

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Основы алгоритмизации и программирования»

«К защите допустить»
Руководитель курсового проекта
ассистент кафедры информатики
_____ М.В. Ганусевич
____.____.2024

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему:
«РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ИГРЫ В СЁГИ»

БГУИР КП 6-05-0612-02 022 ПЗ

Выполнил студент группы 353502
ПЕТРОЧЕНКО Михаил
Максимович

(подпись студента)

Курсовой проект представлен на
проверку _____.____.2024

(подпись студента)

Минск 2024

СОДЕРЖАНИЕ

Введение	3
1 Обзор правил игры и аналогов	4
1.1 Правила игры в сёги	4
1.1.1 Фигуры	4
1.1.2 Ходы	5
1.1.3 Переворот	6
1.1.4 Итог игры	6
1.2 Аналоги	7
2 Теоретическое обоснование разработки программного продукта	9
2.1 Обоснование необходимости разработки	9
2.2 Технологии программирования, используемые для решения поставленных задач	10
3 Проектирование функциональных возможностей программы	13
3.1 Общее описание подходов и алгоритмов	13
3.2 Представление доски	14
3.3 Алгоритм расстановки фигур на доске в соответствии с записью в нотации Фортсайта-Эдвардса для сёги	16
3.4 Алгоритм генерации ходов	17
3.5 Алгоритм сортировки списка ходов	18
3.6 Алгоритм оценки позиции	20
3.7 Алгоритм поиска наилучшего хода	20
Заключение	24
Список литературных источников	25
ПРИЛОЖЕНИЕ А (обязательное) Блок схема алгоритма, реализующего программное средство	26
ПРИЛОЖЕНИЕ Б (обязательное) Ведомость документов	27

ВВЕДЕНИЕ

На сегодняшний день стратегические игры, такие как шахматы и сёги, обретают новую актуальность в контексте цифровой эпохи. С развитием технологий и распространением мобильных устройств игровые приложения становятся все более доступными и популярными. Шахматы, как классическая стратегическая игра, и сёги, с их уникальными правилами и философией, предоставляют игрокам возможность не только развлечься, но и развивать свои умственные способности, стратегическое мышление и аналитические навыки.

На сегодняшний день игра стремительно набирает популярность. В Беларуси игра пользуется большим успехом. Также белорусские спортсмены имеют большие успехи в данной игре: последние 4 чемпионата Европы по сёги были выиграны белорусами (2019 год – Винсент Танян, 2021 – Сергей Корчицкий, 2022 – Владислав Закржевский, 2023 – Антон Старикевич).

Целью данной работы является разработка полноценного приложения для игры в сёги, способного удовлетворить запросы как опытных игроков, так и новичков, а также исследование особенностей процесса разработки игрового приложения в контексте выбранной темы.

Для достижения данной цели предусмотрены следующие задачи:

- 1 Анализ правил игры.
- 2 Анализ и разработка алгоритмов, необходимых для работы программы.
- 3 Разработка пользовательского интерфейса.

Пояснительная записка оформлена в соответствии с СТП 01-2017 [1].

1 ОБЗОР ПРАВИЛ ИГРЫ И АНАЛОГОВ

1.1 Правила игры в сёги

Сёги – традиционная стратегическая игра шахматного типа. Доска для данной игры представляет собой квадрат с сеткой из квадратных клеток (полей) 9×9. Все поля окрашены в один цвет.

1.1.1 Фигуры

Каждый игрок в начале партии имеет по 20 фигур: 1 король, 1 ладья, 1 слон, 2 золотых и 2 серебряных генерала, 2 коня, 2 копыта и 9 пешек. Фигуры имеют форму вытянутого пятиугольника, на обеих сторонах которого записано название основной и превращенной фигуры иероглифами. Фигуры не отличаются по цвету, их принадлежность определяет направление: фигуры расположены острием к противнику. Размер фигуры тем больше, чем она важнее. Внешний вид и название фигур приведены в таблице 1.1

Таблица 1.1 – Виды фигур в сёги

<i>Основная фигура</i>	<i>Внешний вид</i>	<i>Описание хода</i>	<i>Перевернутая фигура</i>	<i>Внешний вид</i>	<i>Описание хода</i>
Король		На одно поле в любом направлении	-	-	-
Ладья		На любое количество полей по горизонтали или вертикали	Дракон		Как ладья и как король
Слон		На любое количество полей по диагонали	Лошадь		Как слон и как король
Золото		На любое соседнее поле, кроме полей сзади по диагонали	-	-	-

Продолжение таблицы 1

Основная фигура	Внешний вид	Описание хода	Перевернутая фигура	Внешний вид	Описание хода
Серебро		На любое соседнее поле, кроме полей слева, справа и снизу	Перевернутое серебро		Как золото
Конь		На два поля вперед и на одно влево или вправо. Может перескакивать фигуры	Перевернутый конь		Как золото
Копьё		На любое количество полей по вертикали вперед	Перевернутое копьё		Как золото
Пешка		На одно поле вперед	Токин		Как золото

1.1.2 Ходы

Партию в сёги играют два игрока, называемые сэнтэ (яп. «ходящие прежде») или же черные и готэ (яп. «ходящие позже») или же белые.

