

Министерство образования Российской Федерации
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (ТГУ)
Факультет информатики
Кафедра теоретических основ информатики

УДК 681.03

ДОПУСТИТЬ К ЗАЩИТЕ В ГАК

Зав. Кафедрой, д.т.н.

Ю. Л. Костюк

2011 г.

Иванов Максим Евгеньевич

**РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЧЕСКОЙ ИГРЫ В ПОКЕР
В ИНТЕРНЕТЕ**

Выпускная квалификационная работа

Научный руководитель

Ассистент

А.В. Петраков

Исполнитель

Студ. Гр. 1471

М.Е. Иванов

Электронная версия дипломной работы помещена

В электронную библиотеку. Файл

Администратор

Томск 2011

Реферат

Выпускная квалификационная работа 44 страницы, 16 рисунков, 6 источников, 2 приложения

ПОКЕРНЫЙ БОТ, ПОКЕР, ТЕОРИЯ ПОКЕРА, АВТОМАТИЗАЦИЯ ИГРЫ, СТРАТЕГИЯ КОРОТКИХ СТЕКОВ.

1. *Объект исследования* - теория покера, покерный клиент.
2. *Цель работы* – проектирование и реализация программного обеспечения для автоматической игры в покер в интернете.
3. *Метод исследования* - теоретический и экспериментальный (на ЭВМ).
4. *Результаты* – Проанализированы стратегии игры в покер, и одна реализована. Реализована программа, автоматически играющая в покер в интернете.

Оглавление

Введение	4
1 Правила покера.....	6
2 Обзор существующих покерных ботов.	9
3 Покерный бот	11
3.1 Что должен уметь покерный бот	11
3.2 Ввод данных	11
3.3 Анализ данных.....	14
3.3.1 Математика покера.....	14
3.3.2 Стратегия коротких стеков.....	15
3.3.3 Классификация оппонентов	18
3.3.4 Стратегия коротких стеков с учетом классификации оппонентов	20
3.3.5 Определение комбинации и шансов улучшить комбинацию	22
4 Программная реализация	26
Руководство пользователя	30
Руководство программиста	31
Заключение	38
Список использованной литературы	39
Приложение А. Список терминов	40
Приложение Б. Диапазоны стартовых рук	43

Введение

В двадцатом веке родился клубный покер, имеющий ныне более десятка разновидностей, среди которых Техасский Холдем, Омаха, Пятикарточный Дро-покер, Семикарточный Стад-покер. Наиболее распространённым из них является Техасский Холдем – первая карточная игра, согласно общеамериканской легенде, появившаяся в городке Робстаун, штата Техас. Из Робстауна Холдем перекочевал в Даллас, а в игорную мекку США, знаменитый Лас-Вегас, его привезли небезызвестные в мире покера Дойл Брансон и Амарилло Слим.

Сейчас Техасский Холдем – официальная игра на турнирах World Series of Poker (WSOP). Впервые этот Чемпионат был проведён в 1970 году и теперь ежегодно собирает тысячи поклонников в стенах казино Horseshoe в Лас-Вегасе.

Популярность Холдема во всём мире неуклонно растёт, и вот покер, некогда бывший *«игрой обманщиков»*, заставляет людей штудировать тома по теории игры, чтобы затем сразиться в мастерстве с соперником из страны на другом конце света.

Покер не смог пройти и мимо интернета. Игры на деньги в онлайн покер начали проводиться в 1998 году на Planet Poker, и популярность онлайн покера росла медленно, но верно к моменту, когда Paradise Poker занял доминирующую позицию на рынке. Крупный скачок произошёл, когда в игру вступили агрессивные маркетинговые программы Pacific Poker и транслируемые телевидением события World Poker Tour в 2002 году. Только в 2003 году объем игроков в онлайн покер увеличился на 500%. Благодаря такому колоссальному росту, в 2004 общий доход в индустрии покера онлайн составил один миллиард долларов США.

Сейчас около 15 всемирно известных покер румов, где можно играть онлайн на деньги. В самых популярных покер румах можно обнаружить более 60 000 человек онлайн.

Пóкер (англ. *poker*) — карточная игра, цель которой — выиграть ставки, собрав как можно более высокую покерную комбинацию, используя 5 карт, или вынудив всех соперников прекратить участвовать в игре. Игра идёт с полностью или частично закрытыми картами. Конкретные правила могут варьироваться в зависимости от разновидности покера. Обобщающими элементами всех разновидностей покера являются комбинации и наличие торговли в процессе игры.

Самой популярной игрой является безлимитный Техасский Холдем. В рамках данной работы обсуждается только этот вид покера. Ввиду того, что игрок не знает карты своих противников, покер является игрой с неполной информацией, как и многие другие карточные игры, в отличие от, например, шахмат, в которых оба игрока видят положение всех фигур на доске и могут точно сказать какой ход верный, а какой нет.

Покер не является рулеткой или блэк-джеком, где все вероятности известны заранее и от уровня игры ничего не зависит – удача и только. В покере игра ведется между людьми.

Выигрывает тот, кто играет лучше. Можно сказать, что покер – это игра ошибок. Чем больше допущено ошибок, тем меньше выигрыш или даже больше проигрыш.

Человеку свойственно ошибаться, входить в азарт, не контролировать эмоции. Лучший игрок в покер – это тот игрок, который постоянно контролирует ситуацию, анализирует происходящее и делает выводы. Возникает идея создать программу – идеального игрока, никогда не ошибающегося, не знающего, что такое эмоции и азарт, принимающего только верные решения, а значит постоянно выигрывающего.

Люди со всего мира играют в покер в интернете. Для этого устанавливается программа–клиент (далее - покер клиент) на компьютере игрока, предоставляющая интерфейс для игры в покер. Главной целью данной работы является разработка программы, автоматически играющей в покер в интернете через покерный клиент, т.е. имитирующей действия человека сидящего за компьютером. Такую программу назовем покерным ботом.

Необходимо исследовать математический аппарат покера, существующие стратегии игры в покер, выбрать подходящую для алгоритмизации стратегию, доработать и запрограммировать ее.

1 Правила покера

Правила техасского холдема:

Структур стола, Блайнды

В холдеме баттоном (от англ. «button» - «кнопка») отмечается номинальный дилер раздачи. Перед началом раздачи игрок, следующий по часовой стрелке после баттона, ставит малый блайнд – первую обязательную ставку. Следующий игрок ставит большой блайнд, который обычно в 2 раза больше малого блайнда.

После этого каждый игрок получает по две закрытых карты. Игроки делают свои ходы в порядке очереди по часовой стрелке, начиная с игрока в позиции «под прицелом» (англ. under the gun). Это позиция игрока, сидящего по часовой стрелке от большого блайнда.

Действия игрока. В холдеме, как и в большинстве других разновидностей покера, игрок может сделать «фолд» - сбросить карты, «бет» сделать ставку, «колл» - принять ставку или «рейз» - повысить ставку. Доступность тех или иных действий зависит от действий предыдущих игроков. У любого игрока всегда есть возможность сделать фолд, то есть сбросить свои карты и отказаться от борьбы за банк. Если до Вас никто не сделал бет (ставку), то Вы можете сделать либо чек (отказаться от ставки, но не сбросить карты), либо ставку. Если один из игроков сделал ставку, то последующие игроки могут сделать фолд, колл или рейз. «Сделать колл» означает добавить в банк количество фишек, необходимое для уравнивания ставки предыдущего игрока. «Сделать рейз» означает добавить в банк количество фишек, превышающее ставку предыдущего игрока.

Игра в холдем состоит из 4 кругов торговли.



Рисунок 1 - Скриншот покерного стола FullTilt Префлоп

Префлоп

После того, как игроки получили свои закрытые карты, каждый из них может продолжить играть, сделав колл или рейз большого блайнда. Первым ходит игрок слева от большого блайнда. На рисунке 1 представлена игра на префлопе на онлайн сайте FullTilt). Каждый раунд торгов продолжается пока все активные игроки (которые не сделали фолд) не сделают равные ставки в банк.

Флоп. Теперь на борд сдаются три карты лицом вверх. Они называются «флопом». В холдеме три карты флопа являются общими картами, доступными всем игрокам, продолжающим игру в этой раздаче. Торги на флопе начинаются с первого активного игрока по часовой стрелке от баттона. Варианты действий практически такие же, как и на префлопе. (см рисунок 2)



Рисунок 2 - Скриншот покерного стола FullTilt Флоп

Однако, если ни один из предыдущих игроков не сделал бет, игрок может сделать чек, передав ход следующему игроку по часовой стрелке.

Торн. После завершения торгов на флопе лицом вверх сдается «терн». Торн – это четвертая общая карта в холдеме. Происходит еще один раунд торгов, начиная с активного игрока, находящегося по часовой стрелке от баттона. (см рисунок 3)



Рисунок 3 - Скриншот покерного стола FullTilt Торн

Ривер. После завершения торгов на терне лицом вверх сдается «ривер». Ривер – это пятая общая карта в холдеме. Раунд торгов снова начинается с активного игрока, находящего по часовой стрелке от баттона, при этом применяются такие же правила торгов, что на флопе и терне. (см рисунок 4)



Рисунок 4 - Скриншот покерного стола FullTilt Ривер

Вскрытие карт

Если после завершения последнего раунда торгов в раздаче осталось несколько активных игроков, то первым карты открывает игрок, который последним сделал бет или рейз. Если же в последнем раунде не было ставок, то первым открывает карты активный игрок, сидящий по часовой стрелке от баттона. Банк забирает игрок с лучшей комбинацией из пяти карт. Если несколько игроков собрали равносильные комбинации, то банк делится между ними поровну. В холдеме ни одна из мастей не имеет преимущества над другими.

После передачи банка в руки игрока, победившего в раздаче, начинается следующая раздача. Баттон перемещается к следующему по часовой стрелке игроку, ставятся блайнды и сдаются новые закрытые карты.

