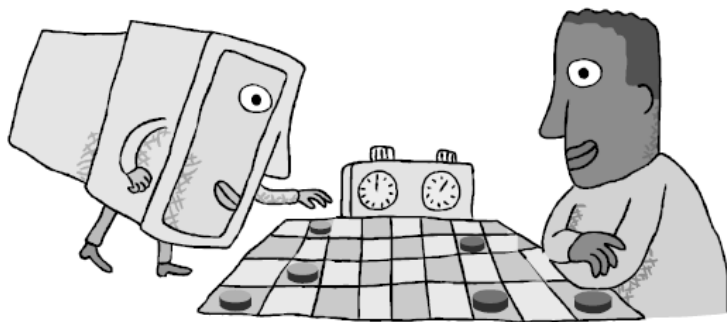


М. М. Глизерин

ПРОГРАММА ИГРАЕТ В ШАШКИ

- ➔ *О чём думает компьютер?*
- ➔ *Что может электронный гроссмейстер?*
- ➔ *Анализируем с программой*



Йошкар-Ола
2007

М. М. Глизерин

ПРОГРАММА ИГРАЕТ В ШАШКИ

Йошкар-Ола
2007

ББК 32.973

Г 54

Глизерин М. М. Программа играет в шашки. — Йошкар-Ола: Педагогическая инициатива, 2007. — 28 с.

Компьютерная вёрстка и дизайн: М. М. Глизерин
Корректор: Н. Д. Глизерина

© Глизерин М. М., 2007

© Издательство «Педагогическая инициатива», 2007

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Что такое игровая шашечная программа?	4
Немного о терминах.	4

КАК И О ЧЁМ ДУМАЕТ ПРОГРАММА

Генератор ходов.	5
Расчёт вариантов.	6
Оценочная функция и оценка позиций.	9
Эндшпильные базы.	10

ЧТО МОЖЕТ ПРОГРАММА?

Что может программа самостоятельно?	13
Что может человек, используя программу?	14
Игра программы против человека.	17

АНАЛИЗИРУЕМ С ПРОГРАММОЙ

Что показывает «Лучший вариант»?	20
Выбираем возможные продолжения.	21
В первую очередь — материал.	22
Анализируя — анализируйте.	23

Приложение

Программы выходят на старт.	25
-------------------------------------	----

ВВЕДЕНИЕ

Что такое игровая шашечная программа?

Игровая шашечная программа — это программа для персонального компьютера (PC), реализующая тот или иной вид шашечной игры (то есть умеющая играть в шашки).

Существуют программы, реализующие различные виды шашек — русские, бразильские, международные (100-клеточные), канадские (144-клеточные), английский чекерс, итальянский чекерс и т. п.

Материал, изложенный в данной книге, отражает, в первую очередь, *возможности и конкретные особенности* современных профессиональных программ, играющих в *русские шашки, бразильские шашки и пул-чекерс*. В то же время очень многое будет справедливо и для программ, реализующих другие виды шашек как на 64-клеточной, так и на других досках (80-, 100-, 144-х клеточных)¹.

Различные компоненты игрового движка, возможности и практическое применение шашечных программ рассматриваются на примере профессиональной игровой программы TORNADO III, играющей в русские шашки, бразильские шашки и пул-чекерс.

Немного о терминах

В главе, рассказывающей об алгоритмах шашечных программ, употребляются такие понятия, как «игровой движок» и «игровая программа». Необходимо пояснить разницу между ними.

Игровой движок — это часть программы, непосредственно отвечающая за реализацию игры в шашки (построение ходов, расчёт вариантов, игра по библиотеке дебютов и эндшпильной базе и т. д.).

Игровая программа, помимо игрового движка, включает в себя пользовательский интерфейс (то есть меню, окна, диалоги,

¹ Общие принципы программирования движков одинаковы для всех шашечных игр. Однако различия в правилах и геометрии досок существенно влияют при прочих равных условиях как на реализацию тех или иных компонентов движка, так и на эффективность их использования. Например, программы, играющие в английский чекерс, рассчитывают значительно глубже, чем программы в русские шашки (это связано с отсутствием размена назад и «тихоходностью» дамки). Еще один наглядный пример: применение в переборе эндшпильных баз на больших досках менее эффективно, чем на малых.

кнопки и т. п.), с помощью которого происходит взаимодействие человека с программой. Кроме того, игровая программа может включать в себя некоторые дополнительные возможности, не относящиеся непосредственно к игровым алгоритмам, но относящиеся к шашкам вообще (например, функции для работы с базами партий, распечатку текста ходов и шашечных диаграмм, построение таблиц и т. п.).

Термин «игровой движок» обычно употребляется в том случае, когда необходимо подчеркнуть, что речь идёт только о той части программы, которая непосредственно играет в шашки. В то же время во многих случаях при упоминании о действиях, присущих игровому движку, используется термин «игровая программа» или просто «программа» (например, *программа рассчитала, программа проанализировала* и т. д.).

КАК И О ЧЁМ ДУМАЕТ ПРОГРАММА

Генератор ходов

Любая шашечная программа начинается с **генератора ходов**, который формализует правила игры. Иными словами, чтобы научить программу ходить по правилам того или иного вида шашек, нужно написать соответствующий генератор ходов.

Основное требование к генератору ходов — это **скорость работы**. Хороший генератор ходов должен работать очень быстро, в противном случае он может очень сильно затормозить скорость расчёта вариантов. При этом, само собой разумеется, генератор ходов не должен содержать ошибок в формализации правил игры.

Создание быстрого генератора ходов для шашечных игр — задача достаточно нетривиальная ввиду того, что в шашках (в отличие, скажем, от шахмат), возможны весьма сложные траектории взятия (особенно «дальнобойными» дамками) в совокупности с правилом, согласно которому сбитые шашки снимаются с доски только после окончания взятия.

Игровой движок TORNADO III оснащен очень быстрым генератором ходов, способным строить позиции (в зависимости от их типа) со скоростью до нескольких миллионов позиций в секунду на процессоре средней мощности.

Расчёт вариантов

Расчёт вариантов на некоторую глубину является **единственным** способом, с помощью которого игровой движок может выбрать лучший ход в той или иной позиции. Расчёт вариантов характеризуется *скоростью* и *глубиной*.

Скорость расчёта вариантов выражается в том, какое количество позиций может построить и оценить (то есть «сосчитать») программа за определённый промежуток времени (на практике обычно обозначается как *поз/сек*, то есть количеством позиций, рассчитанных за секунду).

Глубина расчёта определяет, как далеко «видит» программа. Глубина расчёта выражается в *ходах* или *полуходах*².

Каким образом считает программа? Из некоторой позиции с помощью генератора ходов программа строит *дерево перебора* на определённую глубину. Пример игрового дерева из начальной позиции на глубину 2 полухода показан на рис. 1. Отметим, что в математике деревья принято строить сверху вниз. Самый верхний узел дерева называется *корнем дерева* (в нашем случае это будет позиция, из которой нужно сделать ход). Каждая позиция на нижней глубине (то есть внизу дерева) получает оценку, которая возвращается наверх (об оценке и оценочной функции см. далее). Наверху из этих оценок выбирается та, ветка которой ведет к позиции, наиболее выгодной с точки зрения хода из корня дерева.