Ходы делаются по очереди, запрещается делать два хода подряд или пропускать собственный ход. Ходы в сёги подразделяются на два типа: обычные (ходы на доске) и ходы со сбросом. Обычный ход – перенос фигуры в соответствии с правилами ее хода. При перемещении фигуры на поле, занятое фигурой противника, фигура противника снимается с доски и переносится в резерв взявшего ее игрока. При наличии в резерве фигур, игрок может выставить ее на поле основной стороной вверх и острием, направленным в сторону врага. Далее эта фигура играет на стороне сбросившего ее игрока. Сброс фигуры является полноценным ходом, поэтому игрок может или выставить фигуру на доску, или походить любой из уже

присутствующих на доске фигур. Существует несколько правил, касающихся сброса фигур. Игрок может сбросить фигуру на любое свободное поле за исключением следующих ситуаций:

- запрещено сбрасывать пешку на вертикаль, где уже есть союзная пешка;
- запрещено ставить мат выставлением пешки
- запрещено выставлять фигуру на поле, на котором у нее не будет возможности хода по правилам.

1.1.3 Переворот

Первые и последние три горизонтали считаются лагерем одной из сторон. При входе в лагерь врага, выходе из него или ходе внутри лагеря, союзная фигура может перевернуться или же превратиться. При превращении фигура переворачивается и меняет своё поведение в соответствии с правилами превращения. Превращения фигур описаны в таблице 1.

Превращение фигуры не обязательно, если после хода у фигуры остаются возможные ходы. Например, при ходе пешкой на седьмую и восьмую горизонтали, игрок может по желанию перевернуть ее или же оставить в основном состоянии, но при ходе на последнюю горизонталь он обязан превратить ее в токина, так как пешка на девятой горизонтали не имеет хода. Фигура остается перевернутой до взятия ее другим игроком, после же она может быть выставлена только основной стороной вверх. Переворот при сбросе невозможен, поэтому сброс фигуры на поле, с которого она не будет иметь хода запрещен правилами.

1.1.4 Итог игры

Игра заканчивается, когда один из игроков ставит «мат» другому, то есть ставит его в положение, когда король находится под атакой и не может уйти из-под боя или закрыться ходом другой фигуры или выставлением фигуры. Также игра заканчивается, если один из игроков сдается во время игры. В этом случае побеждает его противник. Также игра заканчивается при фоле – произведенном ходе, противоречащем правилам, но в электронном варианте игры чаще всего подобные ходы сделать невозможно. Ничейные результаты в сёги невозможны, каждая партия результативна. Существуют меры, препятствующие затягиванию игры:

- 1 В случае четырехкратного повторения позиции партия прерывается и

начинается заново со сменой очередности хода, но с сохранением времени игроков.

2 Если позиция повторилась в результате трех шахов подряд, то на четвертый раз шахующий игрок обязан выбрать другой ход, иначе ему будет засчитано поражение.

3 Если оба короля укрепились в лагерях противоположной стороны, исход партии решается подсчетом очков.

1.2 Аналоги

На сегодняшний день существует несколько приложений и сервисов для игры в сёги. Среди онлайн сервисов наиболее популярными являются lishogi.org и 81dojo.com.

