

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**РАЗРАБОТКА КОМПЬЮТЕРНОЙ ИГРЫ-СИМУЛЯТОРА  
ИСТОРИЧЕСКОГО ВАРИАНТА ШАХМАТ «ГАЛА» С  
КОМПЬЮТЕРНЫМ ПРОТИВНИКОМ  
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование  
информационных систем

факультета компьютерных наук и информационных технологий

Сорокина Дмитрия Андреевича

Научный руководитель:

зав.кафедрой к. ф.-м. н., доцент

\_\_\_\_\_

М.В. Огнева

подпись, дата

Зав. кафедрой:

к. ф.-м. н., доцент

\_\_\_\_\_

М.В. Огнева

подпись, дата

Саратов 2021

## **ВВЕДЕНИЕ**

### **Актуальность темы.**

В последние годы неотъемлемой частью современной культуры постепенно становятся компьютерные игры. Приложения, от расширяющих кругозор до развлекательных, помогающих хорошо провести досуг, на самых разнообразных платформах крайне глубоко вошли в жизнь каждого человека.

Компьютерная игра – это имитация определенной деятельности, будь то гонки, управление самолетом, стратегии, квесты. Всё это существует и в реальной жизни, но в играх либо упрощено, либо гипертрофировано с целью вызова интереса пользователя. Важно учитывать два аспекта игр. Первый – получение удовольствия от процесса игры. Второй – сама деятельность, имитируемая в процессе игры и способствующая развитию определенных навыков. Отдельно можно выделить обучающие игры, а также программы-симуляторы, акцентирующие свое внимание на втором аспекте. Игровая форма позволяет пользователю более эффективно заниматься саморазвитием. В отличие от иных форм обучения (например, занятий в школе) отсутствие рамок и жестких правил помогает поддерживать интерес к процессу. Особенно это важно для нормального развития детей и подростков.

**Цель бакалаврской работы** – разработка компьютерной игры-симулятора исторического варианта шахмат «Гала» с компьютерным противником.

Поставленная цель определила **следующие задачи**:

1. Произвести поиск и анализ существующих игр-симуляторов шахмат.
2. Выбрать язык программирования и ознакомиться с источниками информации, связанными с ним.
3. Составить описание игры.
4. Выполнить реализацию игры-симулятора.
5. Протестировать полученную игру.

**Методологические основы** разработки приложения на языке С# представлены в работах Э. Троелсена, А.П. Климова, Г. Шилдта.

**Структура и объём работы.** Бакалаврская работа состоит из введения, пяти разделов, заключения, списка использованных источников и одного приложения. Общий объем работы – 94 страницы, из них 48 страниц – основное содержание, включая 6 рисунков, список использованных источников информации – 20 наименований.

### **КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ**

**Первый раздел «Обзор игр-симуляторов шахмат»** посвящен поиску и анализу существующих игр-симуляторов шахмат. В сети Интернет в свободном доступе можно найти множество игр и онлайн-ресурсов, выполняющих схожую функцию воссоздания шахматных игр в электронном формате. В рамках работы было рассмотрено три из них.

Единственная найденная реализация игры «Гала» была размещена на сайте [www.zillions-of-games.com](http://www.zillions-of-games.com) в 2006-м году. Автором является пользователь данного сайта Матс Винтер (Mats Winther). Других реализаций, в том числе от официальных издателей, обнаружить не удалось. Для пользования игрой требуется совершить покупку основного приложения «Zillions», представляющего собой сборник и конструктор игр, на этом же сайте. Основной особенностью программы является наличие внесенных самим же его автором изменений в правила игры, что можно считать недостатком, однако они объясняются автором как необходимые для упрощения процесса игры и уменьшения времени, затрачиваемого на одну партию.

Шахматный движок представляет собой компьютерную программу, способную играть в шахматы при помощи анализа доступных ходов. Первая программа, способная в полном объеме выполнять эти функции, была написана в 1951-м году английским ученым Аланом Тьюрингом. В

дальнейшем подобные программы улучшались, а к 2005-му году начали превосходить по силе наиболее опытных игроков-людей. В настоящее время существует огромное количество компьютерных программ, способных победить в матче гроссмейстера – игрока высшим шахматным званием. До изобретения подобных систем стратегии изучались и отрабатывались годами через игры людей друг против друга (как в шахматах, так и во многих других настольных играх). Компьютерные противники позволили ускорить процесс путем генерирования новых тактик и более глубокого «понимания» игры в сравнении с человеком. Шахматный движок Stockfish представляет собой реализацию алгоритма MiniMax с альфа-бета-отсечением и прочими модификациями. Он имеет открытый исходный код, что позволяет сообществу вносить свой вклад в совершенствование алгоритма. Это является главным преимуществом алгоритма над аналогами.