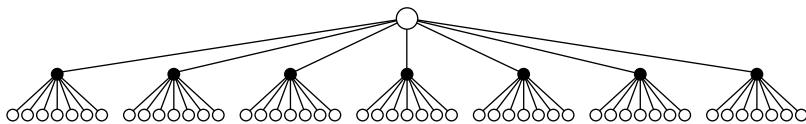


Рис. 1. **Игровое дерево на два полухода из начальной позиции.** В корне дерева белые имеют 7 ходов. После каждого сделанного хода белых чёрные также имеют 7 ходов. В итоге на нижней глубине получаем 49 позиций, оценка которых возвращается наверх.

Всё что относится к построению программой игрового дерева, называется **алгоритмами перебора** (или переборными алгоритмами).

² В шашечном программировании термином «ход» обозначается ход за оба цвета (т. е. ход белых и ход чёрных), а термином «полуход» — ход за один цвет (например, движение только белой или только чёрной шашкой).

Качество переборных алгоритмов является одним из решающих факторов, определяющих силу игры программы. Иными словами, сильный игровой движок должен обладать быстрым и глубоким расчётом. Скорость расчёта современных игровых движков (в русских и бразильских шашках и пул-чекерсе) может достигать миллионов позиций в секунду, а глубина, в зависимости от стадии партии, от 10 до 20 полных ходов (20-40 полуходов).

Создание переборных алгоритмов, способных считать быстро и глубоко, является одной из сложнейших проблем в программировании игр. Дело в том, что с увеличением глубины расчёта количество возникающих позиций растёт экспоненциально и в какой-то момент, просчитав определенную позицию на глубину n , компьютер просто не сможет завершить расчет на глубину $n + 1$ за приемлемый промежуток времени. Поэтому в процессе расчета вариантов возникает необходимость сократить дерево перебора, то есть отбросить как можно большее количество позиций, не влияющих на итоговую оценку.

Проблема сокращения дерева перебора решается с помощью алгоритмов, которые можно разделить на *математические* и *эвристические*.

Математические алгоритмы сокращения дерева перебора известны уже довольно давно и реализованы, в том или ином виде, во всех шашечных программах. В профессиональных шашечных движках используется алгоритм $\alpha\beta$ -отсечений (впервые описанный нашим соотечественником математиком А. Брудно³) и его разновидности (например NegaScout, MTD(f), PVS и другие). Несмотря на то, что базовые принципы реализации этих алгоритмов остаются неизменными, разработчики постоянно стараются улучшить их, найти новые нюансы и возможности для оптимизации («разгона») перебора. Особенность математических алгоритмов сокращения дерева перебора состоит в том, что они поддаются строгому доказательству своей правильности.

Эвристические алгоритмы (или «эвристики») не поддаются строгим математическим доказательствам и основаны на логических умозаключениях, вытекающих из конкретных особенностей игры. Поясним на примере.

³ А. Брудно — выдающийся русский (советский) математик, один из авторов первой отечественной шахматной программы «Каисса».

Предположим, в процессе расчета программа на промежуточной глубине доходит до позиции, в которой у белых преимущество в две шашки. В этом случае, если в позиции нет взятий (то есть, две лишние шашки не будут мгновенно взяты чёрными), программа принимает решение прекратить перебор и вернуть наверх оценку этой позиции. В большинстве случаев эта оценка будет правильной, но может быть и ошибочной. Например, если следующим ходом черные нападут на две шашки (при этом белые на смогут их защитить) или пройдут в дамки и т. п.

Приведенный пример довольно прост, но позволяет понять принцип действия эвристических методов. На практике же применяются достаточно сложные, иногда просто «головоломные» эвристики. Эвристические методы сокращения дерева перебора дают очень хороший результат, позволяя значительно усилить игру программы. Поиск новых идей в этом направлении является одним из важнейших способов усиления игры шашечного движка.

Еще один очень эффективный метод сокращения перебора связан с использованием переборными алгоритмами хэш-таблиц. **Хэш-таблица** — это специальное хранилище данных, находящееся в оперативной памяти компьютера. В хэш-таблицу заносятся позиции, которые программа уже рассчитала и оценила. Позиции, рассмотренные программой ранее, возникают в процессе перебора постоянно. Если такая позиция присутствует в хэш-таблице, программа прекращает перебор, берет из хэш-таблицы оценку и возвращает её наверх.

Для того чтобы использование хэш-таблицы дало ощутимый результат, сама хэш-таблица должна быть достаточно большой. Например, хэш-таблица движка TORNADO III может занимать до 572 Mb оперативной памяти (при размере хэш-таблицы 30 млн. позиций).

Итак, при использовании вышеописанных алгоритмов можно существенно сократить дерево перебора. Но даже в этом случае число позиций в расчёте всё равно останется очень большим. Поэтому очень важно, чтобы алгоритмы перебора обладали высоким быстродействием, то есть выполняли все операции очень быстро. Решение этой задачи относится непосредственно к методам прикладного программирования.

Быстродействие переборных алгоритмов игрового движка TORNADO III достигается за счет тщательной оптимизации ис-

ходного кода, программирования критических участков кода на языке машинных кодов (ассемблера), использования набора команд MMX⁴, учета особенностей аппаратной архитектуры персонального компьютера.

Оценочная функция и оценка позиций

От **оценочной функции** зависит, насколько правильно игровой движок сможет оценить позиции, возникающие в процессе расчёта вариантов (то есть точность расчёта).

Для лучшего понимания того, как работает оценочная функция, в качестве аналогии можно привести игру людей. Так, обдумывая возникающие по ходу игры позиции, мы, как правило, обязаны рассчитывать то или иное количество вариантов. Дойдя в своих расчетах до какой-либо конечной позиции, мы делаем вывод о том, хороша она (на наш взгляд) или плоха (т. е. оцениваем позицию) и, исходя из этого, даем оценку всему рассчитанному варианту.

Аналогичным образом программа, построив определённый вариант и дойдя до конечной позиции, должна оценить её. Для оценки позиции и предназначена **оценочная функция** (ОФ). Понятно, что ОФ является ещё одним из важнейших элементов игрового движка.

Как работает оценочная функция? Во-первых, оценивая позицию, ОФ в обязательном порядке должна **сосчитать материал** (т. е. шашки и дамки) и проверить **наличие ходов** у обеих сторон. Если у одной из сторон нет шашек или ходов, ОФ сразу же возвращает выигрыш или проигрыш (в зависимости от того, какую сторону нужно оценить).

Во-вторых, если говорить упрощенно, в ОФ описаны различные варианты расположения шашек на доске (их также называют **оценочными факторами**: это могут быть, например, сильные и слабые поля (центральные, бортовые, тычки, колы, рожны), сочетания и взаимодействия шашек на сильных и слабых полях, флангах, сильные и слабые группы шашек, всевозможные связки и т. п.), при этом каждое положение шашек имеет соответствующую оценку.