Отметим, что есть кеш игры, SNG (sit and go) турниры (6 – 10 человек), и турниры с большим количеством участников (от 45 и до нескольких тысяч). Они отличаются структурой. В кеш игре блайнды остаются неизменными на протяжении всей игры. В турнирах блайнды увеличиваются через определенное время, например, каждые 6 минут. Из-за этого факта стратегия игры в кеш будет сильно отличаться от стратегии игры в SNG. В рамках диплома рассматривается кеш игра.

2 Обзор существующих покерных ботов.

Существующие разработки можно разделить на два типа.

1. Исследовательские. Их целью является ответ на вопрос: может ли компьютерная программа обыграть человека? Основной упор делается на разработку искусственного интеллекта покерного бота. Здесь работа ведется уже много лет группами из разных университетов мира.
2. Прикладные. Под покерным ботом понимается комплекс программного обеспечения, которое самостоятельно способно играть на существующих покерных сайтах. Задача считать данные из покер клиента, проанализировать их, принять решение о необходимом действии и «нажать» нужные кнопки.

Исследовательские.

В начале обзора уместно сказать, что все боты такого типа играют исключительно heads-up холдем, т.е. один на один.

Polaris

Самая лучшая в своем роде программа, играет только лимитный холдем один на один (который в силу ограниченных ставок значительно больше зависит от математики и подходит для алгоритмизации, чем безлимитный холдем). Разработчики – группа из университета Альберты (University of Alberta Computer Poker Research Group). Их цель – создание компьютерной программы, играющей лучше, чем человек.

Группа основана в 1995 году и по сей день активно работает. В 2007 г. данная программа играла дубликатный матч против двух известных покерных профи: Фила Лаака (Phil "The Unabomber" Laak) и Али Эслами (Ali Eslami). Матч состоял из четырех сессий по 500 рук в каждой. Причем дубликатная система предусматривала раздачу одинаковых карт как компьютеру против человека, так и тех же карт человеку против компьютера, поэтому влияние случайности в игре было минимальным. В первой сессии Polaris выиграл, однако совсем немного, и это было засчитано как статистическая ничья. Во второй сессии Polaris нанес людям ощутимое поражение. После анализа сыгранных в этих сессиях за первый день рук, игроки нашли некоторые особенности игры Polaris, и в оставшихся двух сессиях победили его, хотя результат и нельзя назвать разгромным. По их мнению программа играла очень сильно.

Основная модель действий

- Накапливание информации для тренировки (история сыгранных рук)
- Создание модели поведения противника
- Поиск наилучших действий против поведения противника.

Бот использует огромный математический аппарат, в частности многое основывается на Равновесии Неша.

Равновесие Нэша ([англ. Nash equilibrium](#)) названо в честь [Джона Форбса Нэша](#) — в [теории игр](#) называется тип решений игры двух и более игроков, в котором ни один участник не может увеличить выигрыш, изменив своё решение в одностороннем порядке, когда другие участники не меняют решения. Такая совокупность стратегий выбранных участниками и их выигрыши называются равновесием Нэша.

Monash BPP.

The Monash Bayesian Poker project – основан Кевином Корбом в 1993 году. С тех пор его разработкой занимались в основном студенты. Умеет играть в heads up limit Holdem на уровне начинающего игрока. Модуль принятия решений основан на байесовской нейронной сети. Накапливает статистику игры противника (сыгранные раздачи), моделирует его поведение.

Прикладные.

OpenHoldem (далее ОН) – набор инструментов с открытым исходным кодом для построения покерного бота. Это фреймворк для считывания данных о текущей игре и программируемый логический движок для Техасского Холдема. Он может работать с любым покер румом, но требует тонкой настройки в каждом конкретном случае.

Для получения игровой ситуации (карты, стеки и т.д.) из покерного клиента ОН использует распознавание образов. OpenHoldem предоставляет язык скриптов для написания стратегии игры, также можно писать на C++.

OpenHoldem – фреймворк. Это значит, что нужно предоставить ему несколько вещей, для правильной работы. Первое – «научить» как в конкретном казино отображаются карты, стеки, ставки и т.д. для получения информации о текущей игровой ситуации. Второе – описать логику игры, что делать и в какой ситуации.

Таким образом OpenHoldem дает большой набор инструментов для построения собственного покерного бота, но построить по-настоящему гибкую логику принятия решения на нем не получится, т.к. описать логику, хотя бы немного приближенную к логике человеческого мозга, скриптами невозможно. Из-за этого было принято решение разработать для игры комплекс программного обеспечения с «нуля».

3 Покерный бот

3.1 Что должен уметь покерный бот

Бот – должен считывать информацию из покер клиента, анализировать ее, принимать решение, и выполнять его. Логично разделить программу на три части:

Ввод данных в этой части происходит построение модели покерного стола. Вводным устройством является само программное обеспечение покер-клиента, в том числе все окна, лог файлы и история партий, а также внутренний режим, поддерживаемый исполняемым файлом. Цель этапа ввода - построение точной модели стола в каждый момент времени - ваши закрытые карты, имена и размеры стеков ваших оппонентов, текущие ставки, и так далее.

Анализ – протекает независимо от двух других этапов. Цель: принять модель стола, составленную на этапе ввода и определить необходимое действие: сбросить карты, сделать ставку, поднять ее или принять.

Вывод данных - После того, как на этапе обработки принято решение, вступает в действие этап вывода. Это выражается в нажатии правильных кнопок на экране, или совершении необходимых действий в игре.

3.2 Ввод данных

Цель этапа – точная модель покерного стола в каждый момент времени.

Вводное устройство – покерный клиент.

Результат этапа – модель стола в виде структуры данных, понятной программе.

Существует множество способов считать данные из покерного клиента: прямое чтение нужных параметров из памяти, инъекция кода в адресное пространство клиента, разбор чата дилера, распознавание образов.

Мы будем использовать инъекцию кода в адресное пространство клиента, а именно DLL инъекцию. Это значительно упрощает ввод данных и имитацию пользовательского ввода.

DLL – динамически подключаемые библиотеки – краеугольный камень операционной системы Windows. В DLL содержатся все функции Windows API. Три самые важные DLL: Kernel32.dll (управление памятью, процессами и потоками), User32.dll (поддержка пользовательского интерфейса, в том числе функции, связанные с созданием окон и передачей сообщений) и GDI32.dll (графика и вывод текста). Чтобы приложение (или другая DLL) могло вызывать функции, содержащиеся в DLL, образ ее файла нужно сначала спроецировать на адресное пространство вызывающего процесса.

Как только DLL спроецирована на адресное пространство вызывающего процесса, ее функции доступны всем потокам этого процесса. Фактически библиотеки при этом теряют почти всю индивидуальность: для потоков код и данные DLL — просто дополнительные

код и данные, оказавшиеся в адресном пространстве процесса. Когда поток вызывает из DLL какую-то функцию, та считывает свои параметры из стека потока и размещает в этом стеке собственные локальные переменные. Кроме того, любые созданные кодом DLL объекты принадлежат вызывающему потоку или процессу — DLL ничем не владеет.

В среде Windows каждый процесс получает свое адресное пространство. Указатели, используемые для ссылки на определенные участки памяти — это адреса в адресном пространстве вашего процесса, и в нем *нельзя* создать указатель, ссылающийся на память, принадлежащую другому процессу. Чтобы получить доступ к адресному пространству другого процесса нужно внедрить в него DLL.

Существует множество способов: внедрение DLL с помощью реестра, с использованием ловушек, с помощью удаленных потоков, внедрение троянской DLL, внедрение DLL как отладчика. Мы будем использовать ловушки для внедрения DLL в адресное пространство покер клиента. Windows Hook (ловушки) - это некоторая точка в механизме обработки сообщений, где приложение может подключить подпрограмму для наблюдения за потоком сообщений в системе и обработки сообщений некоторых типов до того, как они попадут в целевую оконную процедуру. Мы будем использовать CBT Hook, соответствующим Windows API является SetWindowsHookEx [6]. Бот будет вызывать ее во время запуска для того, чтобы вставить эту DLL (DLL, которая содержит эту функцию) в адресное пространство каждого процесса, в том числе в каждый процесс покер-клиента, на компьютере. Во внедряемой DLL может быть любой функционал, в том числе все, что необходимо для управления этапом ввода данных.

Проблема глобальных CBT хуков в том, что они служат причиной того, что хук DLL загружается в адресное пространство каждого процесса на компьютере. Если хук DLL очень большой (например, если он содержит коды для выполнения сканирования экрана и т.д.), это может повлиять на производительность системы, так как DLL будет также включаться в процессы, которые нас не интересуют, например, такие как Notepad.exe. Чтобы решить эту проблему используем два этапа процесса вставки. Первый этап — это хук DLL, которая загружается в каждый процесс таким легким способом, насколько это возможно. Затем, когда хук DLL обнаруживает, что он был загружен в процесс покер клиента, он загружает еще одну DLL, которая содержит основную часть кода для процессов ввода и вывода бота. На рисунке 5 показана схема этого процесса. [5]

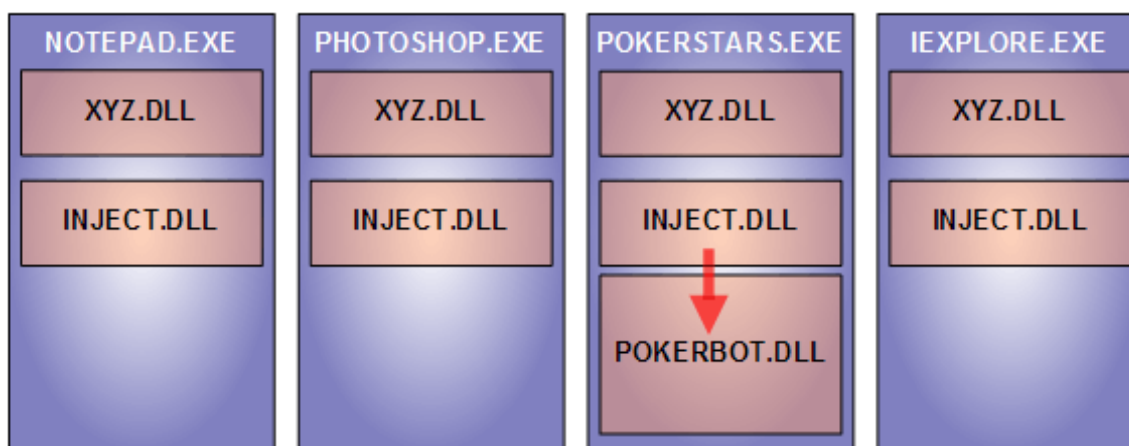


Рисунок 5 - INJECT.DLL - облегченная DLL, в которой содержится процедура CBT хука и код установки. POKERBOT.DLL - это тяжелая DLL, которая содержит логику сканирования экрана. [5]