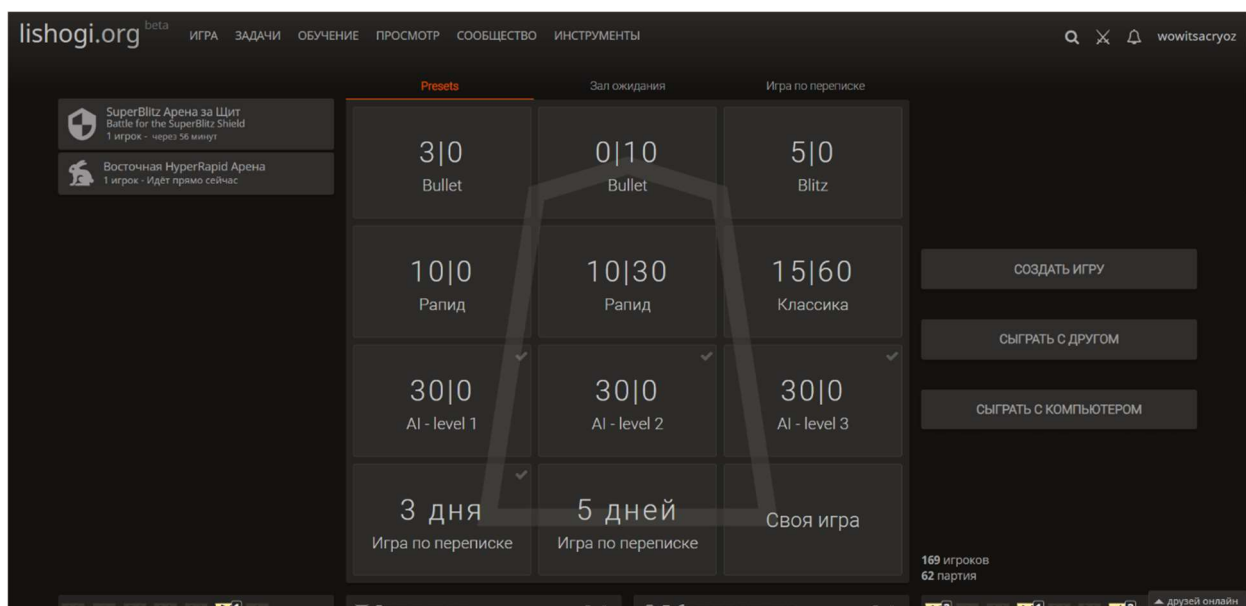


Рисунок 1.1 – Пользовательский интерфейс сервиса lishogi

Сервис lishogi.org является наиболее популярным. За март 2024 года его посетили 266 тысяч человек. Он наиболее хорош для игры с компьютерными соперниками и тактических тренировок, но при игре с реальными людьми бывает трудно найти соперника.



Рисунок 1.2 – Пользовательский интерфейс сервиса 81dojo

Сервис 81dojo.com – японский сервис для игры в сеги, который также имеет раздел для игры с пользователями не из Японии. Сайт имеет типичный для японских форумов интерфейс. За март 2024 его посетили 183 тысячи человек. Этот сервис подходит для игры и последующего разбора партии с соперником, но найти противника начального уровня – нетривиальная задача.

Мобильные приложения для игры в сеги не очень распространены и не отличаются популярностью.

2 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА

2.1 Обоснование необходимости разработки

В современном обществе стратегические игры играют важную роль в сплочении людей и повышении их интеллектуального уровня. В 2020 году в мире начался так называемый шахматный бум. Самый популярный сервис для игры в шахматы, Chess.com, зафиксировал почти двухкратный рост числа пользователей. Статистика, собранная командой сервиса, представлена на рисунке 2.1.

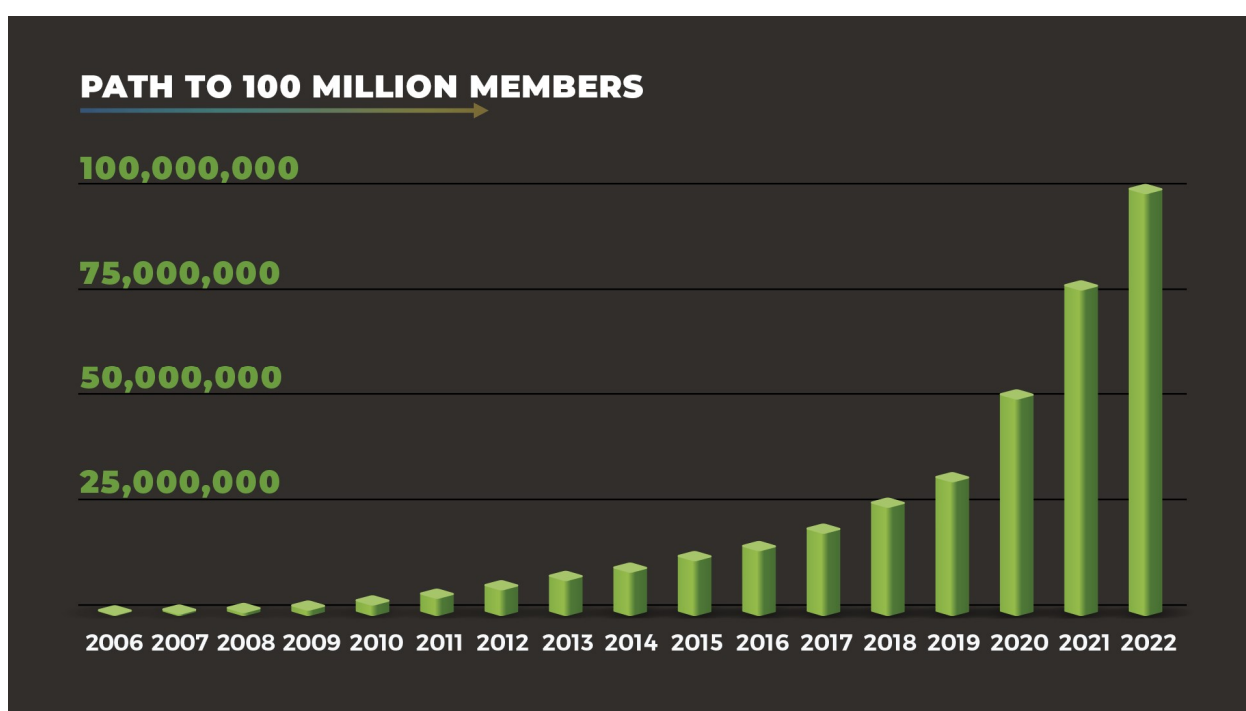


Рисунок 2.1 – Статистика количества пользователей сервиса Chess.com

Данная статистика показывает востребованность стратегических игр в современном обществе. Но сёги, из-за специфики японской культуры и невысокого интереса со стороны западной общественности, пока только набирают популярность как аналог шахмат. Именно этим объясняется малое количество сервисов для игры и их несовершенство.

