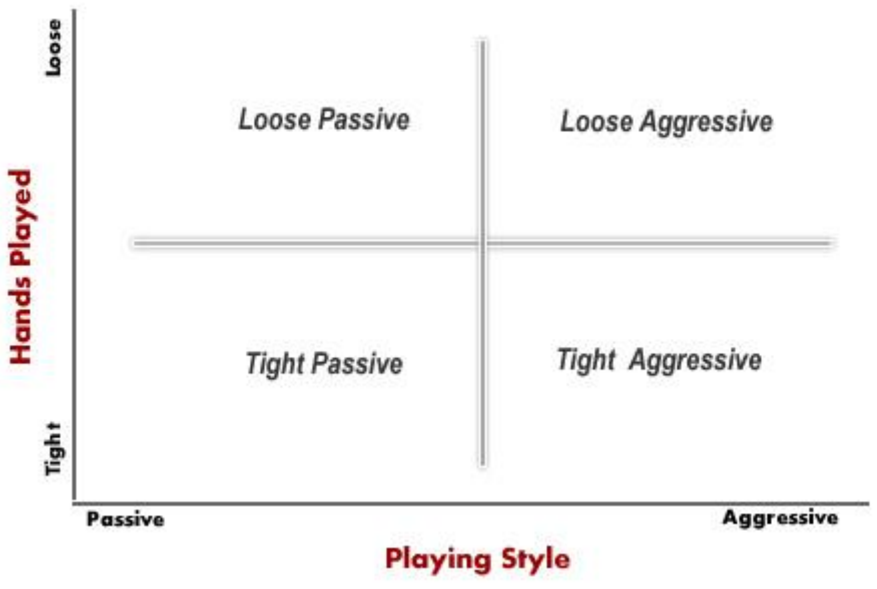


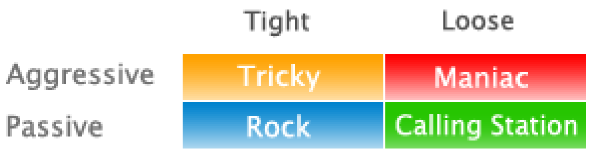
Figure 1 – Player's classification

A player that usually plays a large amount of hands, so this will mean he will check or call very often until the river card, and will not raise or fold that frequently. The best way to play against these opponents is betting with a good hand. Value bet is a term used when our hand is the best hand or good enough against our opponent's range. Facing this kind of player value bet should be very profitable since they won't try to bluff. Bluffing loose passive players is not recommended since they tend not to fold very often, and are always curious to see the next card coming. They are most probably the easiest players to explore in the poker game, since our actions are only guided by our own hand strength, and if they bet or raise it most likely means their hand has some strength;

- **Loose aggressive player (LAG):**
  A player that usually plays a large amount of hands like the LPA but instead of check or calling very often, he will bet or raise instead. The best way to play against this kind of players is simply not trying to bluff, because these players don't usually fold and will often raise. By playing tight, meaning the top high value cards, it is only a matter of time until the hand we hold will make a pair or more on the flop, and against a LAG a pair with the highest value card is usually better than what he is holding. Since his weakness is playing almost every two cards he receives, after having a made hand, the best action is to bet and stick with the hand, not folding to him, since that's what he will always try to do. He will always bet trying to make believe he has that last card to complete the sequence or flush missing at the table. The common mistake against these players is usually folding the best hand to him, because he bet extremely high and succeeded in intimidating. This kind of players is considered to be the most profitable ones to play against;

- **Tight passive player (TPA):**
  A player that plays a small amount of hands but will not bluff very often making him a passive player that will most likely check or fold when he should bet or raise. They tend to fail into capitalize the strength of their hands. The best way to play against these players is trying to bluff by betting when they check, since they don't like risks, they check when they don't complete a hand a bet when they do. Therefore they don't usually bluff. These players are quite easy to play against since their hand can be read, and it is predictable by their actions;

- **Tight aggressive player (TAG):**
  A player that plays a small amount of hands and is very aggressive, he will capitalize on his hand strength or his opponent's fear or weakness. This player will often bet and raise, while playing strong hands, maximizing the amount and the number of times he wins. This is a winning poker player so there is not a best way to play against them, it is recommended to avoid playing against them since there exploitation factor is much reduced.



	Tight	Loose
Aggressive	Tricky	Maniac
Passive	Rock	Calling Station

Figure 2 – Player's nicknames by play style

2.1.1 Collecting Opponents' Information

In order to perform accurate assumptions of the opponent's decisions, knowledge about their game playing style is needed. This is achieved by having history of their hands and hand plays. In some occasions the opponents' hands are not shown, when the hand is folded and there isn't a showdown, or the player chooses not to show them[7]. Some methodologies may benefit with that information, such as neural networks. Different strategies even for something that seems as simple as data collecting could be very complex and very useful in the future [1].

Information can also be collected from various programs like Hold'em Manager [13] or PokerTracker [15], these are software tools that record the hands played and show statistics about the players involved. There are other software tools that can offer these features and even more, such like Weka, which can offer more than data mining, because it has a JAVA API and can be easily integrated with other software. Weka also comes with an interface which allows exploring the hand statistics, classifying, clustering, associate or select attributes.

In order for an agent to be successful at playing against human players in an incomplete information game such as poker, it must rely not only in the perfect strategy, but the best strategy against specific types of players. For that it must evaluate the players play style and categorize it to a group, where a specific strategy will be applied against that type of player. The players should be placed in different groups according to their hand play statistics. The best human players are constantly modeling their opponents, changing their strategy according to best fit the game state. The most successful player is the one that is unpredictable in his strategy and also applies the most efficient one against each type of player.

[7] Option only existent on some online poker tables

The core focus for opponent modeling should be based on best response strategies and Nash equilibrium strategies. These two should always be used in order for an agent to develop knowledge about his opponents and obtain best responses to their actions. In case there isn't enough information on a player's behavior a Nash equilibrium strategy should be considered since it is a conserving strategy.

Many opponent modeling methods have been studied such as evolutionary neural networks, where representation, selection, recombination and mutation apply to obtain better agents with each generation. In [2] this was studied and the agents were implemented as evolving neural networks. These agents participated in tournaments consisted of up to 2000 agents for 500 generations and after each generation the best performing one where selected to be the parents of the next generation. But in order to obtain a good agent a large amount of time is needed in order for the evolutionary agents become any good.

Some others approaches studied in opponent modeling use data structures and live data recording. In the paper [3] this approach was used, where the data was stored in a structure called history. This history was divided in two parts, OneRoundData and GlobalRoundData. For the first one the objective it had was to simply map the opponent's hand, action and bets during one game. As for the second one it aggregates the data from OneRoundData for every player in every N games (where N was 500). This data structure was able to store the hand strength of one player per game, game state and per taken action.

One emerging algorithm used in games of incomplete information is the counterfactual regret minimization. This means the difference between the highest utility attainable upon all possible actions and the utility of the action taken, so for each set of information the counterfactual regret is minimized. In a perfect recall game between 2 players, minimizing their regret will lead to a Nash equilibrium strategy since both players tend to search for the best action between all possible ones using the same process [4]. A best response agent should be the best way to exploit any flaw since it gives the best response at any given situation, and they can be computed with not much effort using the CFR algorithm, but there is a problem, it needs to run N times for the N-player game, so if it was to be applied in a 3-plater game it would take in the order of months to compute, and therefore another method should be used, or a solution to minimize the size of search [9].

A program that does not perform opponent modeling will represent no challenge for an adapting poker player. While opponent modeling in perfect information games, where the state of the game is known to all players, is already a challenge, it goes even further when it is about imperfect information games where the players are not always able to observe the actual state of the game.

2.2 Simulation Software

Some simulation software such as Merkat Open TestBed [10] which is a lighter version of the Poker Academy [11]; can simulate games between poker agents in order to test their implementation and their behaviors against different agents. There is a Java API for the Meerkat Open TestBed that allows the customization of agents.

Luís Teofilo's master thesis [5] is about a similar tool called HoldemML, it creates poker agents and it considers their initial bankroll. It contains different poker variants. The focus of his research was to verify the possibility of analyzing human game logs in order to produce competent poker agents. Below there is a figure illustrating the HoldemML framework.

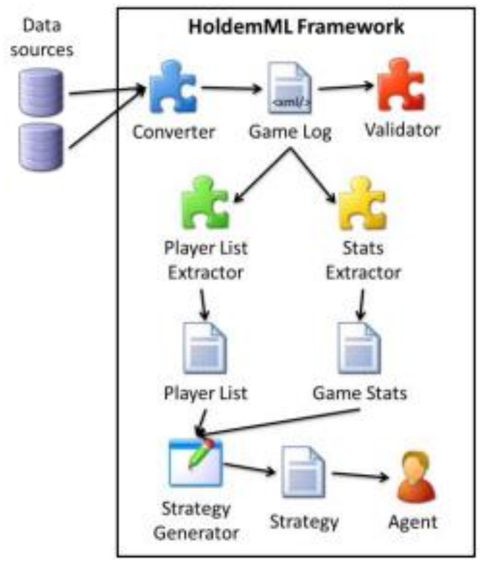


Figure 3 - HoldemML FrameWork

This framework converts game logs into XML. The information is then processed to create 2 documents, “Player List” and “Game Stats”. With these 2 documents a third one is created, the strategy document which is used by the agent to replicate the human strategy. The agent obtained could now be tested using LIACC Poker simulator, the figure below illustrates its architecture.

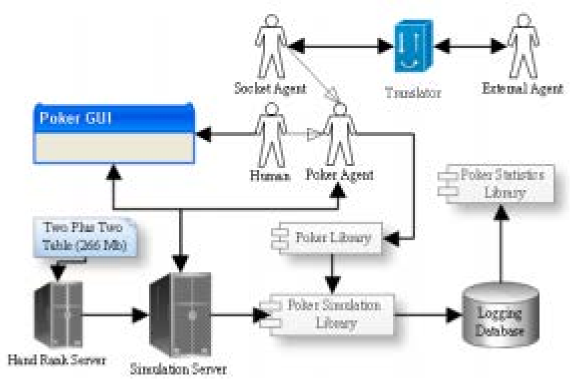


Figure 4 - LIACC Poker Simulator Architecture

The AAAI Competition server has been the most used in the poker games simulations, being able to simulate thousands of games between poker agents; it also is used to determine the winner of the annual poker competition organized by the University of Alberta [12].

2.3 Game State Recognition

2.3.1 Image Processing Software

Software like OpenHold'em can provide some image processing for online poker software table detection, as well as the holding hand, number of opponents in the table and the value of bets and pot. The utility provided from image processing will allow a more accurate game state definition. This is very useful in order to obtain the best action to perform according to each specific situation. An agent will need to know how much money the pot has, how many players are in the hand, what is his current hand and in which position he stands. The user interface of OpenHold'em is shown in figure 5. The commercial version of this tool is called WinHold'em [14].

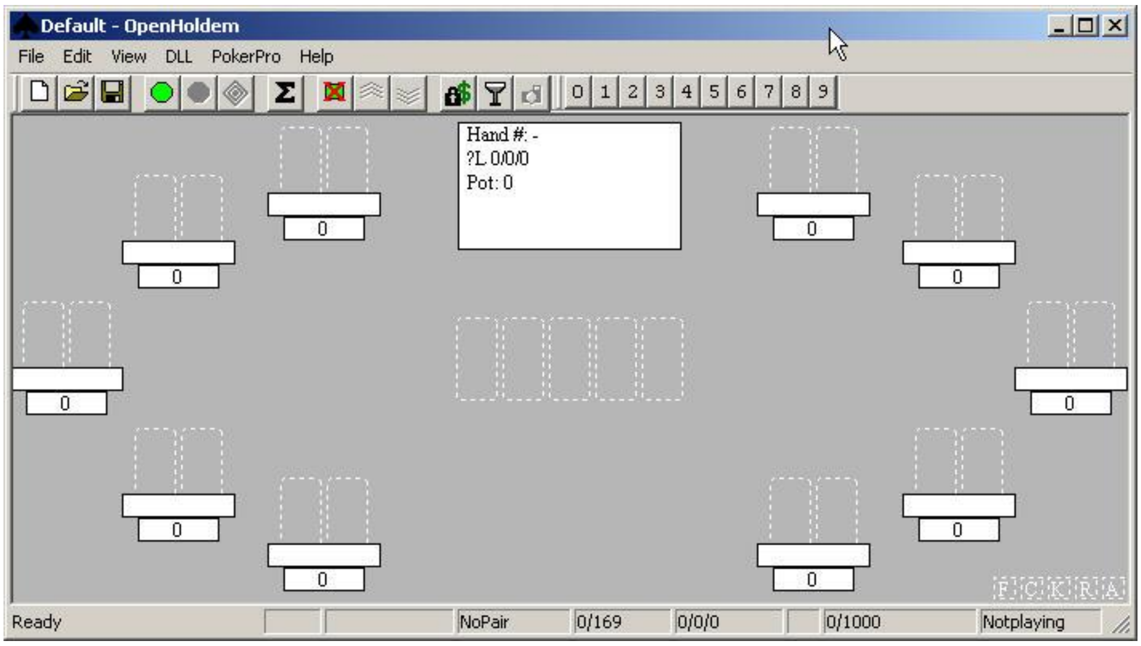


Figure 5 - OpenHold'em

Without using this software or any other, a possible solution could be creating a method of recognizing the online poker software table, seats, chips, position and players using an algorithm that shall run through the computer screen and evaluate the image. The first step should be transforming the image into a gray scale, with a suitable threshold so that patterns can be distinguished [6]. A problem that could be found doing this would be the time to process image recognition. There is a limited time to play each hand, so to use a strategy definition algorithm after using some time for the image processing algorithm, playing each hand could be a hard task to accomplish.

Paulo Martins thesis [18] is an interesting work based on the evaluation of a real poker game through a camera. This means the effective evaluation of cards and chips at a table with an accuracy of 94%.

Haruyoshi Sakai's report, Internet Poker: Data Collection and Analysis [16], discusses data collection methods:

1 – Reading the traffic between the poker client and the server;
2 – Using some software that provides hand histories;
3 – Apply image processing in order to obtain information from the screen.

The report points out the difficulties on each method. The method of his choosing was the third one for being the most promising according to his thoughts. For this to be possible he resorted to the built-in Java library, Robot[8]. This class allows for the creation of simulated user input, both keyboard and mouse. Since simple image matching becomes inefficient as the number of images to compare with grows, the trie based image matching was implemented. This method consists of having a color to act as relevant, and treating the images as strings of 1's and 0's. For example the figure 6 is a letter "T" and the string representation of it would be 1,1,1,10,10,1,1,1. Each number represents the number of foreground color found at each column of this picture. Column one, two and three contain only one white pixel, middle columns contain 10 white pixels, and the last columns contain one white pixel again.

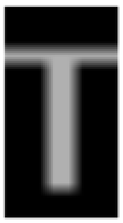


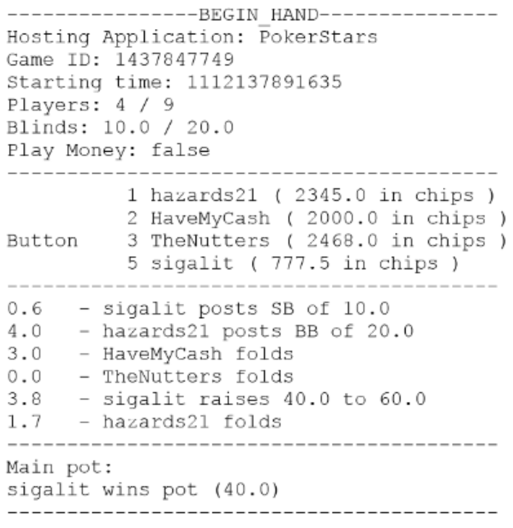
Figure 6 – Letter T

Using this method it was easy to adapt it into text recognition, so text elements like game identification, individual player name and player stack could be read. For player actions it was used simple image matching figure 7.