Как только мы оказались в адресном пространстве покер клиента нам доступны все его функции. В каждом покер клиенте есть чат дилера, в котором отображается вся информация об игре. Если такое окно – стандартный контрол Windows можно получить HWND к окну, а затем и текст через GetWindowText API. Но проблема в том, что многие известные покер сайты, в том числе и FullTiltPoker используют свои собственные контролы, которые в ответ на WM_GETTEXT возвращают пустую строку. Эту проблему можно решить с помощью использования API хука. [6] Подключиться к основному Windows API, который используется всеми без исключения контролами для отображения текста. Это, например, DrawText, TextOut. При подключении такого хука всякий раз, когда покер клиент думает, что вызывает DrawText на самом деле он вызывает измененную версию DrawText, которая извлекает всю необходимую информацию, после этого вызывает исходный DrawText. Это простой способ изучить выходные координаты для определения, например, того что этот текст пишется в чате диллера; а этот текст пишется в строке заголовка и т.д.

Получая эту информацию, создаем точную модель стола в каждый момент времени.

3.3 Анализ данных

Вход – модель покерного стола.

Выход – действие: сбросить карты, сделать ставку, принять ставку.

В этой главе описывается логика принятия решений и ее теоретическое обоснование.

3.3.1 Математика покера

Покерный бот использует математику [1] для принятия решений, а именно математическое ожидание.

Математическое ожидание — мера среднего значения случайной величины в теории вероятностей. Если мат ожидание при игре в орлянку, очевидно равно нулю, то в покере оно зависит от множества факторов. Представим ситуацию:

На ривере, у вас на руках натсовый флеш (см. Приложение сила рук). Против вас два игрока. Первый делает ставку, ваше слово. Если вы уверены, что у первого сет, а у второго, например, две пары, а также что второй готов уравнивать ставку первого, но не ваш рейз, то, очевидно, что для увеличения мат ожидания нужно принять ставку первого, не повышая ее. Если же вы в чем то не уверены: может быть у первого уже фулл хаус и вы биты, может быть, второй не примет вашу ставку ни при каких обстоятельствах, тогда вычисление мат ожидания сильно затрудняется. Вы определяете, с какой вероятностью у первого игрока на руках фулл хаус, с какой вероятностью второй игрок примет вашу ставку, из этого делаете вывод, при каком вашем действии будет самое большое мат ожидание. Это требует определённых умений по чтению рук и мыслей ваших противников.

Шансы банка – это отношение величины ставки, необходимой, либо для вскрытия карт противника, либо для получения следующей карты (терна или ривера) к размеру банка.

Математическое ожидание и шансы банка.

Пример: У нас на руках Ah2h (Туз черви и двойка черви). Борд: Th 6h Kc 3c(десять черви, шесть черви, король крести, 3 крести). А у противника вы предполагаете АК, т.е. топ пара и топ кикер (см Приложение глоссарий). В банке 100\$. Ваш оппонент ставит 60\$. Итого в банке 160\$.

На данный момент мы биты. Наша рука может усилиться до флеша и тогда мы выиграем. Посчитаем вероятность собрать флеш. В колоде 52 карты. Две у нас на руках плюс четыре на борде, осталось 46 карт не открытых. Нам нужна любая карта червей. Всего червей в колоде 13 минус две на руках и минус две на борде, остается 9. Из 46 карт нам помогают усилиться 9. $9 \text{ к } 37 = 1 \text{ к } 4.11$ на выигрыш. В банке 160\$. Нужно доставить 60\$. Шансы банка = $60 \text{ к } 160 = 1 \text{ к } 26.6$. Получаем, что шансы банка намного меньше, чем шансы на выигрыш, а значит, на дистанции мы будем сильно проигрывать. Правильное решение – сбросить. Если бы противник ставил меньше 20\$, было бы выгодно его принимать. [1]

3.3.2 Стратегия коротких стеков

Покерный бот использует стратегию коротких стеков [2]. Основной смысл в том, что все деньги отправляются в банк либо на префлопе, либо на флопе, в самых крайних случаях на терне. Минимизируется количество сложных решений, следовательно, уменьшается количество совершенных ошибок. Используются только самые сильные руки в покере, поэтому вероятность на выигрыш в большинстве случаев будет большей, чем у противника, а значит, на дистанции мы будем в плюсе.

Основная часть решений за покерным столом при игре по стратегии коротких стеков будет приниматься на префлопе. Игра на этой улице описывается чартом стартовых рук.

Позиция – ваше положение за столом относительно малого и большого блайнда. Имеет большое значение – в разных позициях вы будете разыгрывать разные по силе руки.

Позиции за столом на 10 человек. Если игроков меньше, убирала позиции, начиная с ранней позиции.



Рисунок 6. Позиции за покерным столом [4]

Таблица 1 - Чарт стартовых рук [4].

позиция	Стартовые руки при различных действиях до нас				
	Если не было повышений, делаем рейз	Одно повышение, делаем рейз	2а повышения и больше идем алл ин с	Одно повышение после нашего рейза алл ин	Повышение и n*колл отнесем к 2ому случаю
Ранняя	JJ-AA, AK	JJ-KK, AK – all in AA – 3bet – 25% случаев, 75% - all in;	KK-AA	JJ-AA, AK	JJ-KK, AK – all in AA – 3bet + n*bet – 25% случаев, 75% - all in;

Средняя	99-AA, AK,AQ	JJ-QQ,AK– all in KK – 3bet – 35% случаев, 75% - all in; AA – 3bet – 45% случаев, 55% - all in;	KK-AA	JJ-AA, AK	JJ-QQ,AK– all in KK – 3bet + n*bet– 35% случаев, 75% - all in; AA – 3bet + n*bet – 45% случаев, 55% - all in;
Поздняя + блайнды	77 – AA, AT, AJ, AQ, AK, QK	AK– all in JJ – 3bet – 10% случаев, 90% - all –in. QQ – 3bet – 15% случаев, 85% -all-in. KK – 3bet – 35% случаев, 75% - all in; AA – 3bet – 45% случаев, 55% - all in;	KK-AA	JJ-AA, AK	AK– all in JJ – 3bet – 10% случаев, 90% - all –in. QQ – 3bet – 15% случаев, 85% - all-in. KK – 3bet – 35% случаев, 75% - all in; AA – 3bet – 45% случаев, 55% - all in;
<p>Когда все игроки до баттона или катоба (позиция баттон - 1) сбросили и игрок повышает – стил-рейз.</p> <p>Стил рейз с 88-AA, AJ –AK.</p> <p>В ответ на рестил оппонента идем алл ин с TT – AA, AQ,AK</p>					
Размер рейза – случайно - 3bb, 3.5bb, 4 bb. + bb на каждого лимпера (игрок вошедший в игру без повышения).					

Постфлоп:

На рисунке 7 представлены комбинации, которые могут получиться на флопе.

СРЕДНЯЯ ПАРА

Готовая пара из ваших стартовых карт: есть только одна старшая общая карта.



ТОП-ПАРА

Пара из одной из ваших стартовых карт и самой старшей общей карты.



СТАРШАЯ ПАРА

Готовая пара из ваших стартовых карт: нет более старшей общей карты.



OESD

Четыре следующих непосредственно друг за другом карты, которым сверху или снизу не хватает одной карты до стрейта.



ФЛЕШ-ДРО

Четыре карты одной масти, которым не хватает одной карты до флеша.

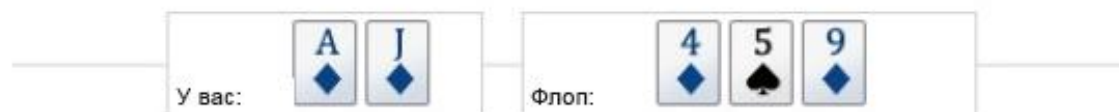


Рисунок 7 - Возможные комбинации [4]

Со средней парой(карманной), топ парой, старшей парой + флеш-дро и OESD и любой картой сильнее (стрит, флеш, сет, фуллхаус, каре, стрит-флеш, роял-флеш). Делаем следующее:

- Если никто еще не делал ставки, то ставим 2/3 банка.
- Если кто-то поставил, сразу идем алл ин.
- Если сделали ставку, ее повысили сразу алл ин
- Если просто колировали ставку, на терне алл ин.

Если ни во что не попали:

- Если банк к началу круга торговли вдвое превышает стек, то идем алл ин.
- Если только один оппонент то делаем продолженную ставку 2/3 стека
- Сбрасываем или чек, если противников > 1 .

3.3.3 Классификация оппонентов

Классификация оппонентов на основе статистических данных. Это очень важная информация, определяющая тип соперника, то, как он реагирует на те или иные действия в разных ситуациях.

Основные параметры статистики [3]

1. Количество вложенных денег в банк (VP\$IP)
2. Процент префлоп рейзов (PFR)
3. Фактор агрессии (AF)
4. Частота вскрытия карт (WSD)
5. Процент рук сброшенных на стилинг (FSBBBTS)
6. Количество попыток украсть блайнды (ATSB)

Первые две характеризуют игру на префлопе, третья на всех улицах.