Также стоит отметить пользу игры в сёги и другие стратегические игры для мозга и интеллектуального развития. Стратегические игры развивают человеческий мозг во множестве направлений.

Основные благоприятные воздействия сёги на человека:

4 Развитие памяти. Игра в сёги вынуждает задействовать долговременную память для воспроизведения дебютных идей и крепостей, а также кратковременную память для поиска наиболее выгодных из данной позиции ходов. Также игра тренирует распознавание паттернов, так как многие тактики сводятся к применению типовых тактических приемов (вилка, батарея и т.д.).

5 Креативность. Игра в сёги активно стимулирует развитие нестандартного мышления. В процессе игры игрок старается придумать нетипичные тактики, такие как жертва фигур и различные ловушки.

6 Логическое мышление. Последовательность ходов в сёги представляет собой строгую логическую систему с предустановленными правилами, но вместе с тем позиция на доске и ход игры изменяется с каждым ходом. Анализ взаимодействия фигур на доске задействует оба полушария мозга.

7 Обучаемость. Поражение в игре может стать отличной мотивацией для анализа партии и улучшения общего навыка. Подобный подход широко применим и по отношению ко многим реальным сценариям обучения и самосовершенствования.

Основные выгоды разработки программного продукта:

1 Привлечение внимания к игре сёги, ее популяризация.

2 Глубокое изучение процесса разработки движков для игр шахматного типа, принципов их работы и алгоритмов, лежащих в основе данных движков.

3 Повышение общего уровня интеллектуального здоровья.

Обобщая вышеизложенное, разработка программного продукта, оправдана с точки зрения общественной пользы, а также с точки зрения получаемых во время выполнения курсовой работы знаний и навыков.

2.2 Технологии программирования, используемые для решения поставленных задач

Выбор языка программирования – одно из важнейших решений при разработке проекта. Для реализации задания данной курсовой работы был выбран язык программирования C++.

C++ – один из наиболее популярных и стабильных языков программирования. Основные характеристики, делающие его наиболее подходящим для проекта – это высокая производительность и объектно-ориентированность. Высокая производительность программ, написанных на данном языке, позволяет реализовать алгоритм поиска наилучшего хода,

который, в силу специфики игры, является перебором всех возможных позиций на несколько ходов вперед, что требует значительных временных затрат и вычислительных мощностей. Объектно-ориентированность языка облегчает проектирование проекта, так как позволяет определить четкую иерархию сущностей, определяющую в том числе и их функционал, и графического пользовательского интерфейса. Также C++ является высокоуровневым языком программирования, благодаря чему обеспечивается легкость и удобство написания кода, что является немаловажным аспектом написания сложных программных продуктов.

C++ обладает обширной экосистемой библиотек и технологий, которые значительно упрощают процесс разработки. От стандартных библиотек, таких как *STL (Standard Template Library)*, до различных фреймворков и инструментов, C++ предоставляет разработчикам множество возможностей для ускорения и улучшения процесса создания программ.

Язык C++ предоставляет разработчикам полный контроль над управлением памятью, что критически важно при создании высокопроизводительных вычислительных алгоритмов, где эффективное использование ресурсов играет решающую роль.

Такой выбор является компромиссом между простотой написания кода и высокой производительностью, а также предоставляет доступ к обширному набору библиотек и технологий. Поэтому C++ становится идеальным решением для разработки программных продуктов, где важны эффективность, высокая производительность и удобство разработки.

В процессе выполнения курсового проекта для проектирования графического пользовательского интерфейса был использован фреймворк Qt.



Рисунок 2.2 – логотип фреймворка Qt

Qt – это фреймворк для разработки кроссплатформенного программного

обеспечения, в том числе предоставляя инструменты для работы с графическим пользовательским интерфейсом. Изначально фреймворк создавался для работы с помощью языка C++, но на сегодняшний день существует множество библиотек для различных языков программирования, позволяющих использовать возможности Qt.

Примером таких библиотек могут послужить PyQt и PySide для языка программирования Python, QtRuby для языка Ruby и PHP-Qt для PHP. Фреймворк активно развивается и улучшается, актуальный стандарт на момент написания – версия 6.7, выпущенная 2 апреля 2024 года.

Qt часто используется в создании приложений для настольных компьютеров так как позволяет запускать написанное с помощью него программное обеспечение путем компиляции для каждой из поддерживаемых операционных систем без необходимости изменения исходного кода. Существуют версии библиотеки для Microsoft Windows, UNIX-подобных систем с графической подсистемой X11, Android, iOS, macOS, QNX, встраиваемых Linux-систем и платформы S60.