Figure 7 – Player actions

[8] http://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html

Applying the method for cards and chips, all betting rounds and the outcome of a hand the text file produced containing the information obtained is shown at figure 8.



```
----------------BEGIN_HAND---------------
Hosting Application: PokerStars
Game ID: 1437847749
Starting time: 1112137891635
Players: 4 / 9
Blinds: 10.0 / 20.0
Play Money: false
-----------------------------------------
          1 hazards21 ( 2345.0 in chips )
          2 HaveMyCash ( 2000.0 in chips )
Button    3 TheNutters ( 2468.0 in chips )
          5 sigalit ( 777.5 in chips )
-----------------------------------------
0.6   - sigalit posts SB of 10.0
4.0   - hazards21 posts BB of 20.0
3.0   - HaveMyCash folds
0.0   - TheNutters folds
3.8   - sigalit raises 40.0 to 60.0
1.7   - hazards21 folds
-----------------------------------------
Main pot:
sigalit wins pot (40.0)
-----------------------------------------
```

Figure 8 – Output text sample

2.4 Agents Developed

Some competitions have been held to test the best performing agents. One example is POKI, the strongest agent at the limit Texas Hold'em in the commercial software POKER ACADEMY [7]. At the two player poker game the PsOPTI agents stand out with well-balanced strategies that can defeat average players and compete against strong players until their minor flaws are unveiled. But the best performing one is VEXBOT with its adaptive imperfect information game-tree search, it is able to model opponents and adjust his strategy accordingly, it eventually learns to outplay another computer program and it is a strong opponent for elite human players [7].

2.4.1 Experiments

Darse Billings performed some experiments to test the agents, VEXBOT won all the matches it played, standing out as the strongest agent in the experiments computer versus computer, and having the largest margin of victory over each opponent. Every match consisted of at least 40000 games of poker and the results are shown in the table 1 below [7]:

Table 1 – Vexbot agains other bots (small bets per game)

Program	Vexbot	Sparbot	Hobbybot	Poki	Jagbot	Always Call	AlwaysRaise
Vexbot	0	0,052	0,349	0,601	0,477	1,042	2,983
Sparbot	-0,052	0	0,033	0,093	0,059	0,474	1,354
Hobbybot	-0,349	-0,033	0	0,287	0,099	0,044	0,463
Poki	-0,601	-0,093	-0,287	0	0,149	0,51	2,139
Jagbot	-0,477	-0,059	-0,099	-0,149	0	0,597	1,599
Always Call	-1,042	-0,474	-0,044	-0,51	-0,597	0	0
AlwaysRaise	-2,983	-1,354	-0,463	-2,139	-1,599	0	0

As we can see by the results given in small bets per game, VEXBOT exploits every other agent, with more success against Always Call and Always Raise agents approaching the theoretical maximum exploitation for these two when no other program has been able to come close to it.

2.5 Lokibot

This agent handles each state of the game differently, either pre-flop, flop, turn or river, it uses two components to play: an evaluation of the hand and a betting strategy, where the strategy is influenced by pot odds and a model of opponent present in [8]. For the pre-flop evaluation, the two initial cards, there are 52 possible cards and thus 1326 possible combinations, but only 169 distinct hand types (2-3 is the same as 3-2). For the hand evaluation the thought process behind it is as simply as the 52 cards of the game, minus the state of the game, for example at pre-flop there are 50 unknown cards, we only know our own, at flop 47-2xNplayers holding a hand, turn 46-2xNplayers holding a hand, river 45-2xNplayers holding a hand. So it is possible to discover how strong the agent's holding hand is at flop, turn or river, and if it can improve further more or not. For example if the hand held is 8 and 9 of clubs, and the flop comes 6 of clubs, 7 of clubs and 2 of hearts, the agent's hand is not the best hand at the moment, but it has a huge potential to become the best hand, since the remaining cards left for turn and river include every other club and any ten or any five, the probability of any of those turn out at the turn or at river makes this hand's potential very high. The names given for these potentials where: positive potential (Ppot) and negative potential (Npot). Finally the betting strategy of this agent relies on hand strength and potential which combined gives effective hand strength (EHS).

2.5.1 Experiments

As for experiments made by Darse Billings [8] with Loki variations, the figure 9 below displays the winning rate of five different players. All of these players are a variation of Loki but with something different missing.

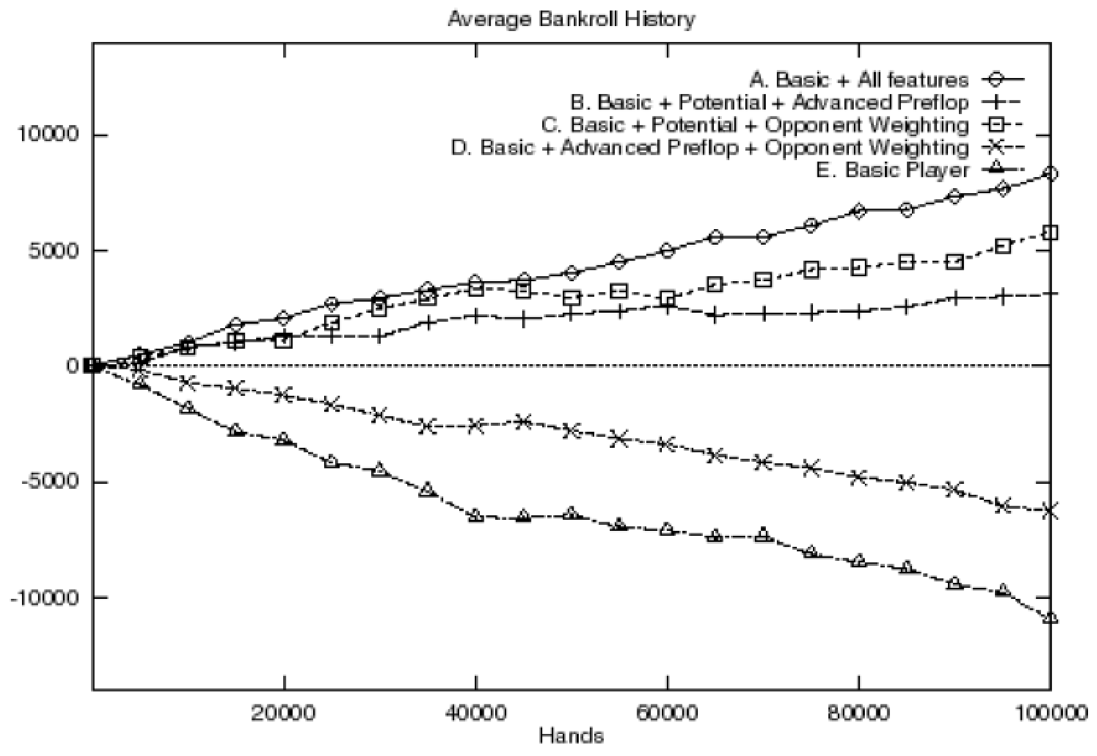


Figure 9 – Average Bankroll History

As we can see the Loki version using all major components stands out over all the other ones that start to miss one or more major components. This agent was also tested against human players on a IRC server, although it was not a game of real money, the agent performed well, playing at the top 10% percent of the players in IRC server.

The study of algorithms for effective poker playing against human players is very appealing since it is required for any system in this field to be able to adapt rapidly, evaluate every step of the game, and adjust while players adjust themselves to it. These are very complex problems, there are other algorithms from other areas of artificial intelligence that may well be suited to this problem, there are so many ways to solve it, a system which needs constant adaption and creativity provides a high demand on research and it will continue to.

2.6 Tools

2.6.1 Holdem Manager Software

Hold'em Manager is one of the most used tools for poker hands analysis along with Poker Tracker [15] which has the same purpose. It gathers hands from the online poker rooms where the user plays at, and it continuously updates statistics for every player playing at each running table. It is a fundamental tool for a player who wants to study his opponents and try to explore any potential mistakes or leaks they might have. The Holdem Manager software provides huge statistical information about the hands played. Various filters can be used to check specific positions or game states. It can be a valuable tool for the study of leaks and future improvements in the agent's decisions. Some important statistics are:

- **VPIP –** this statistic value tells the percentage of times a player makes a call or a raise pre-flop, and it stands for "Voluntarily Put $ In Pot".
- **PFR –** this statistic value tells the percentage of times a player raises a hand pre-flop
- **3-bet –** this statistic value tells the percentage of times a player raises someone's raise pre-flop.
- **4-bet –** this statistic value tells the percentage of times a player raises a 3-bet from another player.
- **C-bet -** this statistic value tells the percentage of times a player bets a flop, after raising pre-flop.
- **Fold to 3bet –** this statistic value tells the percentage of times a player raises and folds to a re-raise. It is possible to know the fold to 3bet for any position at the table. That value will be useful in order to calculate if the expected return is positive or negative against the hand the bot holds.
- **Steal percentage –** this statistical value represents the percentage of times a player raises an unopened pot from the cut-off, button and small blind positions, with the intent of steal the blinds.
- **Agression –** a statistic value that represents the amount of aggressiveness shown by the player in question.

Figure 10 will show an example of the Hold'em Manager software, and a few hands gathered and studied.

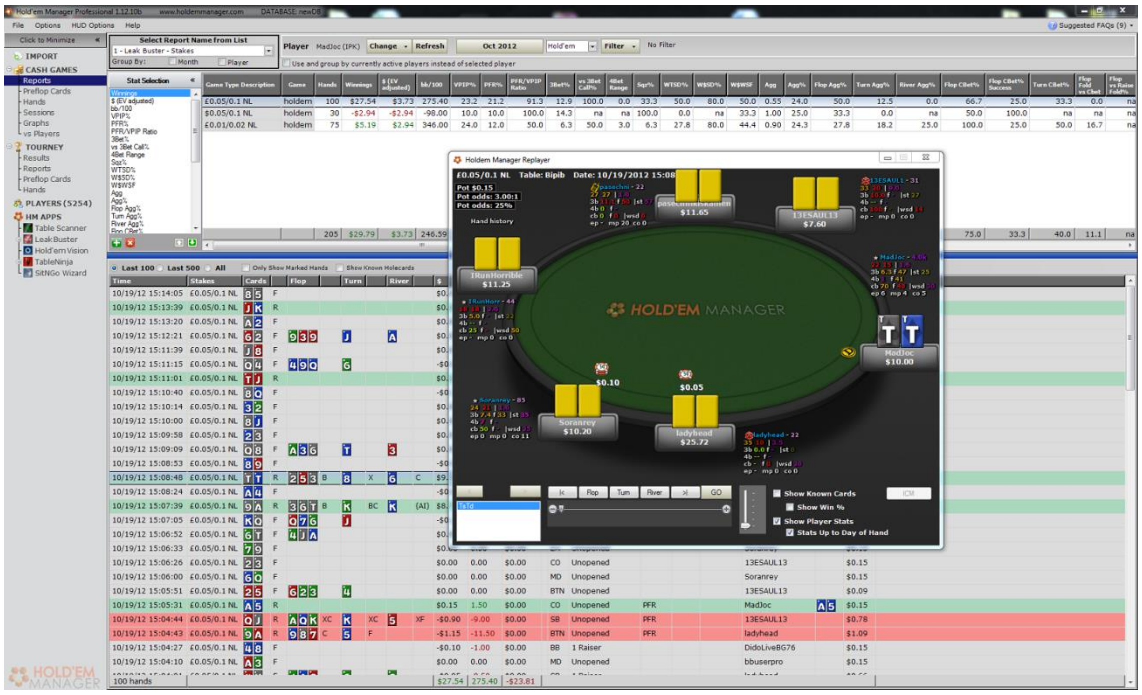


Figure 10 – Holdem Manager software

We can see the number of hands played, as well as how each hand was played, we can open each hand and watch a replay of the hand. The replay shows the statistics of each player at that table and the hand played. The statistics can be seen at figure 11 below:

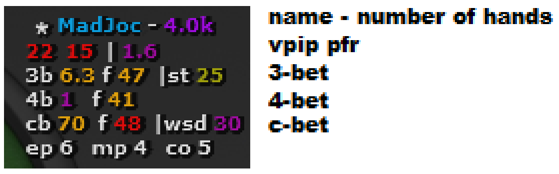


Figure 11 – Player HUD statistics

Analyzing this player, we can see that he has four thousand hands played, and he plays 22% of the hands, and raises 15% of the times. He raises someone's raise 6.3% of the times (3-bet), and folds 47% to someone's 3-bet. Only raises 1% of someone's 3-bet (4-bet) and has a fold percentage of 41% to someone's 4-bet. After raising and getting called pre-flop he continues betting (c-bet) at flop 70% of the times, and folds 48% to someone's c-bet. So from the statistics we could easily conclude that this player usually plays strong hands, and aggressively. This software will be very useful for categorizing players, enabling accurate opponent modeling and will help on defining strategies to outplay them. All the hands are saved in a SQL database format so all the information required can be accessed.

2.7 Conclusions

Haruyoshi Sakai talks about lessons learned, problems found such as confusing game states when parsing betting rounds and recording the correct order in which each player acts. He feels it was a mistake to try and parse betting rounds; a much easier method will be parsing the text console the software offers. He considered this method far more efficient, but by the time he realized it, it would take him more time to rewrite the hand parsing than to finish his image processing implementation.

There is a large amount of research in the Poker game, not only playing agents but also game state recognition, all these studies are important and represent a step forward for games of incomplete information.

In this thesis the text console parsing method was the first to be taken into consideration, since I have had experience in the development of a poker agent. As the game state is recognized in a simple and faster method, there is time for an algorithm to process the best play for that game state. The use of a database alongside with the game state recognition helps in the evaluation of each player at the table, facilitating the definition of game strategies against them.

Chapter 3

Background

This chapter will briefly describe the game of Poker and the game variation chosen for the agent to play. It contains an explanation of poker rules, hand ranks, players that can be classified into different play styles and some advanced information regarding terms, actions and process thoughts behind poker playing.

.

3.1 Poker Game

Poker as an incomplete information game is a constant target of study and research. It is a well-known card game where each player bets an amount of money he thinks his hand is worth. This study will focus only on the No-Limit Texas Hold'em variant, and tables with a maximum of six players.

3.1.1 Table Positioning

Every player as a position that changes every new hand played. The positions are (by order): UTG, MP, CO, BTN, SB and BB. Players at the positions SB and BB must pay the amount accordingly to the table's limits (example: if playing at NL 0.10$, SB = 0.05$ and BB = 0.10$), and the positions shift clockwise every new hand, usually a button chip is placed on the table to mark the button position (figure 12).