VP\$IP - voluntary put \$ in the pot (%) - добровольно поставленные в банк деньги - процент рук, когда игрок делал ставку на префлопе. Здесь все call & raise, но нет check на big blind и fold на small blind. Это важный показатель, делящий игроков на лузовых и тайтовых. Обычно у хороших игроков VP\$IP не превышает 25%. По этому показателю сразу видно с чем человек играет. Чем меньше этот показатель, тем с лучшими картами игрок входит в игру.

PFR - preflop raise (%) – префлоп рейз - процент рук, когда игрок делал рейз или ре-рейз до флопа. 2-й по важности параметр, делящий игроков на пассивных и агрессивных. Адекватное значение в районе 7-13%. Рейз от игрока с низким PFR значит что у него что-то сильное, типа старшей пары или АК. Высокий PFR в сочетании с высоким VP\$IP указывает на сверх агрессивного игрока.

AF - aggression factor - фактор агрессии - значение агрессивности игрока, определяется как $(\% \text{ bet} + \% \text{ raise}) / \% \text{ call}$. Фактор агрессии можно считать для каждого раунда игры, но обычно используют усредненный. Это, третий по важности параметр, позволяющий оценить агрессивность игрока, начиная с флопа. Средний фактор агрессии на флопе и

терне 2,8 (диапазон 2-3,5), на ривере чуть ниже - 2,1 (1,6-2,5). Хороший игрок достаточно агрессивен, медленная игра на полном столе особо не рекомендуется, нормальные карты надо защищать, плохие - сбрасывать.

WSD - went to showdown (%) - дошел до вскрытия - процент сдач, в которых игрок смотрел флоп и дошел до вскрытия. Еще одна важная характеристика, позволяет оценить лузовость/тайтовость противника после префлопа. Средняя величина - 32%, диапазон 27-38%. По величине WSD лучше всего можно определить "сопротивляемость" блефу. Игрок с высоким WSD (так называемый "колинг стейшн") будет отвечать со своей младшей парой или даже голым тузом до упора, блеф становится бессмысленным. Игрок с низким значением боится проиграть, на опасном флопе они могут сбросить любую карту, кроме натса, против них блеф наиболее эффективен.

FSBBBTS - Fold SB/BB to steal (%) – процент рук, сброшенных на малом/большом блайнде против попытки "украсть блайнды" (такой попыткой считается любой рейз с двух последних позиций, если до этого в игру не вошел ни один игрок). Средняя величина для малого блайнда - 83%, для большого - 63%, диапазоны, соответственно 68%-91% и 21-79%. Желательно знать данную величину для противников, у которых вы собираетесь воровать блайнды, чем выше - тем смелее нужно рейзить.

ATSB - Attempts to steal blinds (%) - процент попыток украсть блайнды к предоставленным возможностям. Предыдущие два параметра показывали на защиту блайндов, этот - на их атаку. Среднее значение 28%, в диапазоне от 19% до 33%.

Мы перечислили основные статистики. Они будут учитываться при определении правильного действия.

Введём понятие диапазона рук – это совокупность рук, которые может разыгрывать игрок. Например, все будут играть с двумя тузами, но 23 все будут скидывать.

Применение статистики для классификации оппонентов и определения их диапазона рук.

Игроки в покер могут играть по одной из многих возможных стратегий. Они могут играть тайтово на префлопе, но очень агрессивно на флопе, лузово на префлопе и агрессивно на флопе, тайтово на префлопе, пассивно на флопе. Классифицируем оппонентов, основываясь на основных статистиках.

1. Камень (VIP = 9, AF= 7) - очень тайтовые и агрессивные. Такой игрок играет по очень тайтовой стратегии, ждет карманную пару на префлопе, возможно АК. После флопа пытается извлечь как можно больше денег из сета или оверпары, иначе делает фолд, не собрав ни то ни другое. Такие игроки открывают торговлю только рейзом. Диапазон рук таких противников описан в приложении Б. У них можно часто красть банки и блайнды.
2. Ниты (VIP = 13, AF=9) – Разыгрывают несколько больший диапазон рук на префлопе (см приложение Б). У них также как и у камней можно воровать блайнды и банки, но с большей осторожностью.
3. Нитовый, Нормальный, Лузовый ТАГ (ТАГ – тайтово агрессивный игрок VIP=17-24, AF=13-20). Это хорошие игроки, игра которых может быть непредсказуемой, но

они все равно совершают большое количество ошибок. Диапазоны их рук смотрите в приложении Б.

4. Нормальный ЛАГ (VPIP = 29, AF=24) – самые сложные противники. Их основное различие от ТАГов – это более частая кража блайндов и банков. Диапазоны смотрите в приложении Б.
5. Турист (VPIP = 22, AF=7). Такие игроки имеют значения основных статистик, которые могут значительно колебаться, делают множество ошибок, могут идти до конца со средней парой, не понимают силу позиции, проводят полу блефы ни на чем не основываясь. Их диапазон стартовых рук может быть очень широк, против них нужно разыгрывать только сильные руки и на дистанции мы будем в плюсе.

Остальные игроки играют в покер для развлечения и приносят большие прибыли профессионалам. На основе этой классификации стратегия коротких стеков может быть доработана для извлечения большей прибыли.

Пример:

Если предоставлена возможность для воровства блайндов (Мы в позиции баттона и все до нас сбросили), то мы смотрим, какие игроки на блайндах, если они ниты, то мы можем смело воровать блайнды практически с любой рукой. Если они ЛАГи, то нужно посмотреть их **FSBBBTS** статистику и сделать вывод. Диапазон рук, с которыми мы будем воровать блайнды резко сокращается.

3.3.4 Стратегия коротких стеков с учетом классификации оппонентов

Для извлечения большей прибыли из игры мы должны корректировать свою игру относительно игры соперников. Окончательная стратегия коротких стеков против различных типов оппонентов представлена в виде чарта стартовых рук.

Таблица 2 – Модифицированная стратегия коротких стеков.

позиция	Стартовые руки при различных действиях до нас				
	Если не было повышений, делаем рейз	Одно повышение, делаем рейз	2а повышения и больше идем алл ин с	Одно повышение после нашего рейза идем алл ин	Повышение и n*колл отнесем к 2ому случаю
Ранняя	JJ-AA, AK	<i>Против нитов и камней:</i> QQ-AA, AK –	KK-AA	<i>Против нитов и камней:</i> QQ-AA, AK	JJ-KK, AK – all in AA – 3bet + n*bet – 25% случаев, 75% -

		all in Против всех остальных: JJ-KK,AK – all in AA – 3bet – 25% случаев, 75% - all in;		–all in <i>Против всех остальных:</i> JJ-AA, AK	all in;
Средняя	99-AA, AK,AQ	<i>Против нитов и камней:</i> QQ-AA,AK – all in <i>Против всех остальных:</i> JJ-QQ,AK – all in KK – 3bet – 35% случаев, 75% - all in; AA – 3bet – 45% случаев, 55% - all in;	KK-AA	<i>Против нитов и камней:</i> QQ-AA,AK –all in <i>Против всех остальных:</i> JJ-AA, AK	JJ-QQ,AK – all in KK – 3bet + n*bet – 35% случаев, 75% - all in; AA – 3bet + n*bet – 45% случаев, 55% - all in;
Поздняя + блайнды	<i>Против Камней и нитов:</i> 55 – AA, AT, AJ, AQ, AK, QK,QJ, KJ Против ТАГов: 99-AA,AJ-AK,QK Против ЛАГов: 99-AA, AQ,AK Против всех	<i>Против нитов и камней:</i> QQ-AA,AK – all in Против всех остальных: AK – all in JJ – 3bet – 10% случаев, 90% - all –in. QQ – 3bet – 15% случаев, 85% - all-in.	KK-AA	<i>Против нитов и камней:</i> QQ-AA,AK –all in Против всех остальных: JJ-AA, AK	AK – all in JJ – 3bet – 10% случаев, 90% - all –in. QQ – 3bet – 15% случаев, 85% - all-in. KK – 3bet – 35% случаев, 75% - all in; AA – 3bet – 45% случаев, 55% - all in;

	остальных: ТТ- АА,АК,АQ	КК – 3bet – 35% случаев, 75% - all in; АА – 3bet – 45% случаев, 55% - all in;			
<p>Стил рейз (то же что и в поздней позиции).</p> <p>Рестил – ситуация, когда игрок повышает в ответ на наш стил рейз.</p> <p>Идем алл ин:</p> <p>Против нитов и камней: КК,АА</p> <p>Против ТАГов: JJ-АА,АК</p> <p>Против ЛАГов: ТТ-АА,АК,АQ,АJ</p> <p>Против всех остальных: JJ-АА,АК</p> <p>Размер рейза – случайно - 3bb, 3.5bb, 4 bb. + bb на каждого лимпера.</p>					

Постфлоп игра остается такой же, за исключением того, что против нитов и камней скидываем на их пере повышение:

Со средней парой(карманной), топ парой, старшей парой + флеш-дро и OESD. Делаем следующее:

- Если никто еще не делал ставки ставим 2/3 банка.
- Если кто то поставил сразу идем алл ин.
- Если сделали ставку, ее повысили сразу алл ин
- Если просто колировали ставку, на терне алл ин.






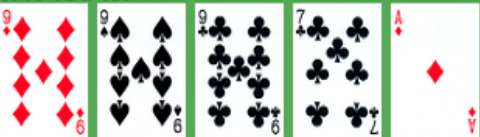

Если ни во что не попали:

- Если банк к началу круга торговли вдвое превышает стек то идем алл ин.
- Если только один оппонент то делаем продолженную ставку 2/3 стека
- Сбрасываем или чек, если противников > 1.