Последовательно сравнивая описанные в ОФ положения шашек с оцениваемой позицией и суммируя соответствующие

⁴ Набор команд MMX — это специальная система команд процессоров Intel и AMD, предназначенная для оптимизации выполнения определенного ряда задач.

оценки (положительные за один цвет и отрицательные за другой), ОФ вычисляет **общую оценку позиции**.

Напрашивается логический вывод: чем больше различных оценочных факторов содержит оценочная функция, тем лучше программа будет понимать позиции и, следовательно, сильнее играть.

Это действительно так, но здесь нужно учитывать следующие моменты. Во-первых, большая ОФ зачастую сильно замедляет перебор вариантов, в итоге падают скорость и глубина расчета. В результате этого программа может начать играть даже хуже. Во-вторых, излишне упрощенная оценочная функция может также очень сильно снизить качество расчёта. В-третьих, очень важен баланс всех оценочных факторов, содержащихся в ОФ.

Поэтому при разработке оценочной функции необходимо обеспечить:

- 1) максимальную скорость выполнения (быстродействия) ОФ;
- 2) оптимальный количественный и качественный подбор параметров ОФ;
- 3) наилучший баланс оценочных факторов.

Проблема быстродействия ОФ движка TORNADO III решается за счет тщательной оптимизации алгоритмов и кода, максимального использования аппаратных ресурсов для увеличения скорости выполнения ОФ. Это позволяет сохранить оптимальную скорость расчета вариантов.

Подбор и баланс параметров ОФ — сложная задача, лежащая как в теоретической, так и практической плоскостях. Например, в процессе подготовки к участию во 2-м Кубке сайта «Шашки в России» (январь, 2007) программой TORNADO III было сыграно более 7000 партий в автоматическом режиме. Партии игрались как между разными версиями игрового движка TORNADO III, так и с другими программами. Основная цель этого «шашечного марафона» как раз и заключалась в подборе и проверки параметров оценочной функции.

Эндшпильные базы

Эндшпильные базы (*базы данных окончаний, ЭБ*) являются неотъемлемой и важнейшей частью современных шашечных программ. Если программа, реализующая шашечную игру на малой доске, не будет иметь эндшпильных баз, она окажется непригодной для выполнения профессиональных задач.

Эндшпильная база — это специальная база данных, которая содержит оценки **всех возможных окончаний** с определенным количеством фигур (т. е. оценки для всех возможных сочетаний простых и дамок на доске с общим количеством в n и менее фигур).

Эндшпильные базы классифицируются:

- 1) по количеству фигур (или по размерности);
- 2) по типу оценок (с абсолютными рангами, с относительными рангами, безранговые).

Классификация по **количеству фигур** определяет наибольшую общую сумму шашек в эндшпильных позициях, содержащихся в ЭБ. Например, 5-и фигурная эндшпильная база (ЭБ 5) включает в себя все позиции 2×3 (3×2), 1×5 (5×1), а также младшие базы с количеством шашек 4 (ЭБ 4), 3 (ЭБ 3), 2 (ЭБ 2). В свою очередь, 6-и фигурная ЭБ содержит все позиции 3×3 , 2×4 (4×2), 1×5 (5×1) и младшие базы на 2-5 шашек и т. д.

Классификация по **типу оценок** указывает на способ разыгрывания результативных эндшпилей.

Базы, ориентированные на оценки с **абсолютными рангами**, содержат для каждой результативной позиции точное количество полуходов до выигрыша или проигрыша при оптимальной игре обеих сторон (уточним, что в данном случае выигрыш или проигрыш подразумевает отсутствие в финальной позиции у одной из сторон ходов или материала). Таки эндшпильные базы называют также *базами окончаний типа «Perfect play»*.

Преимущество таких баз заключается в том, что, используя их, программа, во-первых, найдет самый короткий путь к выигрышу в любой выигранной позиции, а во-вторых, в проигранной позиции выберет самую оптимальную (длинную) защиту. Иными словами, разыгрывание эндшпиля в данном случае будет идеальным.

Базы, ориентированные на оценки с **относительными рангами**, содержат для каждой результативной позиции точное количество полуходов *до перехода в младший эндшпиль*, что есть либо до взятия, либо до превращения простой шашки в дамку (опять же, при оптимальной игре белых и чёрных).

Используя базу с относительными рангами, программа не сможет, как в предыдущем случае, разыграть эндшпиль идеально, так как быстрейший переход в младший эндшпиль отнюдь не подразумевает сильнейшую игру. Тем не менее, используя отно-

сительные ранги, программа сможет довести до победы любой выигранный эндшпиль, но о сильнейшей защите говорить уже не приходится.

Безранговые ЭБ не содержат информации о количестве ходов до выигрыша или проигрыша. В таких базах все позиции, в зависимости от исхода игры в них, имеют всего три оценки: выигрыш (win), ничья (draw), проигрыш (lose). По первым буквам английских слов безранговые ЭБ обозначают также как W/D/L. Во многих случаях, играя по безранговой базе, программа не сможет довести выигранный эндшпильную позицию до победы. ЭБ типа W/D/L используются, в основном, в расчете миттельшпильных позиций (см. ниже).

Эндшпильные базы, будучи частью игрового движка, выполняют две основные задачи. Первая — непосредственно *разыгрывание эндшпильных позиций*. Вторая — это *применение ЭБ для расчёта позиций миттельшпиля*.

Принцип использования эндшпильных баз при разыгрывании окончаний следующий. Чтобы сыграть по базе, программа находит оценку (ранг) позиции, из которой нужно сделать ход. Затем, построив все возможные ходы из исходной позиции, программа находит позицию, ранг которой на единицу меньше. Ход, который приводит в данную позицию и будет лучшим ходом. Разумеется, эндшпильные базы при этом должны быть ранговыми.

Если же программа будет играть дамочный эндшпиль по безранговой ЭБ, может возникнуть такая ситуация, что программа как бы «зависнет». То есть будет «гонять» дамки туда-сюда, не упуская выигрыш, но и не приближая его.

Несколько по иному используются ЭБ в расчете позиций миттельшпиля. Основной принцип таков: когда программа, рассчитывая варианты, доходит по какой-либо ветке до эндшпильной позиции, оценка для неё берется из базы. Понятно, что дальнейший расчет из позиции, для которой известна оценка, не имеет смысла, поэтому программа прерывает перебор и оценка из ЭБ возвращается назад. Использование ЭБ в данном случае существенно сокращает количество рассматриваемых в расчете вариантов и значительно увеличивает точность расчёта, во многих случаях позволяя «пробить» миттельшпильную позицию до результата. Например, при использовании 7-ми фигурной базы полному анализу подаются большинство позиций 5x5, 6x6 и 7x7,

при использовании же 8-ми фигурной базы точность и быстрота расчета до результата возрастают еще больше.