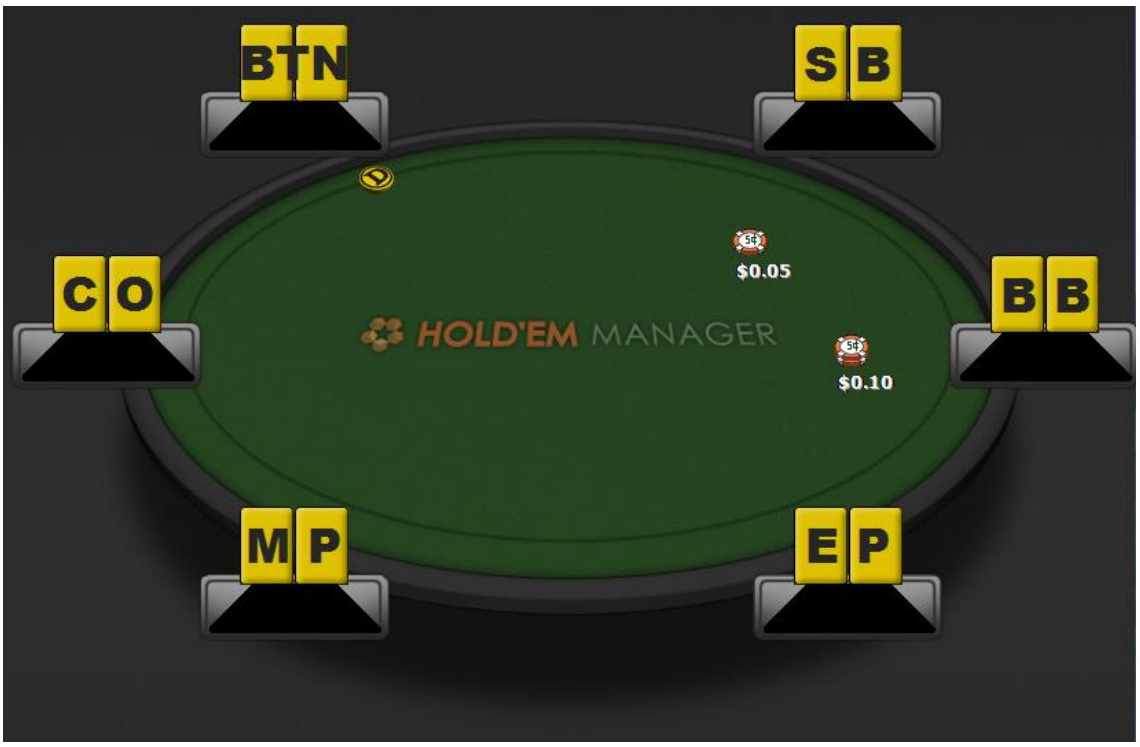


Figure 12 – Table positions

3.1.2 Rules

Playing Texas Hold'em all players face 4 betting rounds, Pre-Flop, Flop, Turn and River. At first, two cards are dealt face down to each player, this round is called pre-flop. Every player must decide, after seeing their hands, if they want to play them or not. The first round of bets occurs and when all bets have been matched this round ends. The flop comes and three cards are dealt face up at the table. At the flop there is the second round of betting, the same rules apply and when all bets have been matched, the turn comes showing one more card, this process repeats itself until the last card is dealt face-up at the table, called the river. At this point there are five cards faced-up at the table (figure 13). After the betting round is over, players must show their cards and the player with the strongest hand wins the pot. Anytime a player bets and the rest decides not to match or raise the bet, the player who bet wins the pot instantly and the other players fold their hands and lose all investment made on that play, in this case players may chose not to show their hands. The two cards dealt to players are called private cards; they are dealt faced down, while cards of the community or shared cards are visible to every player at the table since they are dealt faced up.

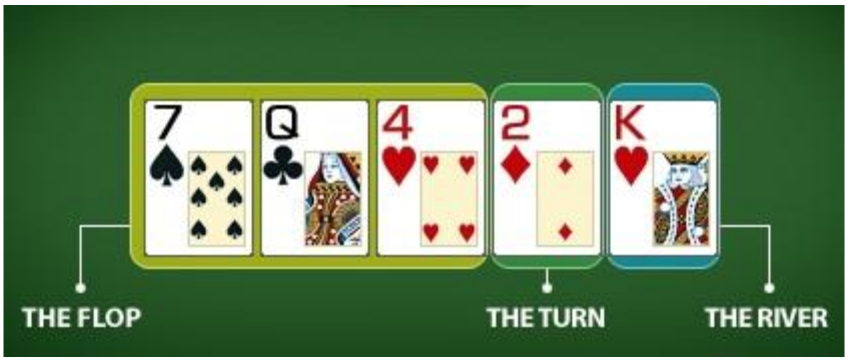


Figure 13 – Betting rounds

3.1.3 Hand Ranks

The winner of the round will be the player holding the strongest hand, the strength of a hand is evaluated by the best combination of five cards. The combinations are shown by the figure 14 (in a descending order):

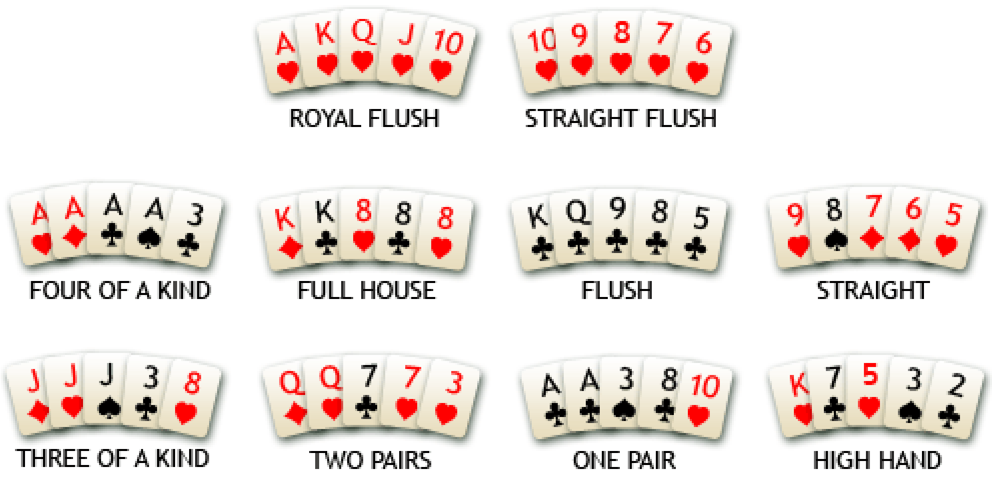


Figure 14 – Hand strengths

Royal Flush:

- Having 5 cards with the same suit and with the highest sequence;

Straight Flush:

- Having 5 cards with the same suit and in a sequence;

Four of a Kind:

- Four cards of the same rank and any other unmatched card;

Full House:

- Having 3 cards of the same rank (three of a kind) and 2 other cards of another rank (pair);

Flush:

- Having 5 cards of the same suit;

Straight:

- Having 5 cards in a sequence;

Three of a kind:

- Having 3 cards of the same rank;

Two pairs:

- Having 2 cards of the same rank combined with another 2 cards of the same rank;

Pair:

- Having 2 cards of the same rank;

High Hand:

- Having none of the above, the highest rank is the high hand.

3.1.4 Advanced Poker Domain Knowledge

In this section there will be a little more information on the theory of poker, more specifically Texas Hold'em and the short stack strategy.

3.1.4.1 Short Stack

Short stack means playing poker usually with a stack (money brought to the table) of 20 big blinds or less. With a small stack the strategy of playing poker is limited mostly to pre-flop and flop decisions. Decisions are made according to positioning at the table, players involved and hand strength.

3.1.4.2 Hand Strength

Hand strength is the probability of a two card combination winning the pot. This probability differs according to each betting round, community cards and number of players involved.

3.1.4.3 Table Position

The position at a table is crucial for making plays, being able to evaluate the position as good or bad, could mean the difference between winning and losing. The most crucial positions are the blinds, since it is to them most of the money is lost, either from steals, or just having a bad hand to try and make a move. The concepts of blind steal and in position to steal must be known so that the leaks from these positions can be lessened. The solution for this leak is using other positions to attempt and recover the money lost to the blinds, stealing the blinds from comfortable positions, for example, from the button.

3.1.4.4 Hand Ranges

Another important topic is hand ranges, which mean the most probable hands that a player could be playing. A top 2% hand range means playing one of these 4 hands: pair of aces, pair of kings, pair of queens or ace king suited.

3.1.4.5 Equity

For each hand held at any position there is a percentage associated to the win or tie chance of it, called equity, for example aces pre-flop against any other random hand have an equity of around 80%.

3.1.4.6 Expected Value

Expected value is a term used frequently by poker players, and means the average of a play being good or bad after large amount of repetitions. A positive expected value will mean that after a large number of times that play as been made, it will eventually turn into profit, a negative expected value is just the opposite.

3.1.4.7 Fold Equity

Fold equity will be an additional equity picked up from the chance of an opponent folding his hand, for example if there is equity of 49% and the probability of an opponent folding is 50%, the fold equity would be the opponents equity multiplied by is folding percentage, assuming he folds 50% this means 0.5*51 = 0.255 and this value is added to 49% meaning the equity of the play is actually 74.5%.

3.1.4.8 Win Rate

Win rate is often measured in big blind per 100 hands, means the amount of money won over a set period of time. The standard abbreviation is X bb/100, X being the number of big blinds. A good player usually has a winning rate between 1 and 4 bb/100.

3.1.4.9 Variance

Variance is the fluctuations in probability, it is the difference between how much profit a certain play would make, and how much it actually did. Eventually in the long run it evens out, but in the short term it could implicate some harsh losses.

3.1.4.10 Outs

Outs mean any card that will improve the current hand strength after the flop is dealt.

3.1.4.11 Bad Beat

It is called a bad beat when a player with the strongest hand ends up losing to another player with a low equity.

3.2 Summary and Conclusions

This chapter was intended to clarify some aspects of the game such as rules, hand strengths, players and deeper knowledge of the game mechanics. At first it seems a fairly easy game with obviously its own set of rules, but looking further into the game, the amount of information to be taken into account before each move increases, and with it the complexity of the game. The accurate assessment of the best play to make will become harder as the information from the game state increases. The game of Poker is bound by mathematic theory, and probability calculations.

Chapter 4

Agent Implementation

This chapter focuses on the development of the project, its architecture and how the agent's behavior was created. The most important features will be explained and all necessary communication between the Holdem Manager database and the java program will be mentioned. At the end there's a brief conclusion containing the problems encountered while the programing phase, and future improvements to be done.

4.1 Programming Language

In this project there are two programming languages, structured query language known as SQL and Java programing language. The SQL language was needed in order to communicate with the Holdem Manager database, so this language was embedded into java code in order to access and query the database anytime needed. Moreover the Java programing language allows the creation of robust programs just by taking advantage of its software technologies like: object-orientation, multi-threading, structured error-handling and garbage collection.

4.2 Application Struture

In this section the program structure is described, and the most important classes and methods are explained. For the architecture of the program, nine different classes where created:

- ***Card.java*** – Class that represents a card, by a suit and a rank;
- ***CardPair.java*** – Class that represents a hand formed by two cards and their value;
- ***Player.java*** – Class that represents a player, and all his hand statistics;
- ***Poker.java*** – Main class that uses all resources available to run the program;
- ***PokerI.java*** – Class that represents the user interface;
- ***Rank.java*** – Class that represents the enumeration of card ranks;
- ***SQL.java*** – Class the establishes the communication with the SQL database
- ***Suit.java*** – Class that represents the enumeration of card suits
- ***TwoPlusTwoHandEvaluator.java*** – Class that represents the calculations for hand strengths, hand equity and all combinatorial calculations.

There are also two auxiliary files:

- ***2cards.csv*** – Containing all possible hand combinations with 2 cards and their associated strength as an integer value;
- ***TwoPlusTwoTable.dat***[9] – Containing all possible made hands, from high card to royal flush, and their respective value.

The three most important classes are: ***TwoPlusTwoHandEvaluator.java***, ***Poker.java*** and ***SQL.java***. They are specified below, where the two first classes will be at the agent development section, while the ***SQL.java*** class will be described at the database communication section.

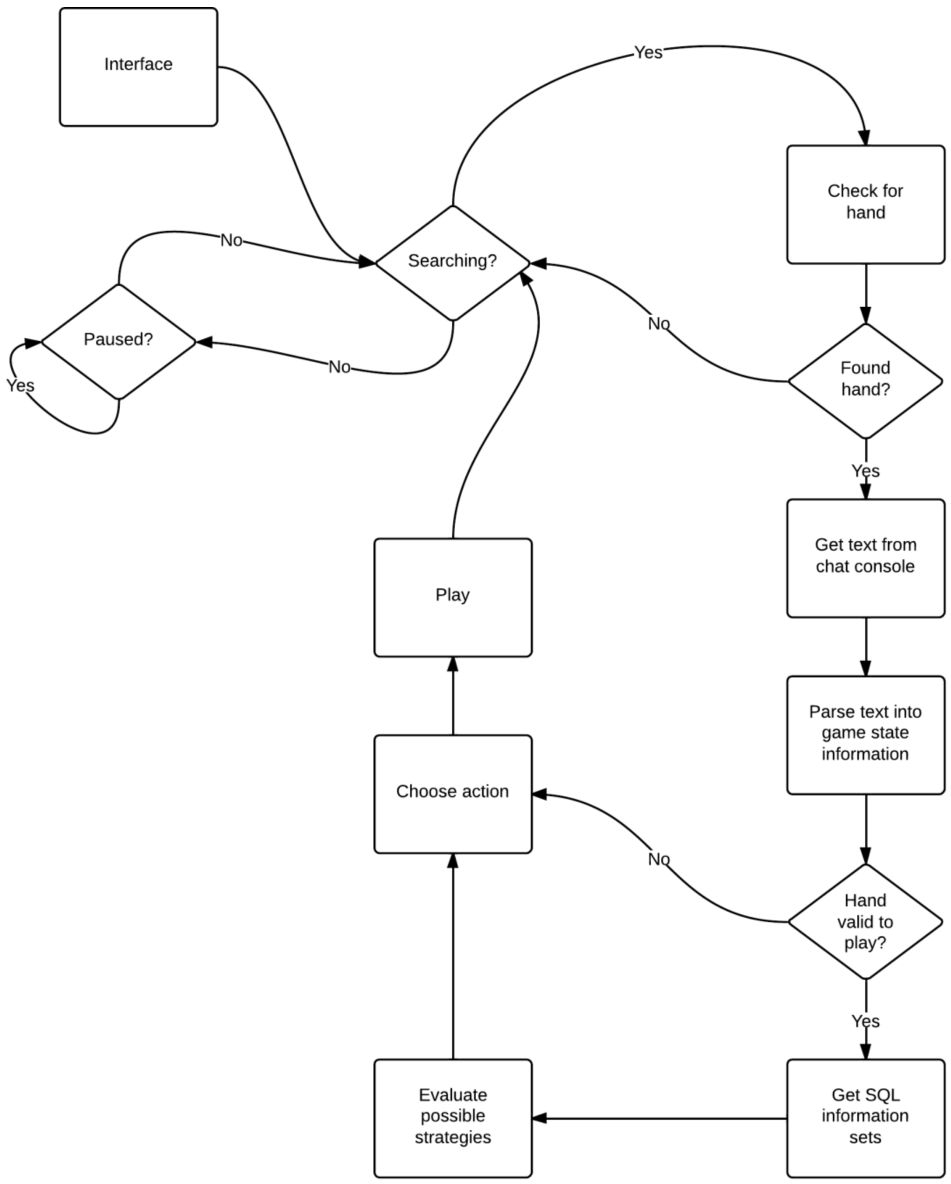


Figure 15 – Flow chart

[9] forumserver.twoplustwo.com

4.3 Agent Development

Here the development of the agent is described as well as important functions created that allowed the strategy definition for each situation the agent faced.

The ***TwoPlusTwoHandEvaluator.java*** class contains some functions which aid the calculations that the agent must perform in order to evaluate the best action to take:

evaluate(Card… hand) – This function will receive a minimum of 5 cards and a maximum of 7 cards as a parameter, and will return the integer value of the best 5 card hand combination. This function uses the pre-processed tables loaded previously, so that the time used to evaluate is minimum. This function is responsible for evaluating the hand strength.

getBoards (Card myCard1, Card myCard2, Card oppCard1, Card oppCard2) – this function will generate all possible cards to create random boards, but not using either the hero's cards or the opponent's cards. This function is called from inside the ***Equity()*** function iterating through all the opponents possible cards (the opponents range perceived from the information gathered by the Holdem Manager database).