3.3.5 Определение комбинации и шансов улучшить комбинацию

Для определения комбинации мы будем сопоставлять ей число. Первая цифра этого числа будет обозначать номер комбинации – чем больше цифра, тем сильнее комбинация. В случае равенства типа комбинаций, сравниваются карты, из которых эта комбинация получена. В каждой комбинации есть «важность» карты (например в фулхаусе 3-йка важнее чем пара, а в каре четверка важнее кикера). Карты сравниваются в порядке важности: если самые «важные» одинаковы, сравнивают, менее важные. Если очередные сравниваемые карты различны по достоинству, то комбинация со старшей картой считается сильнее. Если различных по достоинству карт нет, то комбинации считаются

равными. Если перевести достоинство карты в числа от 1 до 13, то соответственно 2 и 3 цифры числа комбинации будут соответствовать достоинству самой важной карты, 4 и 5 второй по важности и т.д. Таким образом, мы получим, что чем больше число, тем сильнее комбинация. Например, следующие комбинации будут иметь номера:

<p>СТРИТ ФЛЭШ</p>  <p>КАРЕ</p>  <p>FULL HOUSE</p>  <p>ФЛЭШ</p> 	<p>91000000000 (для стритов важна только 1 карта)</p>
	<p>80907000000 (сначала карта каре, затем кикер)</p>
	<p>71007000000 (сначала тройка, затем пара)</p>
	<p>61009060402 (все карты в порядке убывания)</p>
<p>СТРИТ</p>  <p>ТРИ КАРТЫ</p>  <p>ДВЕ ПАРЫ</p> 	<p>50600000000 (стрит – только 1 важная карта)</p>
	<p>40914070000(сначала тройка, затем страший кикер и младший кикер)</p>
	<p>31002080000(Старшая пара, младшая пара и кикер)</p>
<p>Рисунок 8. Примеры комбинаций и соответствующие им числа.</p>	

Однако такое представление неэффективно использует пространство чисел : из 2-х разрядов(100 значений) использует только 13 значений. Результат можно улучшить используя 16-ричную систему: на 1 карту использовать 1-н 16-ричный разряд, комбинации(уменьшить на 1) и карты(уменьшить на 2) при этом нумеровать с 0. Тогда значения из примера будут иметь вид: 0x880000, 0x775000, 0x685000, 0x587420, 0x440000, 0x37C500, 0x280600, 0x137640, 0x086420. Эти числа уже помещаются в стандартное четырехбайтовое число и их можно спокойно использовать. Если же этого будет не достаточно, то можно будет просто пронумеровать все комбинации, а определять их номера с помощью хеш-таблиц или деревьев и отображений (map), т.е. произвести сжатие.

Определить из 7-ми карт 5-тикарточную комбинацию можно так же, как мы это определяем в жизни глазами:

Прежде всего, определим максимальное количество карт одной масти, если их 5 или больше то мы имеем минимум flush, и тогда следует запомнить первых 5 старших карт этой масти.

Теперь посчитаем, каких номиналов сколько карт у нас есть, и разобьем на соответствующие группы по количеству: 4-ре одинаковых в 1-ю группу, тройки одинаковых во 2-ю, пары в 3-ю, одиночные в последнюю.

Отсортируем их в порядке убывания в группах.

Помимо этого получим последовательность различных карт упорядоченных по убыванию

Если мы имеем в этой последовательности 5 или более карт и разница между 1-й и 5-й равна 4, то мы имеем straight, старшинство определяем 1-й картой.

Тоже самое если в ней 6 карт и разница между 2-ой и 6-ой равна 4, старшинство по 2-й, а также если их там 7 и разница между 3-й и 7-ой(по 3-й).

Если первая карта в последней группе “Ace”, а 4-я с конца “5”, то мы имеем straight с кикером “Ace”.

Это был подготовительный этап. Теперь мы определяем комбинацию

Итак, если мы получили и flush и straight одновременно, то мы имеем 8-ю комбинацию – стрит-флеш и кикер берем из определения стрита.

Если у нас в первой группе есть что-либо, то мы имеем 7-ю комбинацию – каре, первым определителем берем эту самую карту, а вторым – старшую из всех 3-х других групп.

Если у нас во второй группе 2-е карты либо 1-а во второй и хотя-бы одна в 3-й, то мы имеем фулхаус, и определители берем соответствующие карты из 3-й и из 2-й (из второй берем первую – старшую).

Если мы на предварительном этапе получили флеш, то мы имеем собственно флеш, и определителями будем считать карты, которые мы запомнили для флеша в порядке убывания.

Если мы на предварительном этапе получили стрит, то мы имеем стрит и определителем будет определитель стрита.

Если во второй группе у нас есть карта, то мы имеем сет, Определителем будет эта карта, а также 2-е старшие карты из 4-й группы.

Если в 3-й группе есть хотя бы 2-е карты, то мы имеем две пары, и определителями будут эти 2-е карты + старшая из оставшихся в 3-й и 4-й группах.

Если в 3-й группе есть 1 карта, то мы имеем пару, и определителями её будут эта самая карта и 3-и старшие карты из 4-й группы.

Если все карты попали в 4-ю группу, то мы имеем 0-ю комбинацию – Кикер, определителями будут старшие 5 карт из этой группы.

Определение шансов на улучшение комбинации

Если закрытых карт больше нет, то определить комбинацию не составляет труда. Если закрыты 1 или 2 карты, то мы предполагаем, что вероятность выпадения каждой карты, кроме открытых одинакова, и для каждой из них(или для каждой пары) определяем комбинацию. Несмотря на количество различных вариантов карт, комбинаций будет ограниченное количество, и мы можем разбить карты на группы по комбинациям, которые они дадут. Таким образом, мы можем оценить вероятность выпадения соответствующей комбинации. В случае когда у нас известны только 2-е карты на руках ситуация немного другая. Результирующих комбинаций может получиться очень много, поэтому чаще оценивают среднюю вероятность выигрыша, которая заранее просчитана.

4 Программная реализация

Программа реализована на языке C# в среде разработки Visual Studio 2010. Программа является объектно–ориентированной.

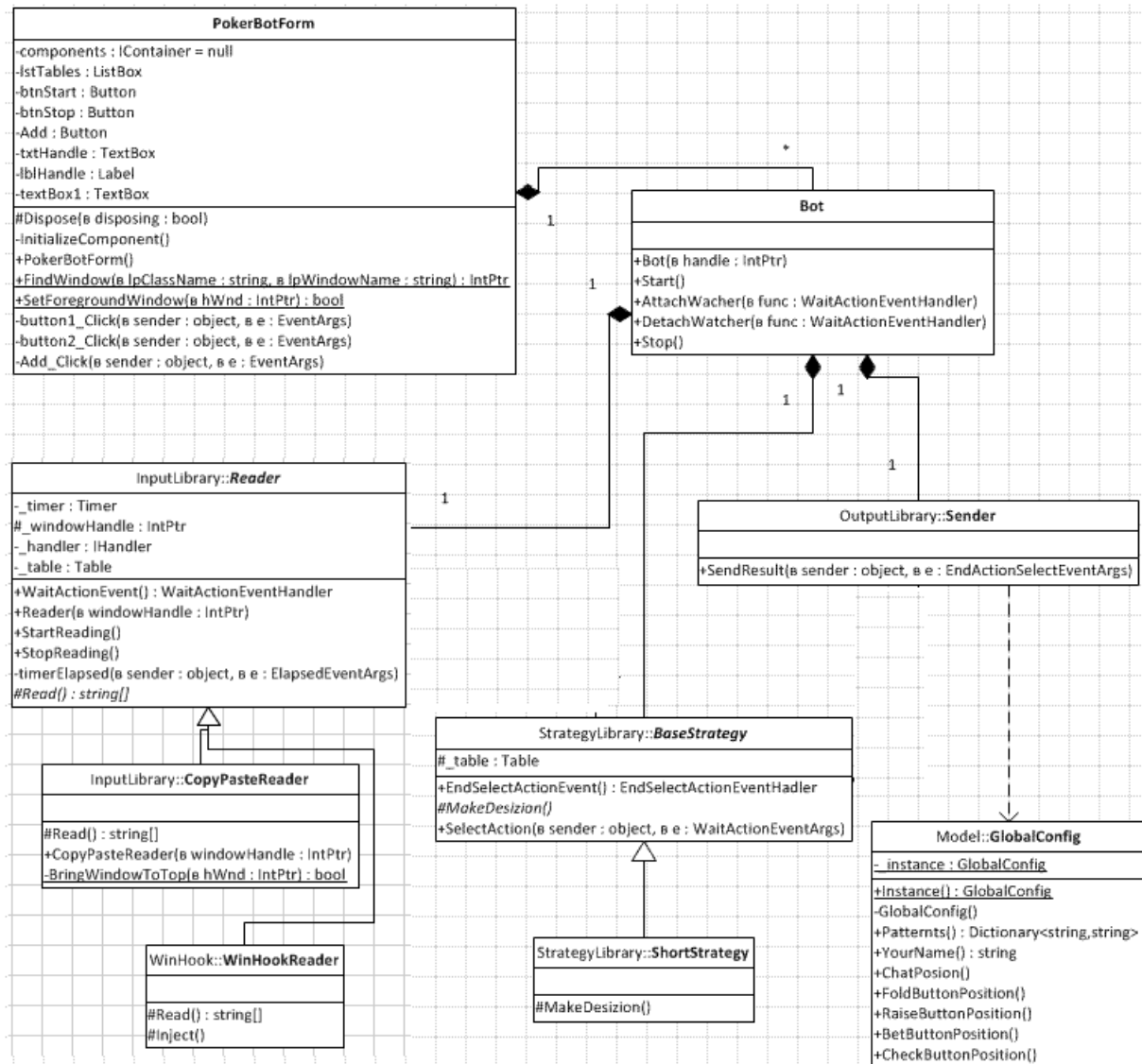


Рисунок 9 - Общая диаграмма классов

Как видно из рисунка № 8. В программе используется один Вид – окно управления ботом (запуск и остановка). Класс Bot – это фасад, предоставляющий интерфейс для работы с ботом. На каждое игровое окно покерного клиента создается экземпляр класса Bot. Класс Bot состоит из трех частей Reader, BaseStrategy, Sender. Reader – отвечает за чтение данных из покер клиента. BaseStrategy – отвечает за принятие решений. Sender – нажимает

нужные кнопки. Классы Reader, BaseStrategy – абстрактные. С абстрактными методами Read(), MakeDecision() – соответственно. Это позволяет использовать различные методы извлечения информации из покер клиента. Также можно использовать несколько стратегий, комбинировать их, и принимать решение на основе результатов каждой из них. Такой подход повышает гибкость программы в целом.