В расчёте вариантов используются, как правило, безранговые базы, так как точная оценка в данном случае не нужна (важен только сам результат). Преимущество же баз типа W/D/L в данном случае заключается в том, что они в намного большей степени, чем ранговые, поддаются сжатию и, соответственно, больше позиций из ЭБ можно загрузить в оперативную память.

Игровой движок TORNADO включает в себя полную 8-ми фигурную базу для русских шашек и полные 7-ми фигурные базы для бразильских шашек и пул-чекерса. Для разыгрывания эндшпилей используются ЭБ типа «Perfect play», для расчета позиций миттельшпиля используются ЭБ типа W/D/L.

ЧТО МОЖЕТ ПРОГРАММА?

Итак, каковы практические возможности игровой программы, обладающей сильным движком (т. е. быстрым генератором ходов, эффективными переборными алгоритмами, мощной оценочной функцией и многофигурной эндшпильной базой)? Ответ на этот вопрос целесообразно рассмотреть с трёх точек зрения:

- 1) Что может программа самостоятельно?
- 2) Что может человек, используя программу?
- 3) Игра программы против человека.

Что может программа самостоятельно?

1. Программа находит комбинации любой длины и сложности. Исключением могут стать комбинации с запиранием простых, дамок и т. п., то есть комбинации, больше присущие композиции, нежели практической игре.

2. В начале и в середине партии программа способна рассчитать любую позицию не менее чем на 10 полных ходов (20 полуходов) на компьютере средней мощности. В целом глубина расчета в дебюте и миттельшпиле может достигать 15 полных ходов.

3. Рассчитывая позицию, программа увидит все тактические нюансы, как явные, так и скрытые.

4. Программа с легкостью рассчитывает форсированные варианты любой длины, в том числе с единственными хорошими ходами.

5. Для программы не является проблемой разобраться в позициях закрытого и полузакрытого типа с большим количеством шашек, причем глубина расчета в таких позициях может быть существенно выше средней.

6. Используя в расчёте многофигурные базы окончаний, программа однозначно оценит большинство позиций с соотношением фигур 7x7, 6x6, 5x5 и т. д.

7. Если программа играет по эндшпильной базе, можно быть уверенным, что она никогда не ошибётся⁵. Вообще, как мы уже сказали выше, эндшпильные базы — это полностью закрытая игра.

Здесь будет уместно спросить: а что же **не может программа?** Есть ли у нее недостатки?

Можно сказать, что недостаток у программы *всего один*, но существенный: **программа не умеет мыслить стратегически**. То есть в позициях, для оценки которых важен не только расчет, но и понимание *далёких последствий* того или иного хода, программа может избрать неправильный план игры. Через определенное количество ходов программа «поймет», что план, выбранный ею, плох, но может оказаться, что защитительных ресурсов уже не будет.

Однако следует заметить, что современные шашечные программы подвержены этому недостатку значительно меньше, чем программы, которые создавались прежде (например, в 90-е годы прошлого или в самом начале нынешнего века). *Это связано со значительно возросшей глубиной расчета*, вследствие чего программа может правильно оценить позиции, с которыми не справилась бы ранее. У первых шашечных программ признаки отсутствия стратегического мышления можно было наблюдать довольно часто; в дальнейшем этот недостаток стал преодолеваться и у современных программ его можно наблюдать значительно реже.

Что может человек, используя программу?

Для человека, профессионально занимающегося шашками, шашечная программа — это, в первую очередь, мощнейший инструмент, открывающий новые возможности для анализа и

⁵ При этом подразумевается, что, как эндшпильная база, так и алгоритмы игры по ЭБ и их практическая реализация не содержат ошибок. Случаи, когда код некоторых программ содержал ошибки, встречались на практике.

изучения шашечной игры. Анализ с использованием шашечных программ имеет две основные отличительные черты — это *скорость* и *точность*.

Анализ любого вида (прямой, обратный и др.) в любой стадии партии выполняется с помощью программы **намного быстрее**. Кроме того, профессиональные шашечные программы обычно располагают специальными инструментами для оптимизации аналитической работы (см. далее).

Анализ, выполненный с помощью компьютерной программы, всегда отличается точностью, во многих случаях недостижимой для человека. Более того, с появлением шашечных программ впервые в истории игры появилась возможность делать анализ, **полностью свободный от ошибок**. Даже наиболее признанным корифеям шашек «докомпьютерной эпохи» не удавалось подобное. Сейчас же с помощью шашечных программ мастерством безошибочного анализа может овладеть любой квалифицированный шашист.

Так как игра в шашки — это, по сути своей, анализ, шашечные программы как инструмент для анализа используются сегодня повсеместно. Перечислим некоторые области их применения.

ПРАКТИЧЕСКИЙ АНАЛИЗ ПАРТИЙ. Программы используются во всём, что включает в себя практический анализ партий. Например, с помощью программ выявляются ошибки и неточности, допущенные игроками, упущенные возможности (достижение выигрыша или ничьей, комбинации, жертвы, атаки и т. п.), рассматриваются возможные варианты и т. д. Благодаря программам значительно выросло число качественно проанализированных партий, опубликованных в печати и сети Интернет.

ДЕБЮТНАЯ ТЕОРИЯ. Появление компьютерных программ дало мощный толчок дальнейшему развитию дебютной теории. Создаются новые системы, окончательно уточняются старые, в том числе классические схемы и варианты. Многие дебютные системы анализируются до результата. Благодаря программам в последние годы быстро шагает вперёд теория современных русских шашек (жеребьёвка ходов и «летающие» шашки).

ИГРА ПО ПЕРЕПИСКЕ. Не секрет, что в последние годы игра по переписке не обходится без шашечных программ. По сути, игра по переписке превратилась в так называемые «продвину-

тые шашки», то есть в вид игры, где человек играет в тесной связке с программой.

Оставив в стороне многочисленные дискуссии по поводу того, нужен или нет такой вид шашек, а также будущее игры по переписке, скажем следующее.

Во-первых, при игре в связке «человек — компьютер» очень много решает уровень подготовки человека как в шашечном плане, так и в плане умения использовать программу. Иными словами, больше шансов победить у шашкиста, который лучше понимает игру (*вспомним, что программы не умеют мыслить стратегически*), и лучше владеет шашечной программой как инструментом анализа. Например, московский шашкист Пётр Васильевич Зажигалин, обладая глубоким пониманием игры и используя игровую программу TORNADO III, два раза подряд с ощутимым преимуществом выиграл заочные чемпионаты Москвы, неоднократно становился победителем и призёром очень сильных по составу турниров по переписке (при том, что и другие участники также пользовались программами).

Во-вторых, партии, сыгранные в «продвинутые шашки», отличаются очень высоким качеством. Не будет преувеличением сказать, что многие из них входят в число лучших партий, иггранных в шашки вообще. В последние годы проводится большое количество заочных турниров (как в классические, так и в современные шашки), в том числе по сети Интернет (*по так называемой электронной переписке*). Партии этих турниров являются ценным вкладом в теорию и практику шашечной игры. Большинство заочных турниров последних лет включаются в *базы шашечных партий*.