Equity (Card c1, Card c2, int percent) – this function will return the win percentage that the hand made by ***c1*** and ***c2*** has against the range given as parameter named ***percent***. This is accomplished by using the ***percent*** parameter to retrieve a sub list of the opponents range. This range is used to get all the possible board combinations, using the ***getBoards()*** function. Then the function ***evaluate()*** is called with the hero's hand and each possible opponent's hand, along with 5 random cards, extracted from the sub list of possible boards, to determine the winner on that specific board. These results are saved in global variables called ***win***, ***lose*** and ***tie***. The function repeats this process one hundred thousand times and returns the win percentage calculated by:

$$\frac{\mathbf{win}}{(\mathbf{win} + \mathbf{tie} + \mathbf{lose})}$$

In summary, this function returns a probable outcome percentage after 100 000 iterations of a hand on boards generated randomly against all possible opponent's hands.

4.3.1 Strategy Definition

The ***Poker.java*** class contains the most important functions for game state evaluation and recognition, user simulated input and strategy definition according to the information gathered. Brief descriptions of some of those functions are presented below:

checkCards() – This function will be determining the strategy to adopt according to all information gathered until this point. All options all divided either by the position held at the table, the strength of the hero's hand, the game state, and the opponents specific pre-flop values. Assumptions like betting all money on a strong hand, or attempting to steal a player who has been abusing his position, or calculating hand ranges and determining the best action according to the expected value, are all possible to be taken. This function can be improved according to game theory, and further opponent evaluation, for now the best calculations are made from the blinds, where it is crucial to minimize the losses, or avoid being stolen by other players. For example, if the agent finds himself at the small blind and has the information that the player from the button has been stealing too much, the agent defends his small blind by 3betting the opponents raise pre-flop, this is done taking into account the expected value from the play.

prepareChat(String[] chat) – This function will be parsing the text gathered from the game console and extract the valuable information of the game state and the agents hand. It makes use of the java Toolkit[10] class in order to obtain the computer's clipboard content as a string representation.

getExpectedReturn(double F3b, double Nplayers, double pot, double EQ, double bbs) – This function will calculate the expected value according to various information gathered from the players, the game state and the function ***Equity()*** from the ***SQL.java*** class. The parameters mean fold to 3 bet, number of players at the hand, value of the pot, equity of the hand, and the value of the blinds. The formula is created by calculating the amount of money won when the player folds or loses the hand to the agent subtracted by the number of times the player doesn't fold and wins the hand against the agent:

$$\Big((\boldsymbol{f3b} \times \boldsymbol{pot}) + \big((\mathbf{1} - \boldsymbol{f3b}) \times \big(\boldsymbol{eq} \times (\boldsymbol{pot} + (\boldsymbol{stack} \times \boldsymbol{players}))\big)\big)\Big) - ((\mathbf{1} - \boldsymbol{fold}\%) \times ((\mathbf{1} - \boldsymbol{eq}) \times \boldsymbol{pot})$$

The formula is just the aggregation of all possible scenarios, which are:

- Winning the pot when the opponent or opponents fold the hand;

$$(\boldsymbol{fold}\% \times \boldsymbol{pot})$$

- Winning the pot when the opponent or opponents call;

[10] http://docs.oracle.com/javase/7/docs/api/java/awt/Toolkit.html#getSystemClipboard%28%29

$$((\mathbf{1} - \boldsymbol{fold}\%) \times (\boldsymbol{equity} \times (\boldsymbol{bbs} + (\boldsymbol{stack} \times \boldsymbol{players}))))$$

- Losing the pot when the opponent or opponents call.

$$((\mathbf{1} - \boldsymbol{fold}\%) \times ((\mathbf{1} - \boldsymbol{equity}) \times \boldsymbol{stack})$$

For every scenario the expected value is calculated, and in the end the total amount expected is just the sum of winning scenarios subtracted by the losing scenario.

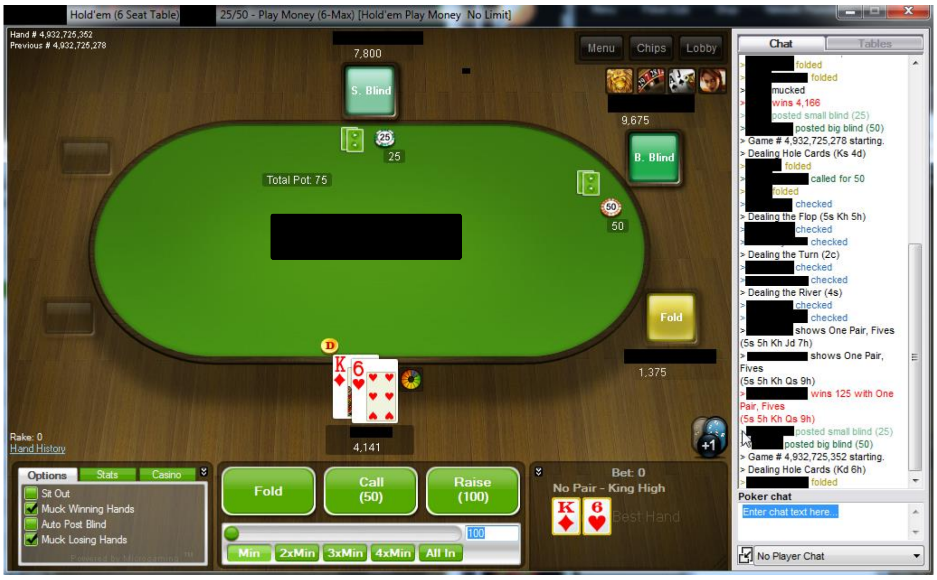
4.3.2 Game State Recognition

The development of the agent started by parsing the text contained in the online poker room console. The console could be easily accessed and configured to only show player's actions and game states. So the only problem was how to obtain the text information from the console. The problem was solved using the Java Robot[11] class found at its API. This class allows for the simulation of user's input this being either mouse or keyboard. By making the mouse move to a determined position and having the shift pressed, it was possible to copy any portion of text needed from the chat. This is an example of a copied portion of the chat console:

> AlexandreC20 posted small blind (€0.01)
> Nemo707 posted big blind (€0.02)
> Game # 4,912,858,846 starting.
> Dealing Hole Cards (10d 3h)
> xavi10x raised for €0.04
> Siim_H folded

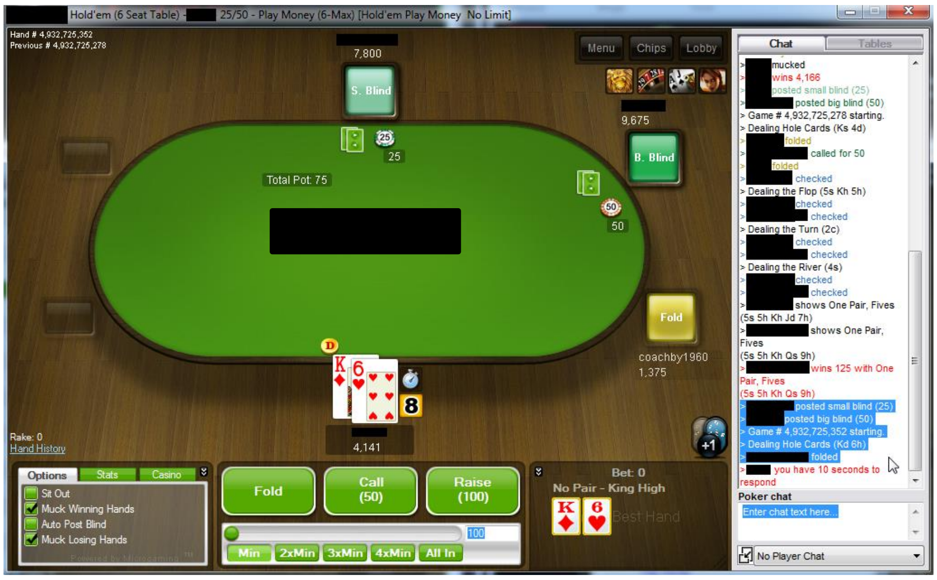
Every line at the chat console as a unique color associated to it, so it is easy to identify what each line refers to only by knowing the color. Once again using the Robot class it was possible to find the color[12] for the small blind posting and thus always having the start of a new hand copied to the computers clipboard, see figures 17 and 18.

[11] http://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html
[12] http://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html#getPixelColor(int,%20int)



```
> [redacted] folded
> [redacted] folded
> [redacted] mucked
> [redacted] wins 4,166
> [redacted] posted small blind (25)
> [redacted] posted big blind (50)
> Game # 4,932,725,278 starting.
> Dealing Hole Cards (Ks 4d)
> [redacted] folded
> [redacted] called for 50
> [redacted] folded
> [redacted] checked
> Dealing the Flop (5s Kh 5h)
> [redacted] checked
> [redacted] checked
> Dealing the Turn (2c)
> [redacted] checked
> [redacted] checked
> Dealing the River (4s)
> [redacted] checked
> [redacted] checked
> [redacted] shows One Pair, Fives
(5s 5h Kh Jd 7h)
> [redacted] shows One Pair,
Fives
(5s 5h Kh Qs 9h)
> [redacted] wins 125 with One
Pair, Fives
(5s 5h Kh Qs 9h)
> [redacted] posted small blind (25)
> [redacted] posted big blind (50)
> Game # 4,932,725,352 starting.
> Dealing Hole Cards (Kd 6h)
> [redacted] folded
```

Figure 17 – Small blind color detection



```
> [redacted] mucked
> [redacted] wins 4,166
> [redacted] posted small blind (25)
> [redacted] posted big blind (50)
> Game # 4,932,725,278 starting.
> Dealing Hole Cards (Ks 4d)
> [redacted] folded
> [redacted] called for 50
> [redacted] folded
> [redacted] checked
> Dealing the Flop (5s Kh 5h)
> [redacted] checked
> [redacted] checked
> Dealing the Turn (2c)
> [redacted] checked
> [redacted] checked
> Dealing the River (4s)
> [redacted] checked
> [redacted] checked
> [redacted] shows One Pair, Fives
(5s 5h Kh Jd 7h)
> [redacted] shows One Pair,
Fives
(5s 5h Kh Qs 9h)
> [redacted] wins 125 with One
Pair, Fives
(5s 5h Kh Qs 9h)
> [redacted] posted small blind (25)
> [redacted] posted big blind (50)
> Game # 4,932,725,352 starting.
> Dealing Hole Cards (Kd 6h)
> [redacted] folded
> [redacted] you have 10 seconds to
respond
```

Figure 18 – Copying game console chat into clipboard

After having the content copied to the clipboard it is easy to access it by using the java class Toolkit[13]. Using the function ***prepareChat()*** the text is parsed and all information about the game state, players involved and our hand is easily achieved. All the essential information is then stored in the appropriate structures, to be used when necessary. Parsing the string obtained was just a simple use of the *split()* function for strings, splitting the unchangeable words on every occurrence, for example "*posted small blind*". The positions were easily discoverable simply because the order for each player to act would always be from the earliest positions until

[13] http://docs.oracle.com/javase/7/docs/api/java/awt/Toolkit.html#getSystemClipboard%28%29

the blinds, so each position was filled until the big blind was reached, or it was our turn to act. From parsing the chat console the program is able to obtain the following sets of information for each hand:

- Positions of players;
- Pot size;
- Players willing to play;
- Action each player took;
- Action is taking place at pre-flop, flop, turn, or river
- Our cards, and flop, turn or river cards when applicable;

This information is used later on for the calculation of the expected value or the best action according to the hand read.

4.4 Database Communication

This section will explain the communication between the program and the Holdem Manager database (figure 19). The first step was establishing a successful connection with the database, a SQL library, named PostgressSQL JDBC, was used to facilitate this procedure. With a successful connection established there was the need to understand the Holdem Manager Database table and column names so that specific statistics would be found. Eventually some information was found on the Holdem Manager forums [17] which contained some examples that where very helpful for the first queries:

- Getting some statistics from players[14];
- Getting current active players[15];
- Getting some columns description[16].

The most helpful information found was a text[17] document containing various statistical definitions for Holdem Manager Database. The text document contained column names and evaluation formulas that helped in the construction of more elaborated queries.

[14] http://forums.holdemmanager.com/manager-general/32307-using-stat-formulas-sql-query.html
[15] http://forums.holdemmanager.com/manager-general/31032-query-get-current-active-players.html
[16] http://forums.holdemmanager.com/manager-general/669-description-required-some-database-columns.html
[17] See the contents of the full query at the end of this thesis, in the appendix section

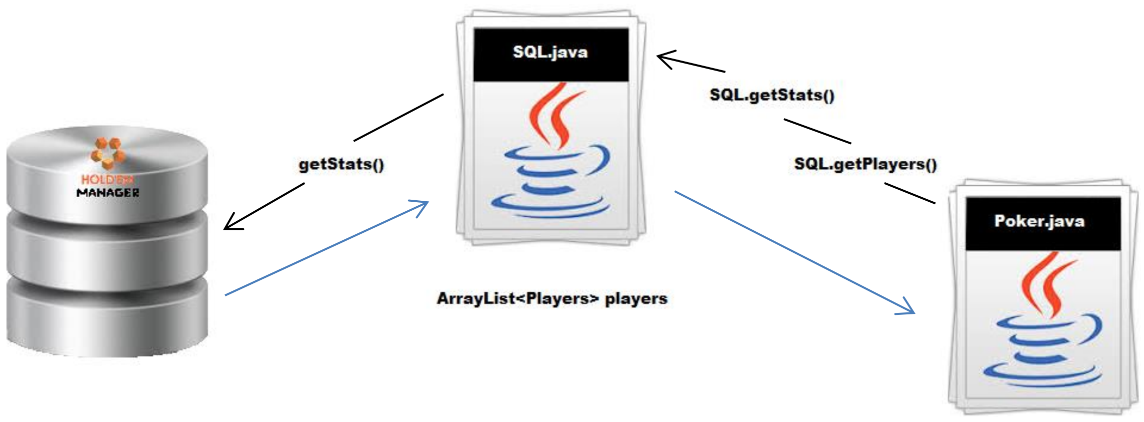


Figure 19 - Agent's Architecture

4.4.1 SQL Queries

The first query made was obtaining the name of my hero at the poker room, a simple query where only the name was searched among all players. The real interesting queries and most difficult to obtain took a little while to construct and validate. They aim to obtain the name of the player currently active at a table, his total hands, vpip, pfr, his raise per every position on an unopened pot, and his fold per position facing a 3 bet. A portion of a query will be presented below which refers to the folding percentage facing 3 bet at the button position:

- "…`sum (case when positiontype_id = 5 and phmisc.threebetresponsetype_id = 1 then 1 else 0 end) as` **`FoldToThreeBet_Button`**`, sum(case when positiontype_id = 5 and phmisc.threebetresponsetype_id > -1 then 1 else 0 end) as` **`FacingThreeBet_Button`**…"

  Fold to 3 bet from the button will be the evaluation of:

$$\frac{\boldsymbol{FoldToThreeBet\_Button} \times \mathbf{100}}{\boldsymbol{FacingThreeBet\_Button}}$$

After the construction and validation of all queries[18], all the information was stored as new players and added to an array of players. Every action, on every hand a player has played is now stored and ready to be accessed in order to classify them and evaluate the best action to take against them.