3 ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ПРОГРАММЫ

3.1 Общее описание подходов и алгоритмов

В настоящее время одним из обязательных атрибутов приложения является опыт взаимодействия пользователя с приложением. Для достижения максимального удобства использования необходимо тщательно выбрать алгоритмы, обеспечивающие работу приложения. В данном разделе будут рассмотрены некоторые алгоритмы и их оптимизации с целью получения наиболее низкого времени ответа от компьютерного оппонента.

Движки для игры в сёги и другие игры шахматного типа должны решать несколько основных задач. Данные задачи это:

- Представление доски;
- Поиск;
- Оценка.

Также большинство движков реализуют «книгу» открывающих ходов (дебютов) и таблицы окончаний (эндшпилей).

Понятие «эндшпиль» в сёги не используется, однако так или иначе партия в сёги может быть разделена на те же 3 этапа, что и шахматная партия. Основное различие при этом заключается в том, что стадия эндшпиля трудно поддается подсчету, так как фигуры не уходят с доски навсегда. Поэтому в отличие от шахмат, где все возможные партии с девятью и менее фигурами на доске были просчитаны компьютером и решены, конечная стадия игры в сёги отличается гораздо большим пространством для тактических манёвров. Исходя из этих соображений таблицы окончаний не будут реализованы в рамках данного проекта. Дебютная теория в сёги так же отличается от шахматной. Доскональное теоретическое изучение дебюта на многие ходы вперед имеет смысл только на самом высоком уровне игры из-за гораздо большего количества возможных ходов. Поэтому при реализации движка будет рассмотрено ограниченное количество идей открывающих ходов.

При выполнении курсового проекта были реализованы следующие алгоритмы:

1 Алгоритм расстановки фигур на доске в соответствии с записью в нотации Форсайта-Эдвардса для сёги (SFEN). Данная нотация является международным стандартом для записи позиций для игры в сёги.

2 Алгоритм генерации всех возможных ходов в позиции с учетом всех правил игры в сёги.

3 Алгоритм, сортирующий ходы в соответствии с их силой. Данный

алгоритм использует набор эвристических параметров для улучшения работы по поиску сильнейшего хода. Это одна из техник оптимизации алгоритма поиска наилучшего хода.

4 Алгоритм оценки позиции на доске. Процесс поиска наилучшего хода в данной позиции неразрывно связан с оценкой позиции на доске, так как сильнейший ход – тот ход, который приводит к наилучшей для игрока позиции на доске.

5 Поиск наилучшего хода в позиции. Данный алгоритм анализирует все возможные ходы, и выбирает наилучший. Для него существует несколько оптимизаций, так как очевидно, что рассматривать все ходы не имеет смысла.

3.2 Представление доски

Существует два основных подхода к представлению доски: поле-ориентированный (square-centric) и фигуρο-ориентированный (piece-centric). Их различия заключаются в выборе основной сущности для представления доски.

При поле-ориентированном подходе доска рассматривается как набор полей, которые могут быть заняты какой-либо фигурой или не быть заняты вообще. Видов представления, применяющих этот подход заключается в представлении доски в виде массива, а фигур – в виде чисел. К поле-ориентированным видам представления относятся доска 9x9 – представление доски в виде обычного двумерного массива, где каждому полю соответствует одно число, а также доска 11x9 – представление доски в виде расширенного массива, где дополнительные вертикали используются в как сигнальные значения для проверки на выход коня за пределы игрового поля. Преимуществом данного подхода является простота его реализации, недостатком же является то, что наименьшее количество пустых полей равно 41, а значит в любой позиции при обходе доски как массива программа совершает больше половины итераций без получения результата.

При применении фигуρο-ориентированного подхода доска рассматривается как набор фигур, имеющих некоторые координаты. К видам представления данного типа относятся битовые доски и списки фигур.

Битовая доска – представление доски в виде множества битов, где бит равный единице обозначает наличие фигуры или присутствие любой другой подобной метрики. Основным преимуществом битовых досок является возможность проведения над ними бинарных логических операций. Например, использовав операцию объединения на все битовые доски,

хранящие карту контролируемых фигурой полей, можно получить карту всех контролируемых игроком полей, а при использовании операции пересечения на битовую доску, содержащую позиции всех фигур противника, и битовую доску, содержащую карту атаки союзной фигуры можно легко понять, какие фигуры соперника атакованы союзной фигурой. Использование логических операций позволяет не только получать требуемые поля легким и интуитивным способом, но и позволяет повысить эффективность, так как их выполнение на современных системах весьма оптимизированно. Также можно представить битовое поле в виде обычного беззнакового целого числа. Это сокращает затраты памяти. Пример применения битовых полей показан на рисунке 3.1.