Для снижения связности объектов используются события. Например, В Reader есть событие – наступления хода игрока, на него подписан BaseStrategy. По наступлении этого события BaseStrategy вызывает метод SelectAction и определяется необходимое действие. Как только действие определено появляется новое событие, на которое подписан Sender, он его выполняет. Таким образом, объекты не знают о существовании друг друга, а значит и изменения в одном никак не повлияют на работу другого. При взаимодействии объект Reader передает объект Model::Table, содержащий точную модель покерного стола, в BaseStrategy.

Model::GlobalConfig – отвечает за конфигурацию системы. Он используется на этапе ввода и вывода. Конкретные конфигурации содержатся в отдельных XML файлах, в которых содержится название покер клиента, характерные для него параметры – расположение управляющих кнопок, чата дилера, лог файлов. Это позволяет боту работать с разными покерными клиентами.

Стратегия коротких стеков реализована в BaseStrategy с помощью использования конструкций if else, определяющих ту или иную ситуацию. При этом данные берутся из модели, а именно из Model::Table. Данные анализируются, определяется комбинация, сохраняется в Model::Combination. Определяются шансы на улучшение комбинации. На основе заполненной модели принимается решение о необходимом действии.

На рисунке 10 представлена диаграмма классов Reader.

Для распознавания текста окна чата дилера используется паттерн цепочка обязанностей. Каждый элемент отвечает за обработку конкретного сообщения, содержащего информацию о Raise, Fold, Call и т.д. Такой подход повышает гибкость этапа ввода данных и может быть расширен для многих изменений. Например, мы захотим обрабатывать не только сообщения дилера, но и сообщения пользователей. Эта проблема решится добавлением нового обработчика (Handler). На этом же этапе происходит сбор информации о стилях игры оппонентов.

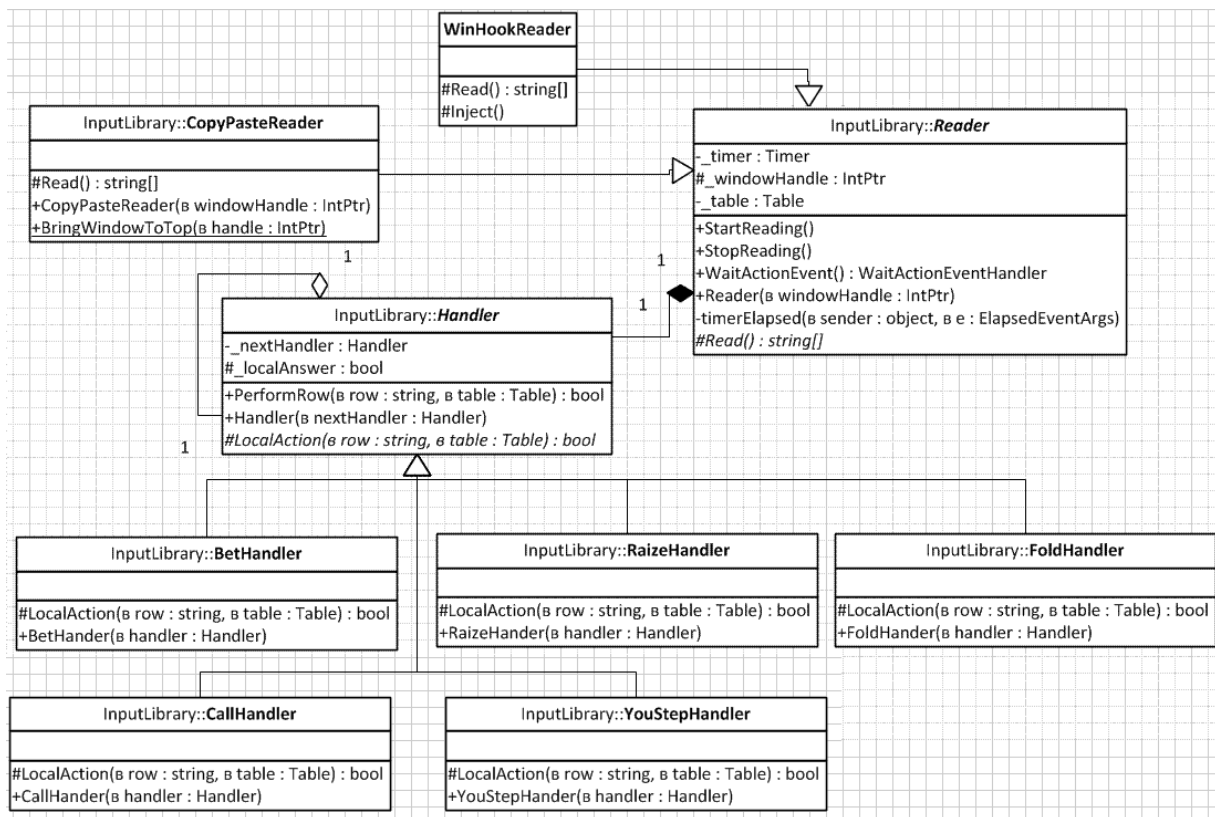


Рисунок 10 - Диаграмма классов Reader

Все данные собранные на этапе ввода хранятся в модели покерного стола. На рисунке 11 изображена диаграмма классов модели.

Модель содержит все данные покерного стола, алгоритмы для определения комбинации и типа противника. Модель используется базовой стратегией для определения верного действия.

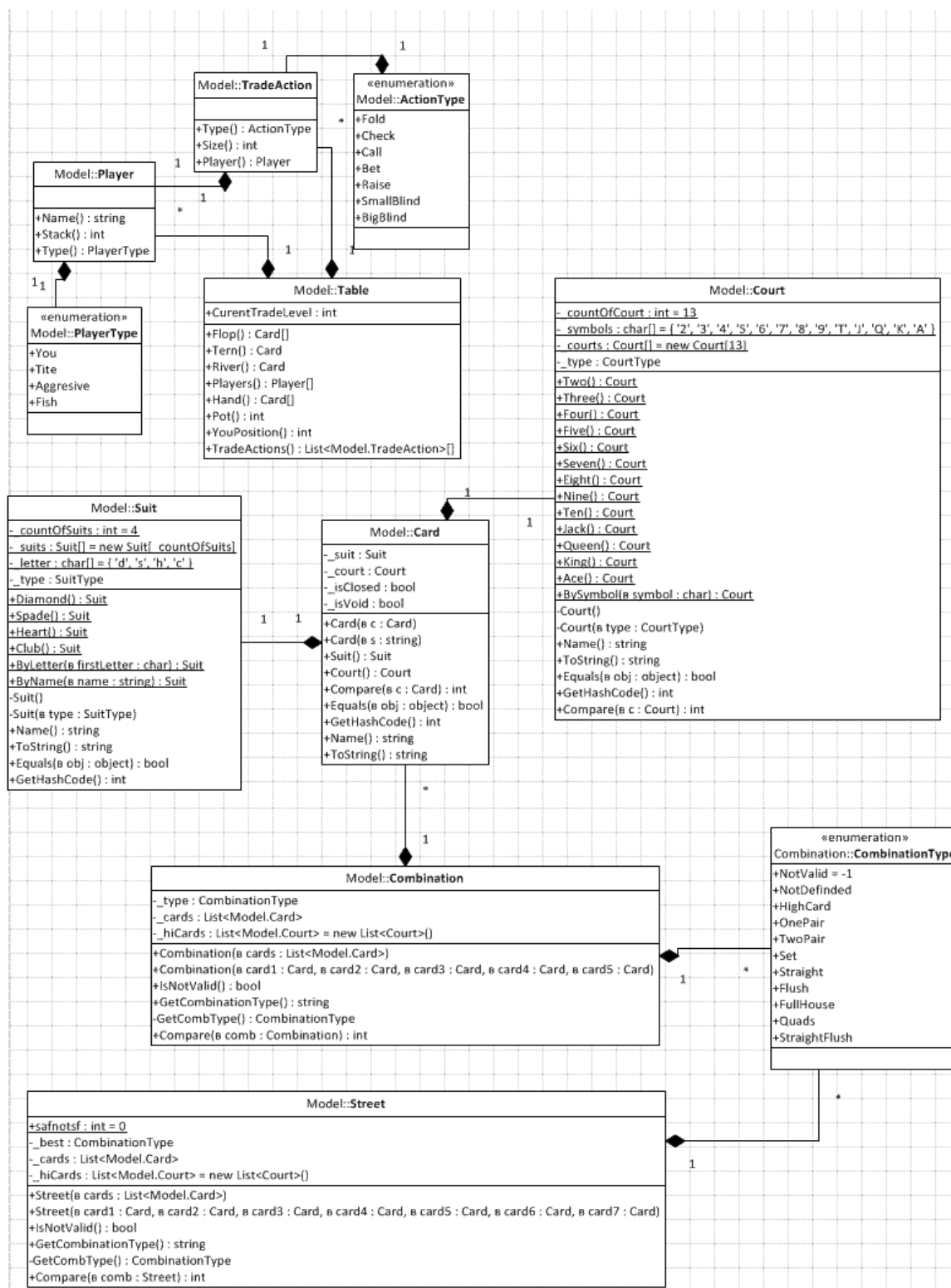


Рисунок 11 - Диаграмма классов модели

Руководство пользователя

Чтобы запустить покерный бот откройте файл PokerBot.exe. На рисунке 13 показано главное окно программы.

Для автоматической игры откройте стол или несколько столов покерного клиента, определите их Handle(например, с помощью утилиты Spy++), введите его в соответствующее поле в интерфейсе бота. Кнопка Add добавляет стол для просмотра и работы.

В ListBox выбирается нужный стол. Кнопками start и stop соответственно запускается и останавливается автоматическая игра.

В окне подробной информации о столе дублируется чат дилера, и выводятся сообщения бота.