[18] The complete query is demonstrated at the end, in the appendix section

4.5 Conclusions

The most challenging aspect in the implementation was the reaction time of the agent, since there is a fixed time to play each hand. The combinatorial calculations where consuming the most time in the agent's reactions, luckily with all the information available some situations did not need combinatorial calculations. For example having a hand like deuce three will result in a fold from the agent, without the need of any calculations since it is one of the worse hands on poker. So by eliminating hands not eligible to play, it boosted the agent's speed and its ability to play more than one table at a time. To determine which hands are available and which aren't, some sub lists where created from the hand strengths table where its value would be above a significant range percentage.

Chapter 5

Experiments and Results

Here it will be described the agent's results and experiments performed in order to evaluate its accomplishments. The primary objective was to create an agent able to play online poker against human players. A secondary objective was to ensure the agent's performance to be at least break even, since there is an amount of money that goes to the poker room called rake. This amount is usually 3% of the total pot, and a player who neither wins nor loses at online poker games is actually winning by a little, since 3% of every pot played is lost to rake.

Both of these objectives have been fulfilled with success. The second objective deserves a special attention since the agent was able to make a profit against human players, being able to exploit some of the human players for the total number of hands played.

5.1 Experiments

The agent was put to play against human players at play money tables, for the purpose of validating the game state recognition relying only on the table chat console, without database communication. This turned out to prove that the functions for table recognition where working and the player's actions, positions and the agent's hand were being read successfully. Since the Holdem Manager Database doesn't recognize play money tables, the only way to verify if the ***SQL.java*** was working properly was to play at a real money table and verify if the player's statistical information was being read and processed by the agent.

Starting at No-Limit Texas Hold'em and blinds[19] of 0.01/0.02€, the agent is configured to play at tables of 6 maximum players. It is necessary to enter tables with a minimum of 4 players plus the agent and configure the poker room to seat at each table with 10 big blinds. After the agent's stack becomes larger than 20 big blinds, there is the need to switch to a new table. Multiple tables can be played simultaneously as long as the time for each play doesn't run out.

The experiment of playing poker with real money went as expected and the agent performed its tasks correctly. The results will be stated in the next section along with the discussion of some plays and interesting evaluations made by the agent.

[19] Amount of money forced to give at a specific position, the small blind position posts half the amount the big blind does

5.2 Results

With the available testing time this agent was able to play 3814 hands against human players, making a profit of 1.48bb/100. This means the agent wins 1.48 big blinds for each 100 hands played, since the big blind value is 0.02€ the amount won is 1.13€.

5.2.1 All time results

A graphical representation of the hands played and the agent's progress is shown on the figure 20.

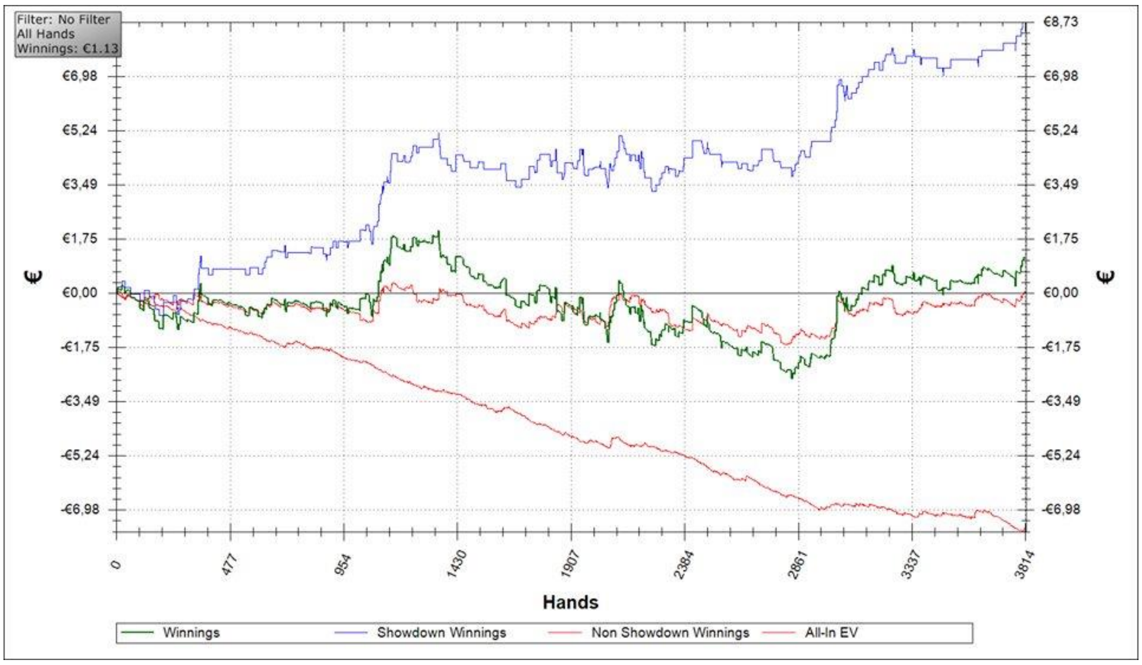


Figure 20 – All time results

The red line close to the green one means the expected value from each play the agent made. The other lines represent the showdown winnings, blue line, and the non-showdown winnings, red bottom line. The red line won't decrease as much when other players fold to us, and decreases the most when folding to others. The conclusion taken from this graph is the importance of stealing or defending blinds, otherwise folding 0.02 and 0.01 cents too many times will result in losing too much money. There is a slight difference on the non-showdown line, bottom red line, after the 2800 hands, the reason is an improvement on the agent's pre-flop steal evaluation of its opponents. Some hands will be detailed in order to observe the agents behavior against different opponents. A few examples can clearly show the agent's awareness and ability to outplay some human players found at the tables.

Example 1, a bad player calling with a worse hand:

- ***Hero[20] posts small blind [$0.01 USD].***
  Player1 posts big blind [$0.02 USD].
  Dealt to Hero [5d Ad]
  Player2 folds
  Player3 ***calls [$0.02 USD]***
  Player4 folds
  Player1 folds
  Hero raises [$0.19 USD]
  Player5 folds
  Player3 calls [$0.18 USD]
  Hero shows [5d, Ad]
  Player3shows [Ts, Js]
  *** Dealing Flop ** [8d, Ah, 8c]*
  *** Dealing Turn ** [9h]*
  *** Dealing River ** [2d]*
  Hero shows [5d, Ad]
  Hero wins $0.38 USD from main pot

This hand the agent planned in stealing the pot from the big blind and from the middle position limp. The player Player3 has vpip of 60% and pfr of 10%, so this player will make a lot of mistakes, and this was one of them, calling with a worse hand, his equity against the agent equity is 46%.

Example 2, defending blind from button steal attempt:

- ***Hero posts small blind [$0.01 USD].***
  Player1 posts big blind [$0.02 USD].
  Dealt to Hero [8s 6h]
  Player2 folds
  Player3 folds
  Player4 folds
  Player5 raises [$0.06 USD]
  Hero raises [$0.24 USD]
  Player1 folds
  Player5 folds
  Hero wins $0.33 USD from main pot

At this situation the agent was able to know the % of times the player Player5 was raising from that position on an unopened pot, since it was high enough to be considered abusive, the agent counter acted by defending his blind being successful at it.

[20] The nicknames will be omitted and replaced by generic name attribution, the Hero refers to the agent

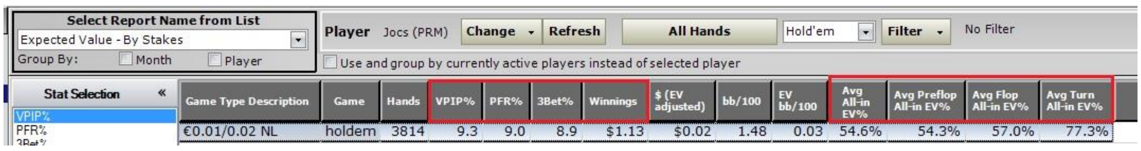
Example 3, defending big blind against small blind steal:

- ***Player1 posts small blind [$0.01 USD].***
  Hero posts big blind [$0.02 USD].
  Dealt to Hero [9h Kd]
  Player2 folds
  Player3 folds
  Player4 folds
  Player5 folds
  Player1 raises [$0.05 USD]
  Hero raises [$0.22 USD]
  Player1 calls [$0.18 USD]
  Player1 shows [Ks, 5s]
  Hero shows [9h, Kd]
  *** Dealing Flop ** [5c, Qs, 6h]*
  *** Dealing Turn ** [9s]*
  *** Dealing River ** [Ac]*
  Hero shows [9h, Kd]
  Hero wins $0.44 USD from main pot

This hand shows the ability of the agent to evaluate hand ranges and decide to push allin against a tight human player. This decision was based on the probability of the opponent folding his hand since it would be worse more often than not. The opponent made the mistake of calling, finding himself in a dominated situation, where his hand is beaten by the agent's hand. The agent's performance in this hand is proudly verified.

5.2.2 Agent's Statistical Information

Below we can see some filters applied on the Holdem Manager software, which show the agent's statistical information:

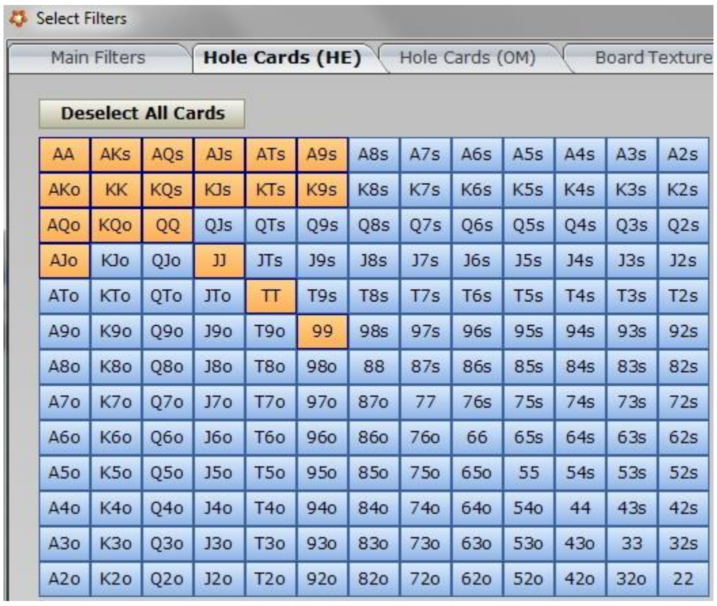

Select Report Name from List
Expected Value - By Stakes
Group By:
Month
Player
Player Jocs (PRM)
Change
Refresh
All Hands
Hold'em
Filter
No Filter
Use and group by currently active players instead of selected player
Stat Selection
VPIP%
PFR%
3Bet%

Game Type Description	Game	Hands	VPIP%	PFR%	3Bet%	Winnings	$ (EV adjusted)	bb/100	EV bb/100	Avg All-in EV%	Avg Preflop All-in EV%	Avg Flop All-in EV%	Avg Turn All-in EV%
€0.01/0.02 NL	holdem	3814	9.3	9.0	8.9	$1.13	$0.02	1.48	0.03	54.6%	54.3%	57.0%	77.3%

Figure 21 – General statistics and expected values

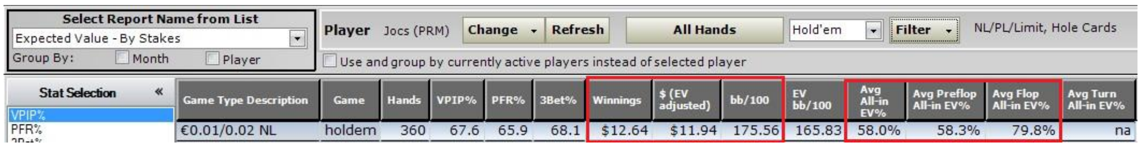
Figure 21 shows the general stats of the agent, a solid player with a vpip of 9.3, pfr of 9.0, and a 3bet of 8.9, a very selective poker player. The most impressive stats should be the average all-in percentages, showing an overall of 54.6% average all-in equity which is excellent.

Filtering by hole cards and choosing the top hand ranges like figure 22 shows, the amount the agent won playing only these hands is displayed at figure 23.



AA	AKs	AQs	AJs	ATs	A9s	A8s	A7s	A6s	A5s	A4s	A3s	A2s
AKo	KK	KQs	KJs	KTs	K9s	K8s	K7s	K6s	K5s	K4s	K3s	K2s
AQo	KQo	QQ	QJs	QTs	Q9s	Q8s	Q7s	Q6s	Q5s	Q4s	Q3s	Q2s
AJo	KJo	QJo	JJ	JTs	J9s	J8s	J7s	J6s	J5s	J4s	J3s	J2s
ATo	KTo	QTo	JTo	TT	T9s	T8s	T7s	T6s	T5s	T4s	T3s	T2s
A9o	K9o	Q9o	J9o	T9o	99	98s	97s	96s	95s	94s	93s	92s
A8o	K8o	Q8o	J8o	T8o	98o	88	87s	86s	85s	84s	83s	82s
A7o	K7o	Q7o	J7o	T7o	97o	87o	77	76s	75s	74s	73s	72s
A6o	K6o	Q6o	J6o	T6o	96o	86o	76o	66	65s	64s	63s	62s
A5o	K5o	Q5o	J5o	T5o	95o	85o	75o	65o	55	54s	53s	52s
A4o	K4o	Q4o	J4o	T4o	94o	84o	74o	64o	54o	44	43s	42s
A3o	K3o	Q3o	J3o	T3o	93o	83o	73o	63o	53o	43o	33	32s
A2o	K2o	Q2o	J2o	T2o	92o	82o	72o	62o	52o	42o	32o	22

Figure 22 – Pre-flop hand filter


Select Report Name from List
Expected Value - By Stakes
Group By: Month Player
Player Jocs (PRM)
Change
Refresh
All Hands
Hold'em
Filter
NL/PL/Limit, Hole Cards
Use and group by currently active players instead of selected player
Stat Selection
VPIP%
PFR%

Game Type Description	Game	Hands	VPIP%	PFR%	3Bet%	Winnings	$ (EV adjusted)	bb/100	EV bb/100	Avg All-in EV%	Avg Preflop All-in EV%	Avg Flop All-in EV%	Avg Turn All-in EV%
€0.01/0.02 NL	holdem	360	67.6	65.9	68.1	$12.64	$11.94	175.56	165.83	58.0%	58.3%	79.8%	na

Figure 23 – Statistical information with filtering from figure 17

The agent made a profit of 12.65€ just by playing those hands, the amount from losses belong to folding big blinds and small blinds. The number of hands played at these positions is specifically 701 at the big blind and 695 at the small blind. If all of these hands where to be folded the amount lost just to blinds would be:

$$(\mathbf{701 \times 0.02}) + (\mathbf{695 \times 0.01}) = \mathbf{20.97€}$$

20.97€ of loss is a considerable amount for just nearly 1400 hands, if all of them where to be folded. The blinds are a very important position and probably the best place to improve the agent's behavior in order to show even more profit. Since playing good hands is easy, the hard task is to evaluate either a hand is good for a certain play or not, even if it is a break even play, it's better than folding a blind.