дракон атакует	&	фигуры соперника	=	атакованные фигуры
. . . 1		1 . . 1 1 . . 1
. . . 1		1 . 1 1 1 1 1 1
. . . 1 1 1
. . 1 1 1
1 1 1 * 1 1 1 1 1	&	. . . * . . 1 . .	=	. . . * . . 1 . .
. . 1 1 1
. . . 1
. . . 1
. . . 1

Рисунок 3.1 – Пример применения бинарных логических операций к битовым доскам

Однако этот подход имеет ряд недостатков, которые перекрывают преимущества от использования. Если в случае шахмат для кодирования доски идеально подходит тип `unsigned long long`, имеющий ровно 64 бита, то в случае сёги подойдет только 128-битные типы данных. Но актуальный на момент написания стандарт языка не поддерживает в полной мере подобные типы данных и применение их связано с некоторыми неудобствами.

Списки фигур – вид представления, при котором доска рассматривается как набор фигур, которые стоят на каких-либо полях. Использование данной идеи позволяет тратить почти в два раза меньше времени на итерирование по всем фигурам. Однако, зная лишь номер поля, узнать, находится ли на нем фигура и если находится, то какая, становится нетривиальной задачей, особенно в условиях обхода списка доступных ходов, состоящего из сотен значений.

В подавляющем большинстве современных движков используются гибридные способы, включающие в себя виды представления из обоих подходов. Например, при использовании битовых досок или списков фигур часто используется также обычный двумерный массив для быстрого

получения информации о конкретном поле.

При разработке курсового проекта был использован именно гибридный подход к представлению доски.

3.3 Алгоритм расстановки фигур на доске в соответствии с записью в нотации Фортсайта-Эдвардса для сёги

Запись в нотации Фортсайта-Эдвардса позволяет представить позицию на доске с помощью строки из ASCII-символов. Строка разделяется на 3 основных части:

- запись позиции на доске;
- кому принадлежит право хода;
- состояние руки обоих игроков.

Каждая фигура кроме коня обозначается первой буквой своего названия (на английском языке), так как первая буква его названия совпадает с первой буквой названия короля. Таким образом обозначения фигур приведены в таблице 3.1.

Таблица 3.1 – Обозначения фигур

Название фигуры в русском языке	Название фигуры в английском языке	Обозначение фигуры
Пешка	Pawn	P
Копьё	Lance	L
Конь	Knight	N
Серебро	Silver	S
Золото	Gold	G
Слон	Bishop	B
Ладья	Rook	R
Король	King	K

Перевернутые фигуры обозначаются как основные, но перед обозначением добавляется '+'. Фигуры сэнтэ (чёрных) обозначаются буквами в нижнем регистре, фигуры готэ (белых) в верхнем. Пример обозначения фигуры: b – чёрный слон, +R – белый дракон (перевернутая ладья). Позиция записывается с точки зрения чёрных.

Принадлежность права хода обозначается одной буквой: 'b' – чёрные (black), 'w' – белые (white).

Третье поле описывает состояние руки обоих игроков. Фигуры в руке чёрных обозначаются буквами в верхнем регистре, а в руке белых – в нижнем. Пример записи позиции, полученной из дебюта «Убийца демонов»: `ln1gkg1nl/1rls5/p1p1ppspp/1p1p2p2/2PNB4/9/PP1PPPPPP/7R1/L1SGKGSNL b`
b. Данная позиция также представлена на рисунке 3.2.



Рисунок 3.2 – Позиция, полученная из дебюта «Убийца демонов».

На рисунке слева находится рука белых, справа – черных. Право хода принадлежит черным. Доска представлена с точки зрения черных.

3.4 Алгоритм генерации ходов

Данный алгоритм использует оба подхода к представлению доски. Доска представляет собой массив, где каждому индексу соответствует единственное поле. Значения индексов каждого поля показано на рисунке 3.3. Тогда перемещение на доске в различных направлениях можно охарактеризовать значением, которое нужно прибавить к текущему полю. Данная концепция проиллюстрирована на рисунке 3.4.

8	7	6	5	4	3	2	1	0
17	16	15	14	13	12	11	10	9
26	25	24	23	22	21	20	19	18
35	34	33	32	31	30	29	28	27
44	43	42	41	40	39	38	37	36
53	52	51	50	49	48	47	46	45
62	61	60	59	58	57	56	55	54
71	70	69	68	67	66	65	64	63
80	79	78	77	76	75	74	73	72

Рисунок 3.3 – Соответствие индексов массива полям на доске

-8	-9	-10
+1		-1
+10	+9	+8

Рисунок 3.4 – Значения изменения номера поля в зависимости от направления движения

Для перемещения фигуры на север на одно поле нужно записать значение, соответствующее фигуре, по индексу данной фигуры -8. Стоит заметить, что направления определяются с точки зрения чёрных, поэтому белым для передвижения пешки на 1 поле нужно будет двигаться на юг.

Алгоритм проходит по спискам фигур, передвигая их во всех возможных направлениях и записывая конечный результат в вектор. Таким образом алгоритм использует и фигуρο-ориентированный подход.

3.5 Алгоритм сортировки списка ходов

Для корректной работы алгоритма поиска наилучшего хода (алгоритма

минимакса), а точнее его оптимизации, называемой альфа-бета отсечением, используется эвристический алгоритм, присваивающий каждому ходу некоторую оценку, зависящую от ряда условий. После список ходов сортируется в соответствии с данной алгоритмом оценкой.