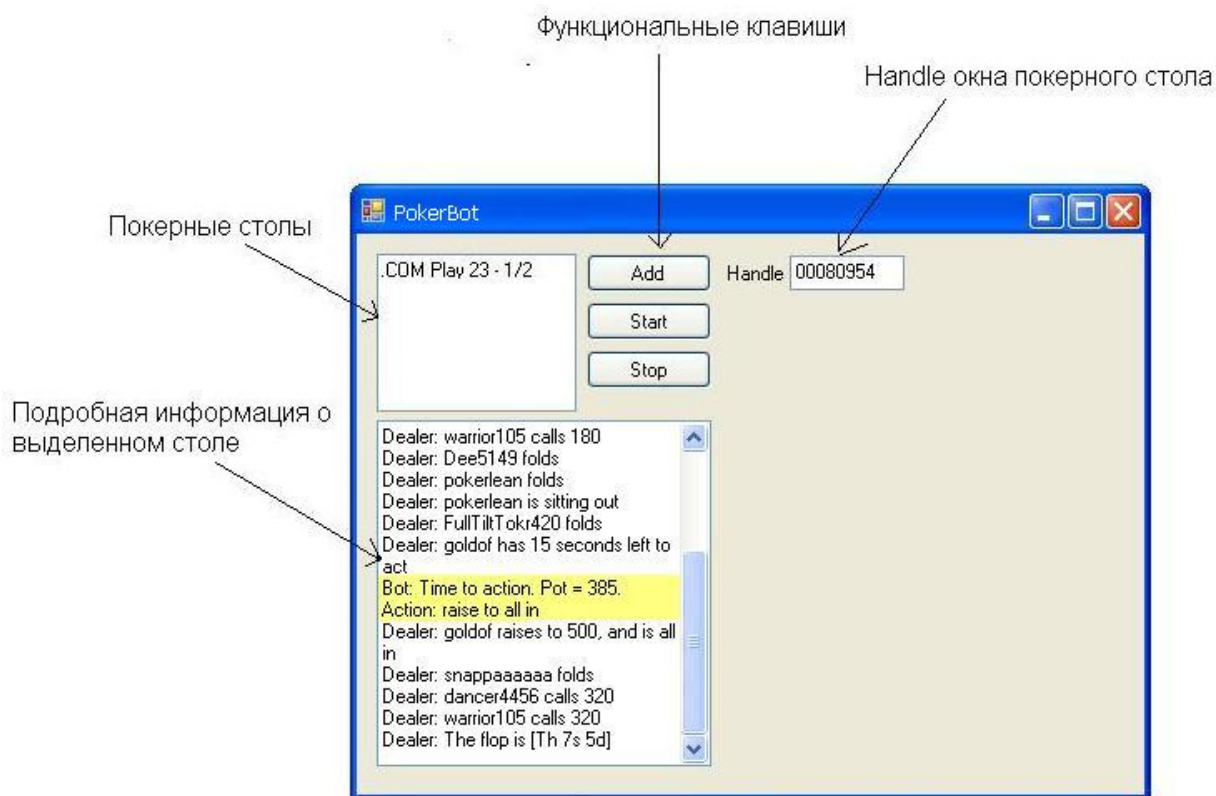


Рисунок 12 - Интерфейс покерного бота

Руководство программиста

Исходный текст программы состоит из совокупности библиотек классов. Язык разработки C#. Для компиляции программы необходимо запустить Visual Studio 2010 и открыть файл проекта `PokerBot.sln`. Скомпилировать проект, после чего можно работать в режиме пользователя.

Рассмотрим основные классы, их свойства и методы:

Model – библиотека классов, представляющая статические данные.

Класс **Court** – обозначает номинал карты.

Перечисление:

CourtType – перечисление типов номиналов.

Статические свойства:

Приватные:

_countOfCourt – общее количество номиналов карт

_symbols – массив символов, которыми представляются номиналы, по 1 символу на номинал.

_courts – массив номиналов карт, на каждый из типов. Для создания используется отложенная инициализация. В совокупности с закрытым конструктором дает шаблон одиночка.

Публичные:

Two, Three, Four ... 13 полей, возвращающих элементы из *_courts*.

Статические методы:

Публичные:

BySymbol – возвращает номинал карты по её символьному представлению.

Свойства:

Приватные:

_type – тип номинала из перечисления *CourtType*.

Публичные:

Name – возвращает строковое представление номинала.

Методы:

Публичные:

Compare – возвращает разницу между номиналами карт

Equals – проверка на равенство

ToString – получение строкового представления

GetHashCode – хешфункция.

Класс **Suit** - Класс для обозначения мастей

Перечисление:

SuitType – перечисление типов мастей.

Статические свойства:

Приватные:

_countOfSuits – общее количество мастей

_letters – массив символов, которыми представляется масти, по 1 символу на масть.

_suits – массив мастей карт, на каждый из типов. Для создания используется отложенная инициализация. В совокупности с закрытым конструктором дает шаблон одиночка.

Публичные:

Diamond, Spade, Heart, Club - поименованные поля, возвращающие элементы из *_suits*.

Статические методы:

Публичные:

ByLetter – возвращает масть по её символьному представлению.

Свойства:

Приватные:

_type – тип масти из перечисления *SuitType*.

Публичные:

Name – возвращает строковое представление масти.

Методы:

Публичные:

Equals – проверка на равенство

ToString – получение строкового представления

GetHashCode – хеш-функция.

Класс **Card** – класс, олицетворяющий карту.

Свойства:

Suit – масть карты

Court – номинал карты

Name – строковое представление карты

Методы:

Compare – разница между картами (по номиналу)

Класс **Player** – класс, олицетворяющий игрока.

Свойства:

Name – имя игрока

Stack – количество фишек у игрока.

Type – тип поведения игрока.

Класс **Table** – класс, олицетворяющий игровой стол

Свойства:

Flop – 3 карты, открытые после 1-го круга торгов

Tern – карта, открытая после 2-го круга торгов

River – карта, открытая после 3-го круга торгов

Players – игроки за столом в порядке их права слова.

CurentPlayer – текущий говорящий игрок

Hand – 2 карты на руках игрока бота

Pot – количество денег в банке

YouPosition – позиция игрока - бота

CurentTradeLevel – текущий уровень торгов

TradeActions – массив списков торговых действий

Класс **TradeAction** – класс, представляющий действие игрока

Свойства:

Type – тип действия

Size – размер ставки если есть

Player – игрок, совершивший действие.

Класс **Combination** – олицетворяет комбинацию карт.

Перечисление:

CombinationType – типы комбинаций.

Свойства:

Приватные:

Type – тип комбинации

_cards – карты в комбинации

_hiCards – старшие номиналы в комбинации

Методы:

Приватные:

GetCombType – определяет тип комбинации

Публичные:

IsValid – проверка комбинации

GetCombinationType – возвращает строковое представление комбинации

Compare – сравнение комбинации

Класс **Street** – класс для выделения комбинации из 7 карт

Перечисление:

CombinationType – типы комбинации.

Свойства:

Приватные:

Type – тип комбинации

_cards – карты в комбинации

_hiCards – старшие номиналы в комбинации

Методы:

Приватные:

GetCombType – определяет тип комбинации

Публичные:

IsValid – проверка комбинации

GetCombinationType – возвращает строковое представление комбинации

Compare – сравнение комбинации

InputLibrary - Библиотека классов, считывающих данные из покерного клиента.

Класс **Reader** – абстрактный класс для чтения данных из клиента.

Свойства:

Приватные:

_timer – таймер периодического вызова считывания.

_handler – начало цепочки обязанностей для обработки считанных строк.

_table – считанный стол.

Защищенные:

_windowHandle – дескриптор окна покерного стола.

Методы:

Приватные:

timerElapsed – метод, срабатывающий по истечению таймера.

Вызывает считывание, и обрабатывает полученные строки цепочкой обязанностей

Защищенные:

Read – абстрактный метод, считывающий данные из клиента в виде массива строк.

Публичные

StartReading – запуск таймера на считывание

StopReading – остановка таймера.

Класс **Handler** – абстрактный класс.

Свойства:

Приватные:

_nextHandler – следующий элемент цепочки обязанностей

Защищенные:

_localAnswer – удалось ли обработать запрос

Методы:

Публичные:

PerformRow – попытка обработать строку, если не получается запрос
отсылается дальше по цепочки обязанностей

LocalAction – обработка строки

Класс **CopyPasteReader** – выполняет чтение массива строк из покер клиента
простым копированием.

Методы:

Read – считывает массив строк из покер клиента

BringWindowToTop – помещает окно клиента поверх всех окон

Класс WinHookReader

Методы:

Inject – вызывает в inject.dll функцию inject, которая загружает ее в
память всех процессов. Inject.dll определяет в какой процесс она
загружена, если в покер клиент, то подгружает библиотеку с кодом
для чтения данных. Последняя считывает данные и записывает в
файл.

Read – считывает строки из файла и возвращает новые.

Все классы ConcreteHandlers (FoldHanler, CallHandler и т.д.) имеют один метод –
переопределенный LocalAction. Он соответствует действию.

Библиотека классов **PokerBot**

Класс **PokerBotForm** - класс окна бота – интерфейс для пользователя.

Класс **Bot** – главный класс – организует работу программы.

Свойства:

Приватные:

_reader – ссылка на объект, считывающий данные из покер клиента

_strategy – ссылка на объект, принимающий решения

Методы:

Публичные:

Start – запускает чтение данных (по сути запускает работу всего бота)

Stop – останавливает чтение данных (прекращает работу бота)

AttachWatcher – выводить считанные данные на форму

DetachWatcher – прекратить выводить считанные данные на форму

Библиотека классов **StrategyLibrary** – классы, отвечающие за принятия решений

Класс **BaseStrategy** – абстрактный класс – принимает модель стола и определяет действие

Свойства защищенные:

_table – модель покерного стола.