5.2.3 Results from Each Position at Table

Applying some further filters, for example by positions, the following results are obtained (table 2):

Table 2 – Results by position

Position	Hands	Winnings	EV (adjusted)	VPIP%	PFR%	3Bet%	Avg All-in EV%
Small blind	695	-1.43€	-2.80€	14.0%	13.5%	11.5%	51.8%
Big blind	701	-4.47€	-4.73€	10.4%	10.1%	9.5%	52.4%
Early	411	1.54€	1.86€	5.6%	5.6%	not valid	64.1%
Middle	620	0.77€	1.34€	6.8%	6.8%	6.5%	57.5%
Cutoff	685	-0.34€	1.18€	7.2%	6.9%	5.5%	56.0%
button	702	5.06€	3.17€	10.0%	9.5%	6.9%	55.6%
Totals	3814	1.13€	0.02€	9.3%	9.0%	8.9%	54.6%

The conclusions taken from these results are mostly that the blinds will always have the threat of losing money; the only way to lessen this leak is a better evaluation of stealing [21]possibilities. Also the agent's performance by position is very satisfying, showing a profit on almost all positions excluding the blinds and the cutoff. The cutoff negative income is probably due to variance, since the expected value from that position is positive. A very satisfying statistic to see is the average all-in percentage, it is above 50% in all positions, this is surely a proof that the more hands the agent plays the more profit it will attain. Taking in mind the small blind VPIP being the highest among all, means the agent tries to steal the big blind every chance he sees fit. Also the highest average all-in percent comes from the early position, which would be expected since it is the position the agent plays more seldom, making his hand ranges a lot stronger. We can see a direct connection between playing more hands, higher VPIP; and having less chances of winning, average all-in percent, but also higher chances of winning by stealing blinds.

5.2.4 Results from Attempting to Steal Blinds

Applying filters by steal attempts, meaning raising at the positions: CO, BTN and SB the following results were obtained (table 3):

[21] Stealing situations refer to the specific positions at a table which encourage the act of trying to steal the blinds from other players, they represent a raise from a player from one of those positions when in an unopened pot.

Table 3 – Results from steal attempts

Hands	Winnings	EV € (adjusted)	Avg All-in EV%
102	3.59€	1.81€	55.1%

These results are extremely positive, since the objective here is stealing blinds while taking into account the fold chance of the opponents. When the steal attempt fails, the most common scenario is being behind the opponents range, but this is not verified, since the average all-in percentage when the agent fails to steal is higher than 50%, thus meaning the agent is still stealing less than he should be. The possibility for opening the range of stealing and having an average all-in percentage between 48% and 49% is still viable since some money is won when the opponents fold. These results show the high importance the steal factor has in the poker game. Figure 24 shows the graphical representation of these hands.

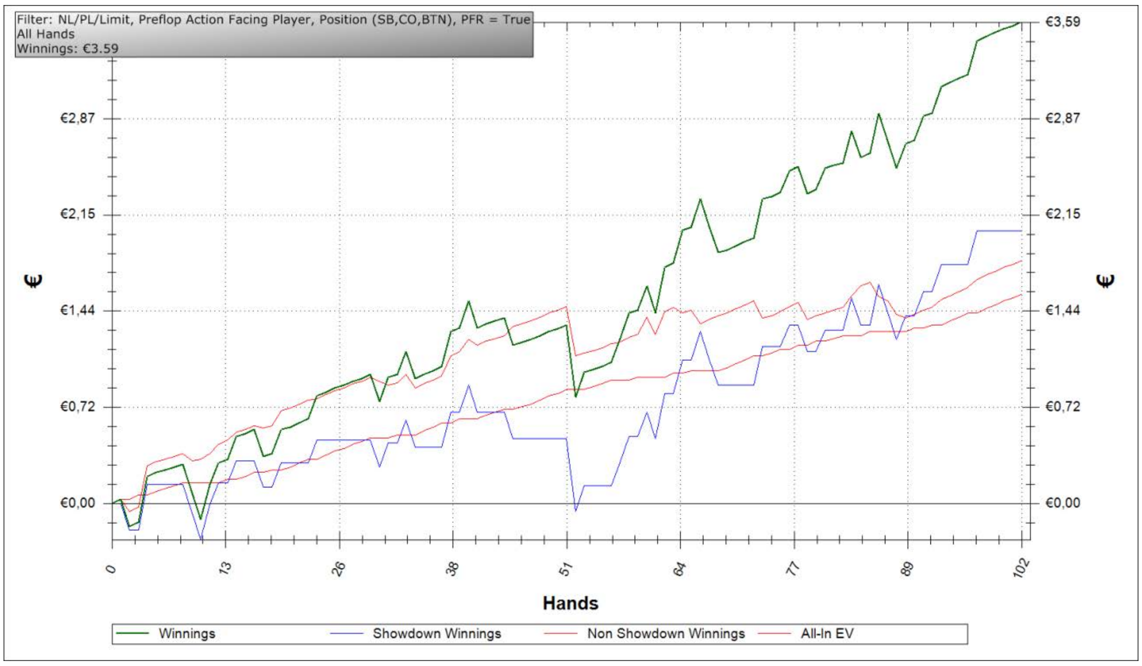


Figure 24 – Winnings from steal attempts

As expected the non-showdown winnings, represented by one of the red lines, is positive along with all the other lines.

5.2.5 Results from Attempting to Defend Blinds

The next table (table 4) will show the results obtained while attempting to defend the blinds, meaning the agent is only positioned at the small blind or the big blind, and tries to minimize his losses by not folding.

Table 4 – Results from attempting to defend the blinds

Hands	Winnings	EV € (adjusted)	Avg All-in EV%
51	0.51€	1.60€	49.3%

Here we see that the calculations for expected value where fairly accurate since among all the all-ins made, the average all-in expected value percentage is 49.3%, meaning when the agent gets called down will still have a good percentage of winning, and when he doesn't get called he wins the blinds plus the raise the opponent made. The expected value from these plays is higher than the actual winnings, this mean that the agent played well, despite of the variance not being on his side. Nevertheless it is still a small amount of hands, and in the long run the winnings will even out with the expected value. It is a very satisfying expected value of 1.60€, since the agent bets 0.20€ at a time. The figure 25 shows the graphical representation of these results.

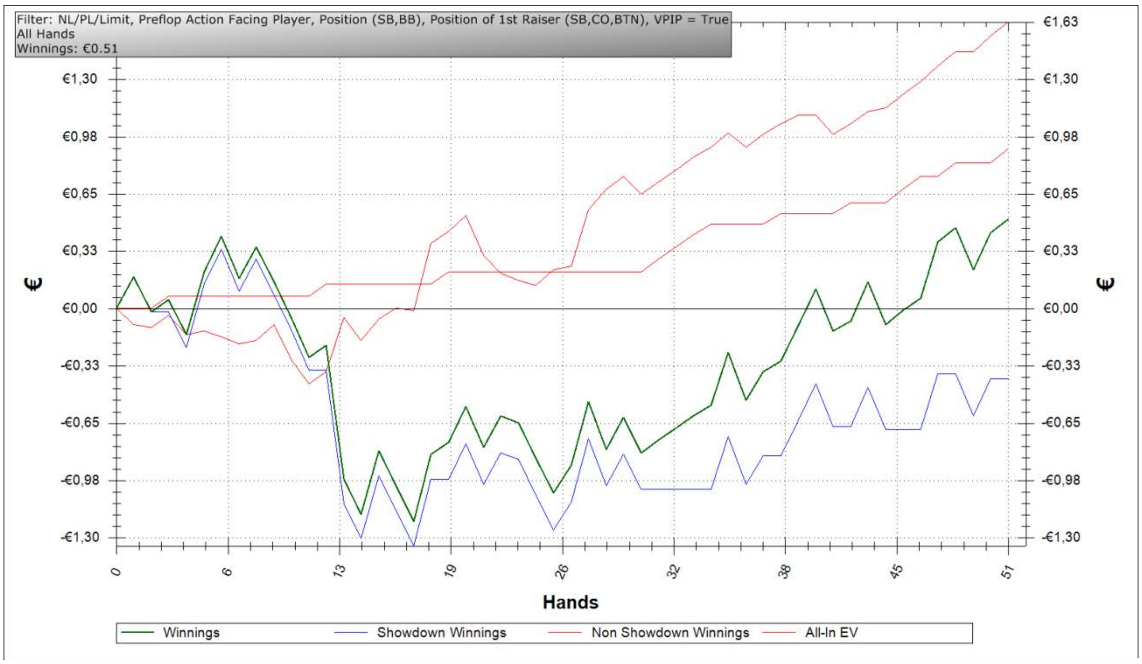

Filter: NL/PL/Limit, Preflop Action Facing Player, Position (SB,BB), Position of 1st Raiser (SB,CO,BTN), VPIP = True
All Hands
Winnings: €0.51

Figure 25 – Winnings from attempting to defend the blinds

Analyzing the figure, we can see the expected value as being the highest followed by the non-showdown winnings. While the actual winnings only show a profit nearly at 40 hands. This was due to some variance at the beginning, specifically near the 13th hand played.

The next result sets will be about the top most profitable players the agent played against and the most unprofitable ones. Table 5 shows the most profitable players and their stats while table 6 shows the most unprofitable ones.

5.2.6 Results for Top 10 Most Profitable and Unprofitable Players

Table 5 – Top 10 most profitable human players played against

Opponent	Hands	VPIP	PFR	Winnings
Player1	14	42.9%	28.6	1.49€
Player2	**271**	**17.7%**	**13.7%**	**1.00€**
Player3	14	64.3%	21.4%	0.97€
Player4	41	82.9%	9.8%	0.79€
Player5	39	41.0%	12.8%	0.76€
Player6	5	40.0%	0.0%	0.73€
Player7	16	31.3%	18.8%	0.69€
Player8	45	57.8%	0.0%	0.62€
Player9	**455**	**24.2%**	**11.2%**	**0.60€**
Player10	**860**	**19.8%**	**16.3%**	**0.56€**

The most significant players to note here are the ones which have a number of hands higher than 100, namely: Player2, Player9 and Player10. These three players show pre-flop statistics fairly good, there are tight aggressive players, and still the agent was able to exploit them and win money over time.

Table 6 – Top 10 most unprofitable human players played against

Opponent	Hands	VPIP	PFR	Winnings
Player1	54	55.6%	25.9	-1.07€
Player2	180	31.1%	12.2%	-0.86€
Player3	38	84.2%	23.7%	-0.62€
Player4	148	26.4%	23.0%	-0.60€
Player5	277	27.8%	19.9%	-0.59€
Player6	67	22.4%	20.9%	-0.59€
Player7	136	20.6%	16.2%	-0.56€

Player8	224	20.5%	19.2%	-0.56€
Player9	25	72.0	40.0	-0.46€
Player10	774	15.0%	13.0%	-0.46€

Looking at the top five most unprofitable opponents, their stats vary from a very tight aggressive player, Player5, to a very loose aggressive player, Player3. A quick look at the hands played against these players can verify if the agent made any bad plays or these losses are just associated to variance. The following tables will show the hands each player had along with the agent's expected value and its actual winnings.

Table 7 – Agent versus Player1

Agent vs Player1	Agent's hand	Player1's hand	Agent's All-in %	Winnings
1st hand	As-Kc	Kd-Js	74.4%	-0.76€
2nd hand	Jd-Jh	Ad-4d	67.1%	-0.20€
3rd hand	Qs-Qc	Js-9c	86.0%	0.20€

The agent's decisions were extremely good; variance is the one to blame for the losses. This behavior will eventually turn into huge profit after a large amount of repetitions. It is clear that the agent has this player completely dominated. The first hand was an extremely unlucky time to receive such a good hand when having such a big stack at the table. Probably the agent had won a big pot just before receiving this hand, and then lost it to variance, meaning the equity was in its favor but due to fluctuations in probability the outcome was the unlikely one.

Table 8 – Agent versus Player2

Agent vs Player2	Agent's hand	Player2's hand	Agent's All-in %	Winnings
1st hand	Ad-As	Ks-Kh	81.9%	-0.43€
2nd hand	6c-6h	7d-7h	18.4%	-0.19€

Once again the first hand was bad luck attributed to variance, and also unlucky for the agent to have more than 20 big blinds at that moment, which increased the variance from the play. The second hand could be discussed as being a good or a bad play; it would depend on the agent's assessment of his position at the table and the opponent's ranges. The two hands are too close to each other to take any conclusions about the play.

Table 9 – Agent versus Player3

Agent vs Player3	Agent's hand	Player3's hand	Agent's All-in %	Winnings
1st hand	Ac-Kc	As-Th	75.0%	-0.23€
2nd hand	As-Jc	Qd-9h	62.9%	-0.20€
3rd hand	8h-8s	Qs-9d	55.3%	-0.20€

Seems the agent is making the right decisions, and is ahead of his opponent on all three hands, unlucky all three were lost. Continuing to present such good all-in percentages will eventually and undoubtedly turn into profit.

Table 10 – Agent versus Player4

Agent vs Player4	Agent's hand	Player4's hand	Agent's All-in %	Winnings
1st hand	Qh-Qc	9c-8c	79.0%	-0.27€

Another unlucky turn out of event, but it was only a 0.27 € loss from this hand. Since the agent lost a total of 0.60€ to this player it could mean the raises from him where too frequent and won the rest of the amount missing from stealing the blinds from the agent. This was a behavior the agent couldn't counter, at least in this small amount of hands. Maybe in the long run the agent could at least maintain an even win rate with this player.

Table 11 – Agent versus Player5

Agent vs Player5	Agent's hand	Player5's hand	Agent's All-in %	Winnings
1st hand	Ac-Kh	9d-9s	44.7%	-0.20€
2nd hand	Ah-Ks	9c-9d	44.7%	-0.20€
3rd hand	Ac-Ts	Kc-Ad	26.3%	-0.20€

This last player is a solid tight aggressive player, and for all these all-in actions the agent was behind every time, the amount of hands is low to conclude this is a bad behavior. Looking at the Holdem Manager hand's history the first two hands are standard plays and considerate good in the situation the agent was in, CO position and BB position. For the last hand the behavior can be corrected to not trying to defend blinds when the game state has one raiser at an early position and more than one caller, when the early position raiser is a tight aggressive player. His range at early position is too strong for the agent to try and defend a weak hand like Ace Ten.

Overall it can be concluded that the agent is performing correct evaluations on hand plays, game states and positional awareness. The profit shown is promising of a winning player at the 0.02/0.01€ No Limit Texas Hold'em.

Conclusions and Future Work

6.1 Conclusions

This work involved some background knowledge on the poker game, specifically Texas Hold'em. The development of this work had the advices of a professional poker player, Michel Dattani, and my co-supervisor by integrating some valuable code for combinatorial calculations.

By the implementation of a strategy definition algorithm which relies on players' statistical information gathered from Holdem Manager Database, a poker player agent was developed and the interface to play online poker as well. The results of this study were very positive and prove evidence that a computer poker player is able to beat human players at an online poker room betting real money. The objective of playing on an online poker room was fulfilled as well as the goal to stand break even against human players. The most surprising aspect was the agent surpassing most of the human players found online by being profitable.

The success of the agent at this limit proves its superiority, possible studies and improvements might increase its ability to evaluate poker plays. Ideally it will be able to play at higher stakes where the skill of the players increase and hopefully still be a profitable poker agent.

6.2 Future work

Some suggestions for future work would be improvement of the blind stealing ability, on the three positions fit to do so: big blind, small blind and button. The agent can be improved in the matter of autonomy at the tables, for instance, leaving a table when holding more than 20 big blinds, entering a new table where the minimum players is 4, leaving a table when it falls below 4 players. Some user interaction could be implemented like emailing the agent's winnings periodically. The best and most challenging improvement to the agent would be playing a full-stack[22] game of poker, having a huge amount of responses to just one information set and evaluating the best one for that hand, player, pot size and position. I believe such agent is not as far as it seems.

[22] Playing at a poker table with the maximum amount possible according the table's limit

References

[1] B. Hoehn, “The effectiveness of opponent modelling in a small imperfect information game,” Citeseer, 2006.

[2] G. Nicolai and R. J. Hilderman, No-Limit Texas Hold’em Poker agents created with evolutionary neural networks. Ieee, 2009, pp. 125–131.

[3] I. Schweizer and K. Panitzek, “An exploitative Monte-Carlo poker agent,” KI 2009: Advances in …, 2009.