ИЗУЧЕНИЕ ЭНДШПИЛЕЙ. Шашечный эндшпиль — самая сложная часть игры для человеческого восприятия и понимания. Наличие у игровых программ эндшпильных баз (то есть полностью закрытой игры в окончаниях) открывает качественно новые возможности для изучения закономерностей и систематизации игры в окончаниях.

Однако, если говорить не об анализе конкретных окончаний (что естественно, не представляет никакой проблемы), а об изучении теории эндшпилей в целом, можно констатировать факт, что в этой области шашечного анализа до сих пор существенных подвижек не произошло. Возможно, это связано с тем, что

задача *обобщения* закономерностей игры в окончаниях лежит как раз на человеке (еще раз напомним — компьютер «мыслит» конкретно).

Частичным решением этой проблемы может явиться создание специальных алгоритмов, выявляющих определенные закономерности разыгрывания окончаний. Но это уже специфическая задача, напрямую не относящаяся к разработке игровых программ.

Кроме того, как ранее, так и в настоящее время, исследование эндшпилей входит в круг интересов очень и очень немногих шашкистов. В последние годы не появилось ни одной значимой работы, посвященной теории шашечных окончаний. Остаётся предположить, что качественный сдвиг в этой области шашек ещё впереди.

КОМПОЗИЦИЯ. Шашечные программы позволяют композиторам создавать произведения, не содержащие ошибок. Так же, как и в случае с практическим анализом и теорией дебютов, уточняются и исправляются композиции, созданные в предыдущие годы без использования шашечных программ. Особенно эффективным в этом смысле является поиск ошибок в этюдах с использованием эндшпильных баз.

Игра программы против человека

Игра против первых компьютерных программ не вызывала особых трудностей. Несмотря на то, что программы хорошо видели тактику, глубина расчёта и позиционное понимание оставляли желать лучшего. Преимущество человека в позиционной игре было очевидным, а из-за отсутствия эндшпильных баз расчёт программ в эндшпиле значительно замедлялся и зачастую программа не могла реализовать или защитит даже простые эндшпили.

С тех пор многое изменилось и сила игры современных программ не ставится под сомнение. Тем не менее возникает вопрос: *какова сила игры программы в сравнении с игрой человека?*

Уже три-четыре года назад стало ясно, что сильнейшие игровые программы сравнялись по уровню с лучшими игроками среди людей. Международный гроссмейстер, чемпион СССР и экс-чемпион мира А. Б. Кандауров сумел победить в матчах в классические шашки сильнейшие на то время программы «Ма-

гистр» и «Тундра», но с минимальным преимуществом. В то же время, участвуя в турнирах среди людей, эти программы неизменно оказывались на призовых местах, а в товарищеских и тренировочных матчах побеждали мастеров и гроссмейстеров.

В течение последних лет игровые программы продолжали развиваться и значительно прибавили (что, например, показал состоявшийся в январе 2007 года 2-й Кубок сайта «Шашки в России»). Поэтому можно сделать вывод, что на данный момент они уже превосходят по силе сильнейших игроков среди людей. Чтобы говорить об этом однозначно, необходимо провести несколько матчей между лучшими игроками и шашечными программами. Но в последнее время подобные матчи не проводятся. Одна из причин этого заключается в том, что возможные участники подобных матчей со стороны людей не без основания предполагают, что программы окажутся сильнее. Например, А. Кандауров, говоря о возможности повторной встречи с «Тундрой», признается, что «сейчас программа «поросла», и я без энтузиазма отвечаю на предложения сыграть с ней матч-реванш».

Значит ли вышесказанное, что при игре с программами у обычных игроков нет шансов и программы всегда будут выигрывать? Как ни странно, вовсе нет. Если, играя с программой, придерживаться некоторых правил, то можно как минимум не проиграть. Каковы эти правила игры с шашечными программами?

1. Нельзя безосновательно ухудшать позицию, создавать сомнительные построения, надеясь запутать программу. В подобных позициях программа обязательно разберется и найдет лучшее для себя продолжение.

2. Лучше всего играть открытые позиции, стараясь при этом занять центр (или, как минимум, не отдавать его).

3. Если вы решили идти на острое или счётное построение (связку, окружение и т. п.), вы должны быть уверены в *значительном преимуществе* за свой цвет, видеть пути его реализации. В ином случае играть такие позиции не имеет смысла. Создавая сложное построение, очень полезно оставлять пути к упрощению (на случай ошибочного выбора плана игры, просчёта и т. п.).

4. Не имеет смысла строить тактические ловушки, причём не только простые, но и самые изощрённые, так как программа без труда увидит их. Построение ловушек и угроз чаще всего

связано с созданием слабостей и ухудшением позиции, при этом вероятность проиграть партию возрастает.

5. При игре в классические шашки очень важно знание теории. Но следует помнить, что если вы играете какой-либо сложный или острый вариант, вы должны быть уверены, что знаете его до конца.

Но если достижение ничьей с программой задача, в общем-то, реальная, возникает вопрос: можно ли выиграть у программы? И если можно, то как?

С этим значительно сложнее. Единственный способ выиграть у программы заключается в том, чтобы построить такую позицию, в которой программа выберет стратегически неправильный план игры. Иными словами, нужно свести партию к такому положению, которое программа не сможет точно рассчитать и оценить. Как уже понятно из всего вышеизложенного, это крайне сложная задача.

Здесь важно учитывать стиль игры конкретной программы, выявить, в каких позициях программа чаще всего проявляет непонимание. Например, программа может неправильно играть против центра в открытой или полуоткрытой позиции, стараясь сразу же разменять его. Если такой размен сильно ухудшит позицию (к примеру, в дебюте), в этом случае можно попытаться сыграть на выигрыш.

Но даже при наличии выигранной позиции против программы достижение выигрыша всё равно может быть достаточно трудной задачей. Достаточно одной ошибки и программа сможет свести партию к ничьей (или даже выиграть). Часто в худшей (проигранной) позиции программа может стремиться перейти в многофигурный дамочный эндшпиль (пусть и проигранный), в котором у человека намного больше шансов ошибиться.

Имеет ли смысл регулярная игра с программой — вопрос отдельный. Практика показывает, что людям частая игра с сильной программой быстро надоедает в виду того, что программы очень редко ошибаются и имеют при этом специфический стиль игры. Однако матчи, проводимые время от времени между сильнейшими игроками и лучшими программами обычно достаточно показательны и интересны.

Кроме того, игровые программы могут быть полезны в тренировочном процессе. Если программа имеет дебютную библиотеку, с её помощью очень хорошо тренировать разыгрывание

начальной стадии партии. Искусственно ограничив силу игры программы, можно тренировать и другие навыки (например игру в различных стадиях партии, реализацию преимущества и т. п.). При этом, если у программы будет ограничена сила игры, выиграть её будет значительно проще.

АНАЛИЗИРУЕМ С ПРОГРАММОЙ

Здесь мы рассмотрим некоторые понятия и приёмы, важные для овладения навыками анализа партий и позиций с использованием игровой программы.

Что показывает «Лучший вариант»?