Список условий для оценки хода и соответствующих значений оценки представлен в таблице 3.1.

Таблица 3.1 – Оценки хода

Критерий оценки	Оценка
Результатом хода является переворот фигуры	+ стоимость перевернутой фигуры
Результатом хода является взятие фигуры	+ 10 * (стоимость взятой фигуры - стоимость берущей фигуры)
Поле назначения контролируется пешкой соперника	- стоимость движущейся фигуры
Ход фигурой на желательное поле	Зависит от поля, пример представлен на рисунке 3.5

9	8	7	6	5	4	3	2	1	
50	60	40	40	40	40	40	50	60	a
70	60	70	80	90	80	70	90	70	b
40	10	70	70	100	70	70	100	40	c
60	10	70	70	90	70	70	90	60	d
60	20	70	70	80	70	70	90	60	e
80	30	90	70	80	70	60	90	80	f
70	90	60	70	70	60	60	90	70	g
20	20	20	20	20	20	20	20	20	h
0	0	0	0	0	0	0	0	0	i

Рисунок 3.5 значения бонуса к оценке для пешки

3.6 Алгоритм оценки позиции

Оценка позиции проходит по сумме критериев. Основным является количество материала у обеих сторон. Фигуры на доске и в руке имеют различную стоимость, выражаемую в пешках. Стоимости всех фигур представлены в таблице 3.2.

Таблица 3.2 – Стоимость фигур

Фигура	Стоимость
Пешка	1
Копьё	4
Конь	5
Серебро	7
Золото	8
Слон	11
Ладья	13
Токин	10
Перевернутое копье	9
Перевернутый конь	9
Перевернутое серебро	8
Лошадь	15
Дракон	17

Так как король – наиболее важная из фигур, ему присуждается высокая цена, чтобы точно дать понять, что потеря его недопустима. В данном курсовом проекте выбрано значение 500, но это может быть любое число, сильно большее стоимости других фигур.

Также важным критерием является контролируемое пространство. В расчет берется количество полей, контролируемых пешками и другими фигурами.

3.7 Алгоритм поиска наилучшего хода

Поиск наилучшего хода в данном проекте осуществляется с помощью алгоритма негатакс (negamax) с использованием альфа-бета отсечения. Сам алгоритм негатакс является улучшением алгоритма минимакс (minimax).

Минимакс – классический алгоритм теории игр, используемый для

минимизации негативных последствий для лица, принимающего решения, из тех, которые лицо не может предотвратить при развитии событий по наихудшему для него сценарию. Первоначально был сформулирован в 1713 году для игры двух лиц с нулевой суммой, то есть такой игрой, где выигрыши двух противостоящих лиц противоположны.

Алгоритм негемакс является вариантом алгоритма минимакс. Он базируется на свойстве игры с нулевой суммой. Алгоритм опирается на тот факт, что оценка позиции для игрока А является противоположной оценке позиции для игрока Б. Таким образом для получения наибольшей выгоды (победы) игрок А стремится к максимизации отрицательности позиции для соперника. Но та же логика применима к ходам игрока Б. Примером могут послужить шахматы, в которых преимущество черных обозначается отрицательным значением, в то время как преимущество белых – положительным. Нейтральной позиции в таком случае соответствует число ноль.

Представим, не ограничивая общности, упрощенную ситуацию, где из каждой позиции можно сделать только два хода. Тогда можно представить все возможные исходы в виде дерева. Пусть черные вершины означают, что в данной позиции право хода принадлежит черным, а белые означают, что следующий ход за белыми. Данный граф представлен на рисунке 3.6.

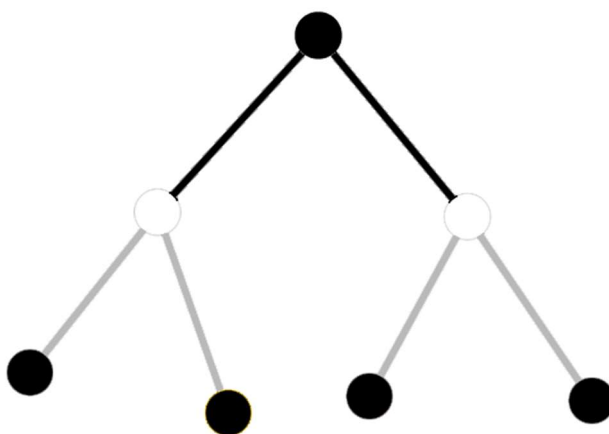


Рисунок 3.6 – Пример дерева возможных позиций в упрощенном варианте

Дерево ходов может быть расширенно до сколь угодно больших размеров, даже до конца игры, но сохранение такого числа позиций в памяти не представляется возможным, особенно для такой игры как сёги. Если в шахматах в среднем из каждой позиции может быть сыграно 35 различных ходов, то для сёги данное значение достигает 80. Общее же количество

позиций для сёги превышает это значение для шахмат более чем в 10^{100} раз (10^{224} против 10^{103}). Также обход такого дерева не позволил бы получить сколь-либо быстрое вычисление результатов, а следовательно игра с подобной программой была бы невозможна.