[4] N. Risk and D. Szafron, “Using counterfactual regret minimization to create competitive multiplayer poker agents,” … International Conference on Autonomous Agents …, no. May, pp. 10–14, 2010.

[5] Luís Filipe Teofilo, L. P. Reis, and H. L. Cardoso, “Computer Poker Research at LIACC,” in Computer Poker Symposium, 2012.

[6] M. Abràmoff, “Image processing with ImageJ,” Biophotonics …, 2004.

[7] D. Billings, “Algorithms and assessment in computer poker,” 2006.

[8] D. Billings, D. Papp, J. Schaeffer, and D. Szafron, “Poker as a Testbed for Machine Intelligence Research,” Artificial Intelligence, no. May 1997, pp. 1–15, 1998.

[9] Zinkevich, Martin; Johanson, Michael ; Bowling, Michael ; Piccione, C. "Regret Minimization in Games with Incomplete Information "(p. 14), 2007.

[10] Open Meerkat Poker Testbed. 2012. http://code.google.com/p/opentestbed/

[11] Poker Academy Pro -The Ultimate Poker Software. 2012. http://www.poker-academy.com

[12] Littman, M., Zinkevich, M. September 2006. "The 2006 AAAI Computer Poker Competition". Journal of International Computer Games Association. pp 166-16

[13] Hold’em Manager. 2012. http://www.holdemmanager.com/

[14] WinHold’em. 2012. http://www.winholdem.net/

[15] PokerTracker. 2012. http://www.pokertracker.com/

[16] Haruyoshi Sakai, “Internet Poker: Data Collection and Analysis”, 2005.

[17] Holdem Manager Forums. 2012. http://forums.holdemmanager.com/manager-general/

[18] Paulo Martins, L. P. Reis, and Luís Teófilo, “Poker Vision: Playing Cards and Chips Identification Based on Image Processing,” Lecture Notes in Computer Science Volume 6669, 2011, pp 436-443.

[19] Poker Terms, http://pokerterms.com/letter/a.php.

Appendix A

Holdem Manager

In this section it can be found some information regarding the Holdem Manager Software.

A.1 Contents of the text document for SQL queries

The content of this text document was very useful in the elaboration of various queries; a small portion of it is displayed below.

HM1StatsDefinitions.txt portion:

```
…<Stat GroupName="Default"
ColumnName="bbPer100"
ValueExpressions="Sum(ph.NetAmountWon/1.0/GT.BigBlind) as TotalBBs"
Evaluate="TotalBBs*100.0/TotalHands"
ColumnHeader="bb/100"
ColumnFormat="0.00"
ColumnWidth="*"
Tooltip="Big Blinds won per 100 hands" />…
```

A.2 Complete Holdem Manager Query

Here it is displayed the complete query elaborated for obtaining the players' statistical informations:

```
    select localtimestamp, playername as player, count(ph.*) as
totalhands,
    round(sum(case when didpfr = true
    AND preflopaction_id = 0
    AND positiontype_id = 0
    then 1 else 0 end) / 1.00 / (sum(case when positiontype_id =
0 AND preflopaction_id = 0 then 1 else 0 end).00001)*100,1) as
SB,  //PFR for small blind position
```

```
    round(sum(case when didpfr = true
    AND preflopaction_id = 0
    AND positiontype_id = 1
    then 1 else 0 end) / 1.00 / (sum(case when positiontype_id =
1 AND preflopaction_id = 0 then 1 else 0 end).00001)*100,1) as
BB, // PFR for big blind position

    round(sum(case when didpfr = true
    AND preflopaction_id = 0
    AND positiontype_id = 2
    then 1 else 0 end) / 1.00 / (sum(case when positiontype_id =
2 AND preflopaction_id = 0 then 1 else 0 end).00001)*100,1) as
UTG, //PFR for early position

    round(sum(case when didpfr = true
    AND preflopaction_id = 0
    AND positiontype_id = 3
    then 1 else 0 end) / 1.00 / (sum(case when positiontype_id =
3 AND preflopaction_id = 0 then 1 else 0 end).00001)*100,1) as
UTG1, //PFR for midle position

    round(sum(case when didpfr = true
    AND preflopaction_id = 0
    AND positiontype_id = 4
    then 1 else 0 end) / 1.00 / (sum(case when positiontype_id =
4 AND preflopaction_id = 0 then 1 else 0 end).00001)*100,1) as
Co, //PFR for cuttoff position


    round(sum(case when didpfr = true
    AND preflopaction_id = 0
    AND positiontype_id = 5
    then 1 else 0 end) / 1.00 / (sum(case when positiontype_id =
5 AND preflopaction_id = 0 then 1 else 0 end).00001)*100,1) as
Button, //PFR for button position

    sum(case when positiontype_id = 0 and
phmisc.threebetresponsetype_id = 1 then 1 else 0 end) as
```

```
FoldToThreeBet_SB, //number of times folded to 3bet at small
blind

   sum(case when positiontype_id = 0 and
phmisc.threebetresponsetype_id > -1 then 1 else 0 end) as
FacingThreeBet_SB, //number of times facing 3bet at small blind

   sum(case when positiontype_id = 1 and
phmisc.threebetresponsetype_id = 1 then 1 else 0 end) as
FoldToThreeBet_BB, //number of times folded to 3bet at big blind

   sum(case when positiontype_id = 1 and
phmisc.threebetresponsetype_id > -1 then 1 else 0 end) as
FacingThreeBet_BB, //number of times facing 3bet at big blind

   sum(case when positiontype_id = 2 and
phmisc.threebetresponsetype_id = 1 then 1 else 0 end) as
FoldToThreeBet_UTG, //number of times folded to 3bet at early
position

   sum(case when positiontype_id = 2 and
phmisc.threebetresponsetype_id > -1 then 1 else 0 end) as
FacingThreeBet_UTG, //number of times facing 3bet at early
position

   sum(case when positiontype_id = 3 and
phmisc.threebetresponsetype_id = 1 then 1 else 0 end) as
FoldToThreeBet_UTG1, //number of times folded to 3bet at midle
position

   sum(case when positiontype_id = 3 and
phmisc.threebetresponsetype_id > -1 then 1 else 0 end) as
FacingThreeBet_UTG1, //number of times facing 3bet at midle
position

   sum(case when positiontype_id = 4 and
phmisc.threebetresponsetype_id = 1 then 1 else 0 end) as
FoldToThreeBet_Co, //number of times folded to 3bet at cutoff
```

```
    sum(case when positiontype_id = 4 and
phmisc.threebetresponsetype_id > -1 then 1 else 0 end) as
FacingThreeBet_Co, //number of times facing 3bet at cuttoff

    sum(case when positiontype_id = 5 and
phmisc.threebetresponsetype_id = 1 then 1 else 0 end) as
FoldToThreeBet_Button, //number of times folded to 3bet at
button

    sum(case when positiontype_id = 5 and
phmisc.threebetresponsetype_id > -1 then 1 else 0 end) as
FacingThreeBet_Button, //number of times facing 3bet at button

    sum(ph.NetAmountWon/1.0/GT.BigBlind) as TotalBBs,  //number
of big blinds won

    round(avg(case when didvpip = true
    then 1 else 0 end)*100,1) as vpip,
    round(avg(case when didpfr = true
    then 1 else 0 end)*100,1) as pfr
    from playerhandscashkeycolumns ph join players pl
    on (pl.player_id = ph.player_id)
    join pokerhands pkh on pkh.pokerhand_id = ph.pokerhand_id
    join playerhandscashmisc phmisc on phmisc.playerhand_id =
ph.playerhand_id
    join gametypes gt on gt.gametype_id = ph.gametype_id left
    join playerhandsriver river on ph.playerhand_id =
river.playerhand_id
    where pl.lastplayeddate > (localtimestamp - interval '10
minutes')
    Group by playername ORDER BY localtimestamp DESC
```

A.3 Holdem Manager Introduction

Detailed information on the Holdem Manager Software can be found at the forums: http://faq.holdemmanager.com/questions/134/Holdem+Manager+Introduction.

A.4 Poker Glossary [19]

Here some Poker terms are described:

Action

(1) Opportunity to act. If a player appears not to realize it's his turn, the dealer will say "Your action, sir."

(2) Bets and raises. "If a third heart hits the board and there's a lot of action, you have to assume that somebody has made the flush."

Ante

A small portion of a bet contributed by each player to seed the pot at the beginning of a poker hand. Most hold'em games do not have an ante; they use "blinds" to get initial money into the pot.

All-In

To run out of chips while betting or calling. In table stakes games, a player may not go into his pocket for more money during a hand. If he runs out, a side pot is created in which he has no interest. However, he can still win the pot for which he had the chips. Example: "Poor Bob. He made quads against the big full house, but he was all-in on the second bet."

Backdoor

Catching both the turn and river card to make a drawing hand. For instance, suppose you have As-7s. The flop comes Ad-6c-4s. You bet and are called. The turn is the Ts, which everybody checks, and then the river is the Js. You've made a "backdoor" nut flush. See also "runner."

Bad Beat

To have a hand that is a large underdog beat a heavily favored hand. It is generally used to imply that the winner of the pot had no business being in the pot at all, and it was the wildest of luck that he managed to catch the one card in the deck that would win the pot. We won't give any examples; you will hear plenty of them during your poker career.

Big Blind

The larger of the two blinds typically used in a hold'em game. The big blind is a full first round bet. See also "blind" and "small blind."

Big Slick

A nickname for AK (suited or not). Its origins are unknown (to me, anyway).

Blank

A board card that doesn't seem to affect the standings in the hand. If the flop is As-Jd-Ts, then a turn card of 2h would be considered a blank. On the other hand, the 2s would not be.

Blind

A forced bet (or partial bet) put in by one or more players before any cards are dealt. Typically, blinds are put in by players immediately to the left of the button. See also "live blind."

Board

All the community cards in a hold'em game -- the flop, turn, and river cards together. Example: "There wasn't a single heart on the board."

Bot

Short for "robot". In a poker context, a program that plays poker online with no (or minimal) human intervention.

Bottom Pair

A pair with the lowest card on the flop. If you have As-6s, and the flop comes Kd-Th-6c, you have flopped bottom pair.

Brick & Mortar

A "real" casino or cardroom with a building, tables, dealers, etc. This is in contrast to an online poker site.

Bubble

(1) The point at which only one player must bust out before all others win some money. (2) The person who was unfortunate enough to finish in that position.

Burn

To discard the top card from the deck, face down. This is done between each betting round before putting out the next community card(s). It is security against any player recognizing or glimpsing the next card to be used on the board.

Button

A white acrylic disk that indicates the (nominal) dealer. Also used to refer to the player on the button. Example: "Oh, the button raised."

Buy

(1) As in "buy the pot." To bluff, hoping to "buy" the pot without being called. (2) As in "buy the button." To bet or raise, hoping to make players between you and the button fold, thus allowing you to act last on subsequent betting rounds.

Buy-In

An amount of money you pay to enter a tournament. Often expressed as two numbers, such as $100+9, meaning that it costs $109 to enter the tournament; $100 goes into the prize fund and $9 goes to the house.

Call

To put into the pot an amount of money equal to the most recent bet or raise. The term "see" (as in "I'll see that bet") is considered colloquial.

Calling Station

A weak-passive player who calls a lot, but doesn't raise or fold much. This is the kind of player you like to have in your game.

Cap

To put in the last raise permitted on a betting round. This is typically the third or fourth raise. Dealers in California are fond of saying "Capitola" or "Cappuccino."

Case

The last card of a certain rank in the deck. Example: "The flop came J-8-3; I've got pocket jacks, he's got pocket 8's, and then the case eight falls on the river, and he beats my full house."

Center Pot

The first pot created during a poker hand, as opposed to one or more "side" pots created if one or more players goes all-in. Also "main pot."

Chat

Typed conversation that you can have with other players at an online poker site (or any online gathering, for that matter).

Check

(1) To not bet, with the option to call or raise later in the betting round. Equivalent to betting zero dollars. (2) Another word for chip, as in poker chip.

Check-Raise

To check and then raise when a player behind you bets. Occasionally you will hear people say this is not fair or ethical poker. Piffle. Almost all casinos permit check-raising, and it is an important poker tactic. It is particularly useful in low-limit hold'em where you need extra strength to narrow the field if you have the best hand.

Chop

An agreement between the two players with blinds to simply take their blinds back rather than playing out the hand if nobody calls or raises in front of them.

Clean Out

A card that would almost certainly make your hand best. If you are drawing at a straight, but there is a flush draw possible, then the cards that make your straight but also the flush are not clean outs.

Cold Call

To call more than one bet in a single action. For instance, suppose the first player to act after the big blind raises. Now any player acting after that must call two bets "cold." This is different from calling a single bet and then calling a subsequent raise.

Come Hand

A drawing hand (from the craps term).

Complete Hand

A hand that is defined by all five cards -- a straight, flush, full house, four of a kind, or straight flush.

Connector

A hold'em starting hand in which the two cards are one apart in rank. Examples: KQs, 76.

Counterfeit

To make your hand less valuable because of board cards that duplicate it. Example: you have 87 and the flop comes 9-T-J, so you have a straight. Now an 8 comes on the turn. This has counterfeited your hand and made it almost worthless.

Crack

To beat a hand -- typically a big hand. You hear this most often applied to pocket aces: "Third time tonight I've had pocket aces cracked."

Cripple

As in "to cripple the deck." Meaning that you have most or all of the cards that somebody would want to have with the current board. If you have pocket kings, and the other two kings flop, you have crippled the deck.

Crying Call

A call that you make expecting to lose, but feel that you must make anyway because of the pot odds.

Cut-Off

The position (or player) who acts one before the button.

Dead Money

(1) Money contributed to a pot by a player no longer in the pot. (2) A player in a tournament who has no realistic chance of winning.

Dog

Shortened form of "underdog."

Dominated Hand

A hand that will almost always lose to a better hand that people usually play. For instance, K3 is "dominated" by KQ. With the exception of strange flops (e.g., 3-3-X, K-3-X), it will always lose to KQ.

Draw

To play a hand that is not yet good, but could become so if the right cards come. Example: "I'm not there yet -- I'm drawing." Also used as a noun. Example: "I have to call because I have a good draw."

Draw Dead

Trying to make a hand that, even if made, will not win the pot. If you're drawing to make a flush, and your opponent already has a full house, you are "drawing dead." Of course, this is a bad condition to be in.

Equity

Your "rightful" share of a pot. If the pot contains $80, and you have a 50% chance of winning it, you have $40 equity in the pot. This term is somewhat fanciful since you will either win $80 or $0, but it gives you an idea of how much you can "expect" to win.

Expectation

(1) The amount you expect to gain on average if you make a certain play. For instance, suppose you put $10 into a $50 pot to draw at a hand that you will make 25% of the time, and it will win every time you make it. Three out of four times, you do not make your draw, and lose $10 each time for a total of $30. The fourth time, you will make your draw, winning $50. Your total gain over those four average hands is $50-$30 = $20, an average of $5 per hand. Thus calling the $10 has a positive expectation of $5. (2) The amount you expect to make at the poker table in a specific time period. Suppose in 100 hours of play, you win $527. Then your expectation is $5.27/hr. Of course, you won't make that exact amount each hour (and some hours you will lose), but it's one measure of your anticipated earnings.

Extra Blind

A blind put in by a player just entering the game, returning to the game, or otherwise changing his position at the table. See also "blind" and "post."

Family Pot

A pot in which all (or almost all) of the players call before the flop.

Fast Play

To play a hand aggressively, betting and raising as much as possible. Example: "When you flop a set but there's a flush draw possible, you have to play it fast."