Методы:

Защищенные:

MakeDecision – абстрактный метод – определяет необходимое действие

Публичные:

SelectAction – при появлении Event (события) вызывает MakeDecision

Класс **ShortStrategy** – класс, содержащий стратегию коротких стеков.

Публичные методы:

MakeDecision – определяет действие, используя модель стола.

Заключение

В общем виде задачу автоматизации игры в покер в интернете можно разбить на следующие проблемы:

1. Извлечение данных из покерного клиента для получения модели покерного стола
2. Анализ модели и выбор соответствующего действия
3. Выполнение действия

В ходе проектирования были изучены различные подходы к извлечению данных из программ сторонних производителей. Один из подходов детально изучен и реализован.

В ходе изучения второй проблемы, были изучена теория покера и его математический аппарат. Проведен анализ базовых стратегий игры. В ходе анализа было установлено, что стратегия коротких стеков легка для алгоритмизации, при этом не является ущербной и даже имеет преимущества перед остальными стратегиями. Базовая СКС была доработана с учетом реальных динамических условий игры и реализована в покерном боте. Реализовано моделирование поведения противника на основе статистик, которые считаются по истории сыгранных рук.

Реализовано автоматическое нажатие кнопок, соответствующих принятому решению.

Таким образом, реализована программа, автоматизирующая игру в покер в интернете.

Список использованной литературы

1. David Sklansky “The Theory Of Poker” - Two Plus Two Pub 1994 – 276 p.
2. Ed Miller, Sunny Mehta, Matt Flynn “Small Stakes No Limit Hold’em” – Two Plus Two Pub 2009 – 235 p.
3. PokerTracker Statistical Reference Guide [Электронный ресурс] / PokerTracker.com – электронные текстовые данные – Режим доступа: http://www.pokertracker.com/products/PT3/docs/PokerTracker3_Statistical_Reference_Guide.pdf - свободный
4. Стратегия коротких стеков [Электронный ресурс] / PokerStrategy.com – электронные текстовые и графические данные – Режим доступа: http://resources.pokerstrategy.com/Strategy/ru/pdf/ps_nl_basic_handout_V4_ru.pdf - по паролю.
5. How I Built a Working Poker Bot [Электронный ресурс] / CodingTheWheel.com – электронные текстовые и графические данные – Режим доступа: <http://www.codingthewheel.com/archives/how-i-built-a-working-poker-bot> - свободный.
6. Джеффри Рихтер “Программирование на языке Visual C++” / Питер 2009 - С. 616 – 728.

Приложение А. Список терминов

В тексте отчета используются покерные термины, ниже приведены часто используемые.

Action – действие – активные ставки (бет-рейз-ререйз) в игре.

All-in – ва-банк, алл-ин – ставка всех имеющихся денег в банк.

Bet – ставка, делать ставку, бет – действие по добавлению фишек в банк при игре, а также собственно ставка.

Big blind – большой блайнд, – в холдеме два первых игрока перед сдачей делают обязательные ставки «вслепую». Второй игрок, а также сделанная им ставка в размере двух малых ставок и называются большим блайндом.

Blinds – блайнды – два игрока слева от дилера, а также сделанные ими до раздачи обязательные ставки.

Bluff – блеф – ставка, когда ваша рука наверняка слабее руки оппонента.

Board – стол, букв. доска – общие карты, выкладываемые на стол.

Button – кнопка, баттон – небольшая кнопка, указывающая, какой игрок считается дилером. Также название этого игрока, говорят, что он сидит на кнопке/баттоне.

Call – ответ, отвечать, уравнивать, кол – если игрок перед вами сделал ставку, то для продолжения игры вы должны уравнивать его ставку.

Cash game – не турнирная игра, букв. игра на деньги – обычная игра на один стол, когда игрок в любой момент может прийти или уйти.

Check – пас, чек – если до игрока не было сделано ставок, то он может продолжить игру, не делая ставку.

Dealer – дилер – сдающий карты. В домашней игре дилером обычно является каждый игрок по очереди, в казино есть профессиональный дилер, а в онлайн казино дилером является компьютер, тем не менее, игроки по кругу считаются дилерами.

Draw – тянуть, дро – комбинация, на данный момент слабая, но имеющая шансы на улучшение. Чаще всего говорят о флэш-дро (у игрока четыре карты к флэшу) и стрит-дро (четыре карты к стриту).

Early position – ранняя позиция – три игрока слева от большого блайнда.

Flop – флоп – три карты, выкладываемые в открытую на столе, а также следующий за этим раунд торговли.

Flush – флеш – одна из сильных покерных комбинаций, пять карт одной масти.

Flush draw – флеш-дро, букв. тянуть флеш – четыре карты одной масти, дающие шанс собрать флэш.

Fold – сброситься, фолд – сбросить карты в ответ на ставку противника.

Heads up – один на один, букв. с поднятыми головами – игра двух игроков, либо когда остальные сбросились, либо в игре один на один.

Hole cards – карманные карты – в холдеме две карты, раздаваемые игроку в закрытую.

In position – в позиции – быть в позиции, значит находится за противником, действовать вслед за ним.

Kicker – кикер, букв. ударяющий – при сравнении одинаковых пар, двух пар, сетов и каре выигрывает игрок у которого старше не парная карта, эта карта и называется кикером.

Late position – поздняя позиция – баттон и игрок перед ним.

Limp – лимп, букв. хромать – входить в игру на префлопе колом, без рейза.

Loose – лузовый, букв. широкий, свободный – игрок, играющий слишком много рук, а также игра, где играет много лузовых игроков.

Loose aggressive – лузово-агрессивный – характеристика игрока, который играет много рук, причем часто ставит и поднимает ставки.

Loose passive – лузово-пассивный – характеристика игрока, который играет много рук, причем обычно идет до ривера, отвечая на ставки противников.

Middle position – средняя позиция – три игрока между ранней и поздней позицией (шестой-восьмой игроки считая от дилера).

No limit – безлимит – величина ставки, которую игрок может сделать в любом из раундов торговли, ничем не ограничена сверху, игрок в любой момент может пойти ва-банк.

Offsuit – разномастные – две карты разных мастей.

Open – открывать – делать первую ставку в данном раунде торговли.

Overpair – оверпара, букв. пара старше – карманная пара, старше наибольшей карты на столе.

Pair – пара – низкая покерная комбинация, две карты одного ранга.

Pocket pair – карманная пара – пара, полученная игроком на префлопе.

Position – позиция – положение игрока за столом относительно дилера. От дилера по часовой стрелки различают малый и большой блайнды, ранние, средние и поздние позиции.

Pot – банк – куча фишек, составлявших ставки игроков, находится в середине стола.

Quads – каре

Raise – поднимать, рейз – увеличивать ставку, уже сделанную другим игроком.

Reraise – ререйз, букв. повторный рейз – рейз после рейза оппонента, также называется 3-бет.

River – ривер, букв. река – пятая карта на столе, а также следующей за ней последний раунд торговли.

Rock – камень, булыжник – обозначение очень тайтового игрока.

Royal flush – роял флэш – наивысшая комбинация в покере, пять карт от десятки до туза одной масти.

Set – тройка, сет – три карты одного ранга, составленные из карманной пары и карты на столе.

Short stacked – малый стек – игра с очень маленьким стеком, которого не хватит чтобы делать ставки до ривера, игрок вынужден будет раньше поставить ва-банк.

Showdown – вскрытие, показ карт – по окончании торговли на ривере, уравнившие карты игроки показывают карты и определяют победителя. В онлайн покере обязательно показывается только победившая рука, проигравшие руки можно не показывать.

Small blind – малый блайнд, букв. малый слепой – в холдеме два первых игрока перед сдачей делают обязательные ставки «вслепую». Первый игрок, а также сделанная им ставка и называются малым блайндом.

Stack – стек – 1. все фишки, имеющиеся у игрока для игры, 2. стопка фишек.

Steal the blinds – воровать, украсть блайнды – рейз из последней позиции, когда не было ставок до этого, чтобы заставить сброситься блайнды и взять банк еще на префлопе (или повторить ставку и взять банк на флопе).

Straight – стрит – пять последовательных карт.

Straight draw – стрит-дро, букв. тянуть стрит – четыре карты к стриту, дающие шанс собрать стрит.

Straight flush – стрит-флеш – вторая по силе покерная комбинация, пять последовательных карт одной масти.

Street – улица – раунд торговли, торн и ривер иногда называют третьей и четвертой улицей.

Texas hold'em – техасский холдем – самая популярная покерная игра.

Three bet – три ставки, 3-бет – то же что и ререйз.

Tight – тайтовый, букв. тесный – игрок, играющий только самые лучшие карты на префлопе, очень небольшой процент сдач.

Tight aggressive – тайтово-агрессивный – игрок, играющий лучшие карты, причем агрессивно – ставит и рейзит.

Tight passive – тайтово-пассивный – игрок, играющий лучшие карты, но пассивно – редко поднимает, часто отвечает.

Top kicker – старший кикер – самый высокий кикер при вскрытии на равной комбинации.

Trips – тройка, трипс – три карты одного ранга, причем пара на столе соединяется с одной карманной картой игрока.

Turn – торн, букв. поворот – четвертая карта на столе, а также следующей за ней предпоследний раунд торговли.

Two pair – две пары – низкая покерная комбинация, две карты одного ранга, плюс две карты другого ранга.

Under pair – нижняя пара – пара ниже младшей карты стола.

Приложение Б. Диапазоны стартовых рук

На всех рисунком сиреневым показаны те руки, которые разыгрывает оппонент. Красным – однамастные руки, голубым – разномастные, зеленым – карманные пары.

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

Рисунок 12 - Камень (VPIP = 9, AF=7)

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

Ниты 13 - (VPIP = 13, AF=9)

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

Рисунок 14 - Нормальный ТАГ (VPIP=17 AF=13)

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

Рисунок 15 - Лузовый ТАГ (VPIP 24 AF=20)

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

Рисунок 16 - Нормальный ЛАГ (VPIP = 29, AF=24)

Диапазоны стратовых рук помогают скорректировать свою стратегию игры против того или иного типа противника.