После построения дерева нужно выполнить статическую оценку конечных узлов (листьев), то есть оценить позицию безотносительно возможных ходов. Алгоритм подобной оценки представлен в пункте 3.6.

После оценки листьев можно узнать значения узлов верхних уровней. Рассмотрим ситуацию, представленную на рисунке 3.7.

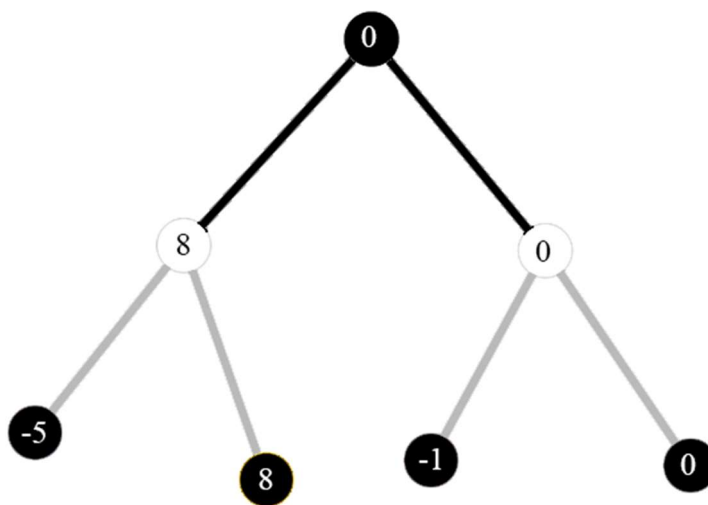


Рисунок 3.7 – Пример дерева позиций с оценками

Подъем начнем с узлов со значениями -5 и 8. Так как задача белых – максимизировать оценку, игрок выберет тот вариант, при котором оценка будет 8. Для оставшихся листьев очевидно, что игрок белыми выберет вариант с оценкой 0. Исходя из этого игрок чёрными будет стараться минимизировать оценку (отсюда и название алгоритма – минимакс) и выберет вариант 0. Хотя наилучший исход для игрока черными это -5, очевидно, что игрок белыми не допустит подобного сценария развития игры.

Альфа-бета отсечение – алгоритм поиска, стремящийся сократить количество позиций (узлов), рассмотренных при работе алгоритма минимакс или его аналогов. Его идея заключается в том, что вычисление узлов по некоторому пути можно прекратить, если было выяснено, что для этого пути значение оценки в любом случае хуже, чем для уже рассмотренной, и в дальнейших позициях результат может идти только в сторону ухудшения ситуации.

Преимущество альфа-бета отсечения заключается в том, что многие

ветви могут быть отсечены уже после рассмотрения одной. При этом очень важна предварительная сортировка элементов, так как чем больше с начала будет рассмотрено выигрышных вариантов, тем больше проигрышных может быть отсечено. Именно для этой цели была составлена эвристика, описанная в пункте 3.5. Рассмотрим действие алгоритма с альфа-бета отсечением. Для этого немного расширим уже рассмотренное дерево.

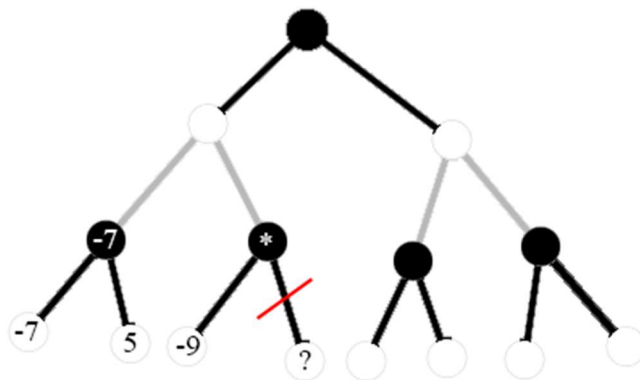


Рисунок 3.8 – Пример работы альфа-бета отсечения

После рассмотрения первых двух листов минимальная оценка, которую может получить игрок чёрными равна -7. При рассмотрении третьего листа становится очевидно, что значение узла, отмеченного звездочкой, меньше либо равно -9. Такая ситуация не устраивает игрока белыми, значит узел со знаком вопроса отсекается и не требует дополнительных вычислений. Становится также очевидна и необходимость сортировки – рассмотрев более «хорошие» варианты алгоритм может отсечь больше «плохих». Данный подход является оптимизацией так как не влияет на результат работы алгоритма, а лишь уменьшает количество действий, совершаемое им.

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

[1] Доманов, А. Т. Стандарт предприятия / А. Т. Доманов, Н. И. Сорока. – Минск : БГУИР, 2017. – 167 с.

ПРИЛОЖЕНИЕ А
(обязательное)
Блок схема алгоритма,
реализующего программное средство

ПРИЛОЖЕНИЕ Б
(обязательное)
Ведомость документов