Fish

A poor player -- one who gives his money away. It's a well-known (though not well-followed) rule among good players to not upset the bad players, because they'll stop having fun and perhaps leave. Thus the phrase, "Don't tap on the aquarium."

Flop

The first three community cards, put out face up, all together.

Fold Equity

The extra value you get from a hand when you force an opponent to fold. That is, if you don't have to see a showdown, your hand has more value than if you do.

Foul

A hand that may not be played for one reason or another. A player with a foul hand may not make any claim on any portion of the pot. Example: "He ended up with three cards after the flop, so the dealer declared his hand foul."

Free Card

A turn or river card on which you don't have to call a bet because of play earlier in the hand (or because of your reputation with your opponents). For instance, if you are on the button and raise when you flop a flush draw, your opponents may check to you on the turn. If you make your flush on the turn, you can bet. If you don't get it on the turn, you can check as well, seeing the river card for "free."

Free Roll

One player has a shot at winning an entire pot when he is currently tied with another player. For instance, suppose you have Ac-Qc and your opponent has Ad-Qh. The flop is Qs-5c-Tc. You are tied with your opponent right now, but are free rolling, because you can win the whole pot and your opponent can't. If no club comes, you split the pot with him; if it does come, you win the whole thing.

Gap Hand

A starting hand with cards more than one rank apart. For instance, T9 is a one-gap hand. 86 is a two-gap hand.

Gutshot Straight

A straight filled "inside." If you have 9s-8s, the flop comes 7c-5h-2d, and the turn is the 6c, you've made your gutshot straight.

Heads-Up

A pot that is being contested by only two players. Example: "It was heads-up by the turn."

Hit

As in "the flop hit me," meaning the flop contains cards that help your hand. If you have AK, and the flop comes K-7-2, it hit you.

House

The establishment running the game. Example: "The $2 you put on the button goes to the house."

Implied Odds

Pot odds that do not exist at the moment, but may be included in your calculations because of bets you expect to win if you hit your hand. For instance, you might call with a flush draw on the turn even though the pot isn't offering you quite 4:1 odds (your chance of making the flush) because you're sure you can win a bet from your opponent on the river if you make your flush.

Jackpot

A special bonus paid to the loser of a hand if he gets a very good hand beaten. In hold'em, the "loser" must typically get aces full or better beaten. In some of the large southern California card clubs, jackpots have gotten over $50,000. Of course, the jackpot is funded with money removed from the game as part of the rake.

Jam

To move all-in in a no-limit (or pot-limit) game.

Kicker

An unpaired card used to determine the better of two near-equivalent hands. For instance, suppose you have AK and your opponent has AQ. If the flop has an ace in it, you both have a pair of aces, but you have a king kicker. Kickers can be vitally important in hold'em.

Leak

A weakness in your game that causes you to win less money than you would otherwise. Example: "She takes her pocket pairs too far; it's a leak in her game."

Limp

To call. Generally the term refers to pre-flop action. For instance: "He limped in early position with 77."

Live Blind

A forced bet put in by one or more players before any cards are dealt. The "live" means those players still have the option of raising when the action gets back around to them.

Live

Cards that are not duplicated in an opponent's stronger hand. For example, if you have A9 and your opponent has AJ, then your ace is not "live" because making a pair of aces won't do you any good. The nine, however, is live; making a pair of nines gives you the better hand.

Maniac

A player who does a lot of hyper-aggressive raising, betting, and bluffing. A true maniac is not a good player, but is simply doing a lot of gambling. However, a player who occasionally acts like a maniac and confuses his opponents is quite dangerous.

Made Hand

A hand to which you're drawing, or one good enough that it doesn't need to improve.

Micro-Limit

Games so small that they couldn't be profitably dealt in a real cardroom. They exist only at online poker sites. You might arbitrarily call games $.25-.50 and smaller "micro-limit."

Muck

The pile of folded and burned cards in front of the dealer. Example: "His hand hit the muck so the dealer ruled it folded even though the guy wanted to get his cards back." Also used as a verb. Example: "He didn't have any outs so he mucked his hand."

No-Limit

A version of poker in which a player may bet any amount of chips (up to the number in front of him) whenever it is his turn to act. It is a very different game from limit poker.

Nuts

The best possible hand given the board. If the board is Ks-Jd-Ts-4s-2h, then As-Xs is the nuts. You will occasionally hear the term applied to the best possible hand of a certain category, even though it isn't the overall nuts. For the above example, somebody with Ah-Qc might say they had the "nut straight."

Offsuit

A hold'em starting hand with two cards of different suits.

One-Gap

A hold'em starting hand with two cards two apart in rank. Examples: J9s, 64.

Out

A card that will make your hand win. Normally heard in the plural. Example: "Any spade will make my flush, so I have nine outs."

Outrun

To beat. Example: "Susie outran my set when her flush card hit on the river."

Overcall

To call a bet after one or more others players have already called.

Overcard

A card higher than any card on the board. For instance, if you have AQ and the flop comes J-7-3, you don't have a pair, but you have two overcards.

Overpair

A pocket pair higher than any card on the flop. If you have QQ and the flop comes J-8-3, you have an overpair.

Pat

A hand that you make on the flop. For instance, if you have two spades in your hand and the flop has three spades, then you've flopped a pat spade flush.

Pay Off

To call a bet when the bettor is representing a hand that you can't beat, but the pot is sufficiently large to justify a call anyway. Example: "He played it exactly like he made the flush, but I had top set so I paid him off."

Play the Board

To show down a hand in hold'em when your cards don't make a hand any better than is shown on the board. For instance, if you have 22, and the board is 4-4-9-9-A (no flush possible), then you must "play the board": the best possible hand you can make doesn't use any of your cards. Note that if you play the board, the best you can do is split the pot with all remaining players.

Pocket

Your unique cards that only you can see. For instance, "He had pocket sixes" (a pair of sixes), or "I had ace-king in the pocket."

Pocket Pair

A hold'em starting hand with two cards of the same rank, making a pair. Example: "I had big pocket pairs seven times in the first hour. What else can you ask for?"

Post

To put in a blind bet, generally required when you first sit down in a cardroom game. You may also be required to post a blind if you change seats at the table in a way that moves you away from the blinds. Example: a player leaves one seat at a table and takes another in such a way that he moves farther from the blinds. He is required to post an extra blind to receive a hand. See also "extra blind."

Pot-Committed

A state where you are essentially forced to call the rest of your stack because of the size of the pot and your remaining chips.

Pot-Limit

A version of poker in which a player may bet up to the amount of money in the pot whenever it is his turn to act. Like no-limit, this is a very different game from limit poker.

Pot Odds

The amount of money in the pot compared to the amount you must put in the pot to continue playing. For example, suppose there is $60 in the pot. Somebody bets $6, so the pot now contains $66. It costs you $6 to call, so your pot odds are 11:1. If your chance of having the best hand is at least 1 out of 12, you should call. Pot odds also apply to draws. For instance, suppose you have a draw to the nut flush with one card left to come. In this case, you are about a 4:1 underdog to make your flush. If it costs you $8 to call the bet, then there must be about $32 in the pot (including the most recent bet) to make your call correct.

Price

The pot odds you are getting for a draw or call. Example: "The pot was laying me a high enough price, so I stayed in with my gutshot straight draw."

Protect

(1) To keep your hand or a chip on your cards. This prevents them from being fouled by a discarded hand, or accidentally mucked by the dealer. (2) To invest more money in a pot so blind money that you've already put in isn't "wasted." Example: "He'll always protect his blinds, no matter how bad his cards are."

Put On

To mentally assign a hand to a player for the purposes of playing out your hand. Example: "He raised on the flop, but I put him on a draw, so I re-raised and then bet the turn."

Quads

Four of a kind.

Ragged

A flop (or board) that doesn't appear to help anybody very much. A flop that came down Jd-6h-2c would look ragged.

Rainbow

A flop that contains three different suits, thus no flush can be made on the turn. Can also mean a complete five card board that has no more than two of any suit, thus no flush is possible.

Rake

An amount of money taken out of every pot by the dealer. This is the cardroom's income.

Rank

The numerical value of a card (as opposed to its suit). Example: "jack," "seven."

Rebuy

An option to buy back into a tournament after you've lost all your chips. Tournaments may offer one or more rebuys or (often) none at all.

Represent

To play as if you hold a certain hand. For instance, if you raised before the flop, and then raised again when the flop came ace high, you would be representing at least an ace with a good kicker.

Ring Game

A regular poker game as opposed to a tournament. Also referred to as a "live" game since actual money is in play instead of tournament chips.

River

The fifth and final community card, put out face up, by itself. Also known as "fifth street." Metaphors involving the river are some of poker's most treasured cliches, e.g., "He drowned in the river."

Rock

A player who plays very tight, not very creatively. He raises only with the best hands. A real rock is fairly predictable: if he raises you on the river, you can throw away just about anything but the nuts.

Runner

Typically said "runner-runner" to describe a hand that was made only by catching the correct cards on both the turn and the river. Example: "He made a runner-runner flush to beat my trips." See also "backdoor."

Satellite

A tournament that does not award cash to its winners, but a seat (or seats) in a subsequent "target" tournament.

Scare Card

A card that may well turn the best hand into trash. If you have Tc-8c and the flop comes Qd-Jd-9s, you almost assuredly have the best hand. However, a turn card of Td would be very scary because it would almost guarantee that you are now beaten.

Second Pair

A pair with the second highest card on the flop. If you have As-Ts, and the flop comes Kd-Th-6c, you have flopped second pair. See "top pair."

Sell

As in "sell a hand." In a spread-limit game, this means betting less than the maximum when you have a very strong hand, hoping players will call whereas they would not have called a maximum bet.

Semi-Bluff

A powerful concept first discussed by David Sklansky. It is a bet or raise that you hope will not be called, but you have some outs if it is. A semi-bluff may be correct when betting for value is not correct, a pure bluff is not correct, but the combination of the two may be a positive expectation play. Example: you have Ks-Qs, and the flop is Th-5s-Jc. If you bet now, it's a semi-bluff. You probably don't have the best hand, and you'd like to see your opponents fold immediately. Nevertheless, if you do get callers, you could still improve to the best hand.

Set

Three of a kind when you have two of the rank in your hand, and there is one on the board.

Short Stack

A number of chips that is not very many compared to the other players at the table. If you have $10 in front of you, and everybody else at the table has over $100, you are playing on a short stack.

Showdown

The point at which all players remaining in the hand turn their cards over and determine who has the best hand -- i.e., after the fourth round of betting is completed. Of course, if a final bet or raise is not called, there is no showdown.

Side Pot

A pot created in which a player has no interest because he has run out of chips. Example: Al bets $6, Beth calls the $6, and Carl calls, but he has only $2 left. An $8 side pot is created that either Al or Beth can win, but not Carl. Carl, however, can still win all the money in the original or "center" pot.

Slow Play

To play a strong hand weakly so more players will stay in the pot.

Small Blind

The smaller of two blind bets typically used in a hold'em game. Normally, the small blind is one-third to two-thirds of a first round bet. See also "big blind" and "blind."

Smooth Call

To call. Smooth call often implies slow playing a strong hand. Example: "I flopped the nut flush but just smooth called when the guy in front of me bet -- I didn't want to scare anybody out."

Soft-Play

To go easy on another player at the table (e.g., not betting or raising against him). Suppose you and your brother are the last two people left in a hand. On the river, you have the nuts, but he bets. If you don't raise, you are "soft-playing" him. Please note that soft-playing is prohibited in tournaments and can result in penalties, up to and including forfeiture of winnings.

Splash the Pot

To toss chips directly into the pot rather than put them in a stack in front of you. Don't do it.

Split Pot

A pot that is shared by two or more players because they have equivalent hands.

Split Two Pair

A two pair hand in which one of each of your cards' ranks appears on the board as well. Example: you have T9, the flop is T-9-5, you have a split two pair. This is in comparison to two pair where there is a pair on the board. Example: you have T9, the flop is 9-5-5.

Spread-Limit

A betting structure in which a player may bet any amount in a range on every betting round. A typical spread-limit structure is \$2-\$6, where a player may bet as little as \$2 or as much as \$6 on every betting round.

Stop-and-Go

A play where you call (rather than re-raising) a raise, but then come out betting on the next card.

Straddle

An optional extra blind bet, typically made by the player one to the left of the big blind, equal to twice the big blind. This is effectively a raise, and forces any player who wants to play to pay two bets. Furthermore, the straddler acts last before the flop, and may "re-raise."

String Bet

A bet (more typically a raise) in which a player doesn't get all the chips required for the raise into the pot in one motion. Unless he verbally declared the raise, he can be forced to withdraw it and just call. This prevents the unethical play of putting out enough chips to call, seeing what effect that had, and then possibly raising.

Structured

Used to apply to a certain betting structure in poker games. The typical definition of a structured hold'em game is a fixed amount for bets and raises before the flop and on the flop, and then twice that amount on the turn and river. Example: a \$2-\$4 structured hold'em game: bets and raises of \$2 before the flop and on the flop; \$4 bets and raises on the turn and river.

Suited

A hold'em starting hand in which the two cards are the same suit. Example: "I had to play J-3 -- it was suited."

Table Stakes

A rule in a poker game meaning that a player may not go into his pocket for money during a hand. He may only invest the amount of money in front of him into the current pot. If he runs out of chips during the hand, a side pot is created in which he has no interest. All casino poker is played table stakes. The definition sometimes also includes the rule that a player may not remove chips from the table during a game. While this rule might not be referred to as "table stakes," it is enforced almost universally in public poker games.

Tell

A clue or hint that a player unknowingly gives about the strength of his hand, his next action, etc. May originally be from "telegraph" or the obvious use that he "tells" you what he's going to do before he does it.

Thin

As in "drawing thin." To be drawing to a very few outs, perhaps only one or two.

Tilt

To play wildly or recklessly. A player is said to be "on tilt" if he is not playing his best, playing too many hands, trying wild bluffs, raising with bad hands, etc.

Time

(1) A request by a player to suspend play while he decides what he's going to do. Simply, "Time, please!" If a player doesn't request time and there is a substantial amount of action behind him, the dealer may rule that the player has folded. (2) An amount of money collected either on the button or every half hour by the cardroom. This is another way for the house to make its money (see "rake").

To Go

The amount a player must call if he wishes to continue playing. Example: "The big blind was $20. Sarah raised $40 more, making it $60 to go."

Toke

A small amount of money (typically $.50 or $1.00) given to the dealer by the winner of a pot. Quite often, tokes represent the great majority of a dealer's income.

Top Pair

A pair with the highest card on the flop. If you have As-Qs, and the flop comes Qd-Th-6c, you have flopped top pair. See "second pair."

Top Set

The highest possible trips. Example: you have Tc-Ts, and the flop comes Td-8c-9h. You have flopped top set.

Top Two

Two pair, with your two hole cards pairing the two highest cards on the board.

Top and Bottom

Two pair, with your two hole cards pairing the highest and lowest cards on the board.

Trips

Three of a kind.

Turn

The fourth community card. Put out face up, by itself. Also known as "fourth street."

Under the Gun

The position of the player who acts first on a betting round. For instance, if you are one to the left of the big blind, you are under the gun before the flop.

Underdog

A person or hand not mathematically favored to win a pot. For instance, if you flop four cards to your flush, you are not quite a 2:1 underdog to make your flush by the river (that is, you will make your flush about one in three times). See also "dog."

Value

As in "bet for value." This means that you would actually like your opponents to call your bet (as opposed to a bluff). Generally it's because you have the best hand. However, it can also be a draw that, given enough callers, has a positive expectation.

Variance

A measure of the up and down swings your bankroll goes through. Variance is not necessarily a measure of how well you play. However, the higher your variance, the wider swings you'll see in your bankroll.

Wheel

A straight from ace through five.