Выполняя расчёт позиции на определенную глубину, программа может показывать так называемый **лучший вариант**, то есть цепочку ходов из рассчитываемой позиции, которые программа считает лучшими.

Возникает вопрос: действительно ли вся цепочка от начала и до конца состоит из лучших ходов или программа может где-то ошибаться? Чтобы ответить на этот вопрос, необходимо понять следующее: чем глубже программа может рассчитать ту или иную позицию, тем с большей вероятностью она выберет из неё лучший возможный ход.

А что такое цепочка ходов? Это последовательность позиций, в каждой из которых программа выбирает сильнейшее продолжение. Но если для первой (корневой) позиции из этой последовательности глубина расчёта будет максимальной, то с продвижением вниз по цепочке глубина расчёта для каждой последующей позиции будет уменьшаться. Следовательно, будет уменьшаться и вероятность выбора для данной позиции действительно лучшего хода.

То есть, оценивая вариант, предложенный программой, нужно иметь ввиду, что:

- первые ходы варианта могут быть действительно лучшими;
- ходы в середине варианта могут также быть лучшими, но могут и не быть;
- ходы в конце варианта, скорее всего, не будут лучшими.

На практике количество правильных ходов в цепочке «Лучший вариант» зависит, во-первых, от *типа позиции*, а во-вторых от *глубины*, на которую в данный момент рассчитала программа.

Так, если вариант, рассчитанный программой, является форсированным (комбинация, единственные удовлетворительные ходы и т. п.), то лучших ходов в нём может быть большинство, или даже весь вариант может состоять из них. Более того, если программа «пробивает» позицию до результата (выигрыша, ничьей или проигрыша), то можно быть уверенным в правильности всей цепочки лучших ходов.

В остальных случаях точность лучшего варианта может быть большей или меньшей в зависимости от характера позиции. Наименее точной будет цепочка ходов из открытой позиции с большим выбором равных продолжений; в позиции с меньшим выбором хороших ходов точность расчёта будет уже большей и т. п.

В этом случае важна глубина расчёта, так как можно предположить, что чем глубже сосчитает программа, тем большее количество сильных ходов будет содержать лучший вариант.

В программе TORNADO III для отображения лучшего варианта предусмотрено отдельное окно (открывается выбором *F1->Окна->Лучший вариант* или *Alt + V*).

Выбираем возможные продолжения

Еще одна возможность, которую реализуют профессиональные шахматные программы, состоит в том, что программа может рассчитать и оценить все возможные ходы из позиции и показать результат в виде списка ходов с оценками.

В программе TORNADO III эта возможность представлена аналитическим режимом **«Оценить все ходы»** (*F1->Анализ позиции->Оценить все ходы* или *F9*). Примеры списка всех ходов позиции с оценками показаны ниже.

Ход	Оценка	Глуб.

b2-c3	32	(25)
b4-a5	28	(25)
g3-f4	6	(23)
e3-f4	4	(23)
b4-c5	-20	(23)
g3-h4	-112	(23)
e3-d4	-400	(23)

Пример 1

Ход	Оценка	Глуб.

e1-d2	Выигрыш	(29)
b4-c5	Выигрыш	(29)
f2-g3	1	(27)
f4-e5	1	(27)
b4-a5	1	(27)
d4-e5	Проигрыш	(27)
d4-c5	Проигрыш	(27)

Пример 2

Режим «Оценить все ходы» позволяет значительно ускорить процесс анализа.

Так, из первого списка (пример 1) мы видим, что первые два хода из анализируемой позиции ведут к ощутимому преимуществу (оценки +32 и + 28), два следующих хода дают равную игру (+6 и +4), пятый ход значительно ухудшит позицию (–20), а два последних хода связаны с потерей материала (–112 и –400)⁶. Понятно, что при анализе имеет смысл в первую очередь и наиболее глубоко рассматривать ходы с наибольшими оценками, ходы, дающие равную игру, подвергать менее глубокому анализу, а продолжения с худшей игрой и потерей материала вообще не рассматривать.

Второй список ходов (пример 2) показывает, что из семи возможных ходов первые два выигрывают, три следующих хода ведут к ничьей, а два последние проигрывают. Естественно, что в данном случае нас будут интересовать только выигрывающие ходы.

Цифра в скобках указывает, на какую глубину программа рассчитала ход. Увеличение глубины расчета происходит *итеративно*, то есть, пока программа не проанализирует все ходы на глубину n , она не начнет расчет на глубину $n + 1$. Также нужно иметь ввиду, что расчёт в режиме оценки всех ходов происходит медленнее, так как, по сути, программа выполняет на каждой итерации обычный перебор для каждого из ходов.

В первую очередь — материал

Очень часто в процессе анализа бывает необходимо, в первую очередь, выяснить, можно ли в той или иной позиции выиграть материал. Если, например, выигрыш материала возможен, то искать продолжения с позиционным преимуществом уже не имеет смысла.

Для поиска выигрыша материала в программе TORNADO III предусмотрены специальные режимы «Найти лучший ход (только материал)» и «Оценить все ходы (только материал)». В этом режиме программа при расчете игнорирует все позиционные факторы, то есть ОФ в данном случае только считает материал.

Возникает вопрос: для чего нужен этот режим, если и в процессе обычного расчёта программа показывает выигрыш материала? По двум причинам.

⁶ ОФ игрового движка TORNADO III оценивает простую шашку в 100 единиц, дамку в 300 единиц.

1. Если в переборе вариантов оценочная функция (ОФ) не учитывает позиционные факторы, программа сможет рассчитать позицию на большую глубину (см. выше). В качестве примера представим, что рассчитывая некую позицию, программа с полной ОФ сможет выполнить перебор на n полуходов, а с учётом только материала на глубину $n + 2$ полуходов. При этом материал выигрывается на глубине $n + 1$. Как видно, выигрыш материала программа сможет найти только в том случае, если ОФ будет учитывать в переборе только количественное соотношение сил.

2. Предположим, что рассчитав позицию, программа нашла преимущество у белых, оценив его в полшапки (+50, +48, +52 и т. п.). На самом деле такая оценка может означать одно из двух:

- 1) белые получили очень большое позиционное преимущество, но без выигрыша материала;
- 2) белые выиграли материал, но у чёрных есть существенная позиционная компенсация.

Чтобы понять, что же именно обозначает полученная оценка, нужно использовать расчёт с учетом только материала. В первом случае оценка будет равна 0, во втором программа покажет выигрыш шапки (100).

Анализируя — анализируйте

Некоторые шашисты, играющие по переписке, зачастую анализируют следующим образом: ставят программе определенную позицию, запускают на какое-то время обдумывание (расчет вариантов) и ждут, пока программа сделает ход. После чего этот же ход сразу делается в партии.

Допустим ли такой подход к компьютерному анализу? Да, но только в одном случае — когда программа рассчитала позицию до результата (или играет по эндшпильной базе). В остальных случаях подобный способ анализа является в принципе неправильным. По сути, это не анализ, а просто самостоятельная игра программы.

Как мы уже отметили ранее, сильнейшие шашечные программы и самостоятельно способны на многое (в том числе, и в игре по переписке), но всё же компьютерный анализ — это несколько другое.

Суть и преимущества компьютерного анализа заключаются как раз в активном взаимодействии человека и программы. Только в этом случае можно добиться наилучших результатов. Сегодня для серьезной аналитической работы в шашках овладение техникой компьютерного анализа является одной из основных задач.

Ниже приведем несколько советов, касающихся компьютерного анализа в шашках.

1. Выбирая план игры, рассматривайте не только то, что предлагает программа. Ищите собственные продолжения, анализируйте их с помощью программы. Зачастую именно идеи предложенные человеком вкупе с компьютерным анализом дают наилучшие результаты.

2. Проверяйте ходы, предложенные программой. Проверка в данном случае заключается в том, чтобы «продвинуть» вариант на несколько ходов вперед с дополнительным расчётом возникающих на доске положений. Особенно это нужно делать в том случае, если вы чувствуете, что программа не досчитывает. Это отнюдь не всегда занимает много времени, но значительно увеличивает точность анализа и помогает избежать неверных продолжений, если программа выбрала неправильный план игры.

3. Если вы разрабатываете дебютную теорию или «шлифуете» варианты, предназначенные для игры за доской, необходимо обращать внимание на то, каким путём программа достигает выигрыша или ничьей. Дело в том, что в понимании программы нет таких понятий, как «простой» или «сложный». Так, программе всё равно, каким путём достигается выигрыш или ничья. Если, например, ничья достигается несколькими способами, то программа может выбрать любой, в том числе и самый сложный, в котором человеку в практической партии будет трудно разобратся.

То же самое относится и к игре в сложных позициях. Если программа видит продолжение, приводящее к преимуществу или равенству, она его не «потеряет», какие бы ходы ни делал противник. Поэтому надо внимательно анализировать предлагаемые программой варианты и выбирать из них приемлемые для игры за доской.

ПРОГРАММЫ ВЫХОДЯТ НА СТАРТ

27-28 января 2007 года в Москве после более чем трехлетнего перерыва состоялся очередной турнир среди компьютерных программ, играющих в русские шашки. Этот турнир получил сложное название «1-й чемпионат Европы среди компьютерных программ по русским шашкам на призы сайта „Шашки в России“». Иными словами, это 2-й кубок сайта «Шашки в России» и одновременно первый неофициальный чемпионат Европы. Организаторы турнира — международный гроссмейстер А. Б. Кандауров и Секция-64 ФМЖД.

О прошедшем турнире и его итогах главный редактор «ШВ СЛШИ» **В. Н. Ростовиков** беседует с **Михаилом Глизериным**, автором популярных шашечных программ «TORNADO» и участником этого турнира.¹

— Михаил, поздравляю Вас с успешным выступлением — программа TORNADO III завоевала бронзовую медаль, повторив результат первого Кубка сайта, состоявшегося в 2003 году. Но если учесть, что в этот раз в турнире принимало участие всего три программы, довольны ли Вы игрой TORNADO и итогами турнира?

— Да, я доволен итогами турнира. Дело в том, что все три программы (TORNADO, Kallisto, Aurora) показали очень высокий класс игры и все партии, кроме одной, завершились вничью. В единственной результативной партии TORNADO уступила Kallisto, что и повлияло на итоговое распределение мест. Но в таком результате присутствует очень большой элемент случайности.

— Насколько мне известно, в турнире планировалось большее число участников. Но на

старт вышли всего три программы. В чём причина?

— Действительно, организаторы турнира рассчитывали на то, что будут играть как минимум семь программ (Aurora, Damira, Kallisto, Plus600, Skifi, Tornado, Tundra).

Но если С. Старцев (Plus600) объявил об отказе участвовать в турнире заранее, то о том, что не будут играть Tundra, Damira и Skifi, организаторы узнали перед самым началом турнира. Что-либо предпринимать было уже поздно и судейская коллегия решила проводить турнир из трех программ.

Причины отказов у всех авторов разные. Насколько мне известно, авторы «Тундры» не смогли принять участие по независящим от них обстоятельствам. Петербуржец С. Нефедов, автор новой шашечной программы Skifi, не успевал (по его словам) уложиться в сроки, хотя вначале и проявлял очень большой энтузиазм. Авторы

¹ Интервью опубликовано в журнале «Шашечный вестник СЛШИ», № 52 (февраль, 2007).

«Дамиры», также неоднократно подтверждавшие свое участие, аналогично сослались на «неготовность».

Но если со Skifi всё более или менее понятно, то для «Дамиры», которая позиционируется как программа «гроссмейстерского уровня» и давно продается, ссылка на «неготовность» выглядит, мягко говоря, неубедительно. Могу предположить, что авторы «Дамиры» просто не были уверены в том, что программа покажет заявленный ими уровень игры.

Была идея включить в состав несколько программ из тех, что участвовали в компьютерных турнирах в предыдущие годы (например, «Гроссмейстер Агафонов», «WinRuss», «Draughts Navigator»). Но так как эти программы в последние годы не развивались вообще, то они не смогли бы составить никакой конкуренции. Проще говоря, были бы поставщиками очков лидерам. Поэтому в итоге было решено, что буду играть три программы при большем количестве кругов.

— Вернемся к Вашей программе. Что изменилось в игровой программе TORNADO со времени участия в 1-м Кубке сайта «Шашки в России» в 2003 году?

— Изменилось очень многое. Если точнее, сейчас играла совсем другая программа. В 2003 году в Кубке выступала программа TORNADO II. После этого турнира я начал разрабатывать новую игровую программу, в которой движок был написан заново, практически с нуля.

И если вторая версия TORNADO по некоторым параметрам уступала сильнейшим на то время движ-

кам, то третья версия программы, имея намного более мощные алгоритмы, встала на один уровень с сильнейшими игровыми программами. Что собственно, и показал прошедший 2-й Кубок сайта.

Программа TORNADO III, выйдя в августе 2004 года, активно развивалась все последующее время и развивается сейчас.

— Принес ли прошедший турнир какую-нибудь пользу компьютерным шашкам и шашечному программированию?

— Безусловно, причем в нескольких аспектах. Если говорить, например, о TORNADO, то мне в процессе активной подготовки к турниру удалось значительно усилить программу, реализовать в игровых алгоритмах новые интересные идеи. Одних только тестовых партий между программами было сыграно в автоматическом режиме более 7000! Насколько я знаю, и И. Коршунов (Kallisto), и А. Свирин (Aurora) не сидели сложа руки.

Кроме того, турнир дал очень много пищи для размышлений о том, в каком направлении двигаться дальше, какие компоненты игровых движков развивать в первую очередь.

Вот такой пример. Несколько лет назад, во времена первых турниров, сильнейшие игровые движки рассчитывали не очень глубоко, но имели неплохие оценочные функции (например, «Гроссмейстер Агафонов», WinRuss). Спустя какое-то время появилась программа Магистр III (авторы — петербуржцы отец и сын Князевы), которая имела очень простую оценочную функцию, но считала на порядок глубже остальных. Результат — программа Магистр III

дважды становилась чемпионом России, переигрывая соперников самым убедительным образом.

С тех пор в программировании шашечных движков стал преобладать упор на «разгон» глубины расчета. Безусловно, работа над оценочными функциями тоже велась, и немалая, но всё-таки преобладало первое. В итоге мы имеем ситуацию, что глубина расчета у движков сейчас очень большая, оценочная же функция отстает. И это только один пример.

И, конечно, прошедший турнир наглядно показал, что соревнования среди программ необходимо проводить по другой системе, а именно играть большее число партий с маленьким контролем.

— Это поможет решить проблему низкой результативности?

— Да, сокращение контроля времени на партию и увеличение числа партий — одно из основных решений этой проблемы.

Возвращаясь к теме предыдущего вопроса хочу сказать, что мощный эмоциональный заряд для дальнейшей работы над шашечными программами дало мне и дружеское общение с участниками и зрителями турнира. Пусть их было мало, но они — настоящие энтузиасты.

— Я заметил одну интересную особенность — все программы играли с разными эндшпильными базами: TORNADO с 8-ми фигурной, Kallisto с 6-и фигурной, Auroga с 5-и фигурной. Но это не повлияло на результат турнира. Получается, что можно обойтись без «тяжёлых» многофигурных баз, занимающих, кстати, гигабайты дискового пространства?

— То, что существенная разница в эндшпильных базах (ЭБ) не повлияла на игру программ между собой стало еще одним важным практическим результатом турнира. Думаю, дело здесь опять же в значительно увеличившейся глубине расчета вариантов. Считая глубоко с самого начала партии, программы сводят к минимуму риск проиграть материал и всё завершается, в конце концов, ничьей.

Но это не значит, что «тяжелые» ЭБ теперь не нужны. Во-первых, игра программ между собой — увлекательное, но всё же несколько оторванное от «реальной жизни» действо. На практике игровые шашечные программы используются в основном для анализов партий и позиций. А здесь значение многофигурных ЭБ переоценить невозможно. В общем случае без них быстрый и точный анализ будет сильно затруднен, а в некоторых случаях и вовсе невозможен. Такой пример.

В русских шашках очень много ничейных дамочных окончаний с преимуществом у одной из сторон. И программа без ЭБ (или с маленькими ЭБ) будет стабильно показывать преимущество в одну или несколько шашек или дам. Чтобы понять, что это на самом деле ничья, может понадобиться очень много времени. Если же программа будет оснащена многофигурными ЭБ, результат анализа будет быстрым и однозначным — ничья.

Во-вторых, эндшпильные базы — один из самых действенных способов усиления игрового движка. Ведь чем больше ЭБ, тем раньше программа сможет «пробить» партию до результата. В перспекти-

ве «тяжелые» ЭБ помогут закрыть часть игры уже с самых первых ходов. Что, кстати, уже происходит в английском чекерсе.

— Вызвал ли розыгрыш Кубка среди программ зрительский интерес?

— К сожалению, нет. Из москвичей за два дня никто (!) не пришел в клуб, а всего было два иногородних зрителя — это Н. Гиматдинов из Ханты-Мансийска и С. Педько из Харькова. Они как раз были проездом в Москве. Ситуация очень странная, ведь шашечные программы уже почти массово вошли в «шашечный быт».

— Какие награды получили участники турнира?

— Александр Кандауров, автор сайта «Шашки в России», наградил призеров турнира медалями, а победитель получил переходящий Кубок. Кроме того, А. А. Леман наградил призеров дипломами ФМЖД.

— Где можно посмотреть партии турнира?

— Все партии, а также фотографии доступны на сайте «Шашки в России» (<http://www.shashki.com>).

— Михаил, немного отвлечемся от основной темы нашего интервью. Читатели «ШВ» часто интересуются Вашими программами, задают различные вопросы. Могут ли они обратиться к Вам напрямую?

— Да, могут. Всегда доступен сайт, на котором можно найти всю основную информацию о программах TORNADO. Адрес сайта <http://home.mari-el.ru/mlehrer/> Кроме того, можно написать мне на электронный адрес mlehrer@mari-el.ru

или связаться по телефону (8362) 42-19-82.

— Шашечные программы TORNADO — это не только игровой движок, но также базы партий и базы шашечных комбинаций. Развиваются ли эти программы?

— Безусловно развиваются. Более того, в планах на будущее намечена интеграция всех компонентов пакета TORNADO в один мощный программный комплекс.

В настоящее время разрабатывается новая шашечная программа с условным названием TORNADO IV. Она объединит в себе игровые движки для различных видов шашек, средства для просмотра и редактирования партий, а также базы партий, базы шашечных комбинаций и инструменты для работы с ними.

Также готовится к выпуску новая база, в которую войдут более **100.000 партий** в русские шашки.

— Возвращаясь к основной теме нашего разговора, хочу спросить, планируете ли Вы участвовать в дальнейшем в подобных турнирах? Какие цели ставите себе? Не боитесь возможных неудач?

— Да, я обязательно буду участвовать во всех турнирах среди компьютерных программ. Цель прежняя — быть лучшим. А что касается неудач... Я сам активно играющий шашист (*в 2006 году Михаилу присвоили звание мастера — прим. ред.*) и для меня турнирная борьба, победы и поражения — состояние привычное. Поэтому будем бороться!

— Желаю удачи!

ПАКЕТ ПРОГРАММ **TORNADO**

TORNADO III игровая шашечная программа

- Игровая программа гросс-мейстерского уровня
- Играет в русские шашки, бразильские шашки, пул-чекерс
- Быстрый и глубокий расчет
- Мощная оценочная функция
- Библиотека дебютов
- 8-ми фигурная база окончаний (русские шашки)
- 7-ми фигурная база окончаний (бразильские шашки, пул-чекерс)
- Специальные возможности для анализа партий и позиций

TORNADO Draughts Base II база шашечных партий

- Более 100.000 партий в русские шашки
- Быстрый поиск, удобный просмотр статистики
- Расширенные возможности поиска
- Три вида шашек: русские, бразильские, пул-чекерс
- Поддержка различных форматов PDN и TXT
- Работа с несколькими базами партий одновременно

TORNADO Tactics Assistant 1.0 база шашечных комбинаций

- Программа для изучения шашечных комбинаций и тактики на 64-х клеточной доске
- Более 2000 комбинаций в три вида шашек (русские, бразильские, пул-чекерс)
- Навигация по базе и просмотр шашечных комбинаций
- Поиск и сортировка комбинаций по их характерным признакам
- Обучающий модуль

Дополнительную информацию о программах TORNADO Вы можете получить у автора программ. Контактный телефон: (8362) 42-19-82.

E-mail: mlehrer@mari-el.ru Адрес: 424003, Республика Марий Эл, г. Йошкар-Ола, Ленинский проспект, д. 41, кв. 18. Глизерин Михаил Матвеевич
Сайт программ в сети Интернет: <http://home.mari-el.ru/mlehrer/index.htm>