Шахматный движок AlphaZero был разработан компанией DeepMind и успешно сыграл свои первые матчи против других движков в конце 2017-го года. Алгоритм AlphaZero основан на использовании глубокой нейронной сети и обучения с подкреплением. Нейронная сеть получает в качестве входных данных текущее расположение фигур на игровом поле и возвращает вектор значений для каждого из вариантов ходов, являющихся оценкой возможного результата игры при выборе данного варианта. Алгоритм «учится» определять эти оценки исключительно через самостоятельную игру. Человек в процессе обучения не участвует, что позволяет алгоритму освободиться от рамок предвзятого мышления. Сами оценки используются в будущих партиях. В отличие от Stockfish, использующего алгоритм с альфа-бета-отсечением, AlphaZero использует метод Монте-Карло для поиска в дереве. Процесс поиска представляет собой серию симулированных самостоятельно сыгранных партий, через которые осуществляется движение от корня дерева в виде текущей ситуации на игровом поле до достижения листа дерева. На основе результатов симуляции и вычисляются выходные значения алгоритма. Главным недостатком данного алгоритма можно назвать

отсутствие его в публичном доступе. Следовательно, использовать его на данный момент не представляется возможным.

**Второй раздел «Обзор средств разработки приложения»** посвящен обзору выбранного языка программирования и иных средств разработки программы. Для разработки приложения был выбран язык C#. Это современный объектно-ориентированный язык программирования высокого уровня. Структура игры представляет собой набор компонентов в форме автономных и самоописательных пакетов, реализующих отдельные функциональные возможности. Код приложения состоит из нескольких файлов – пользовательских библиотек классов, отвечающих за различные аспекты игры. В соответствии с необходимостью и ролью часть методов и переменных скрыта при помощи соответствующего спецификатора (`private`) от изменения извне. Это позволяет избежать нарушения корректности данных и оказания негативного воздействия. Для повышения производительности программы использовались встроенные возможности языка C# по созданию многопоточного приложения.

**Третий раздел «Правила разработанной игры-симулятора “Гала”»** посвящен рассмотрению правил игры, реализуемых в разработанном приложении. Игра является пошаговой и рассчитана на двух игроков, в роли одного или двух из которых может выступать компьютерный противник. Целью игры, а также условием победы является перемещение двух главных фигур – короля и королевы – на четыре центральные клетки игрового поля. Шахматная доска имеет размер 10 на 10 клеток, что на две клетки больше, чем у стандартных шахмат (8 на 8 клеток). Игровое поле при этом можно разделить на пять зон. Первые четыре зоны угловые зоны смежны с каждым из четырех углов шахматной доски. Последняя центральная зона представляет собой крест с шириной линий в две клетки. Поведение фигур может изменяться, если они попадают в центральную зону.

**Четвертый раздел «Анализ функционала разработанного приложения»** посвящен обзору реализованных функциональных возможностей приложения. Игра-симулятор шахмат «Гала» имеет набор игровых режимов и настроек, позволяющих подстроить игру под себя. В первую очередь, следует выделить существующие игровые режимы: человек против человека, человек против компьютера, компьютер против компьютера. Компьютерный игрок использует для выбора оптимального хода алгоритм MiniMax с альфа-бета-отсечением. Принцип работы алгоритма основан на рекурсивном выполнении каждого из возможных ходов каждой из фигур на игровом поле. После выполнения определенного числа ходов за всех игроков происходит оценка ситуации на игровом поле. Так как после каждого перемещения фигуры одного игрока возможно перемещение произвольной фигуры соперника, схему рекурсии можно представить в виде дерева, называемого деревом поиска.

Поиск правильного хода облегчается при помощи умной системы подсказок: выбор дружественной шахматной фигуры подсвечивает доступные ходы. При этом система учитывает необходимость защитить главную фигуру – короля или королеву – когда одна из этих фигур уничтожена. Такой подход позволяет быстрее адаптироваться к правилам игры.

У пользователя приложения имеется возможность самостоятельно расставить шахматные фигуры на игровом поле перед началом матча. Это можно использовать для рассмотрения определенных сценариев игры и тестирования новых стратегий. Для начала ручной расстановки фигур необходимо нажать на кнопку «Manual Board», затем выбрать желаемый режим игры из трех представленных выше. При этом текущий матч будет завершен, если он в данный момент в процессе. Далее, с помощью соответствующего меню можно выбрать фигуру определенного цвета и поместить её на игровое поле. Также доступна опция удаления ранее

поставленной фигуры. Для начала игры в этом режиме нужно разместить на шахматной доске короля и королеву обеих цветов.

Все совершенные во время игры ходы записываются в специальный журнал в следующем порядке: цвет совершившей ход фигуры, её название, координат начальной и конечной позиции в виде пары из буквы и цифры (две пары разграничены двоеточием). Также возможно добавление необязательных специальных символов (символ «+», если ход поставил под угрозу единственную оставшуюся главную фигуру противника, и символ «x», если текущим ходом была уничтожена фигура).

Помимо исполняемого файла, к игре также прилагается каталог data. В этом каталоге находится набор картинок в формате png, которые представляют собой изображения шахматных фигур.

**Пятый раздел «Обзор структуры и методов приложения»** посвящен рассмотрению общей структуры программы и методов реализованных классов. Код разработанного приложения разбит на несколько файлов, каждый из которых содержит класс со своими методами или иные типы данных. Все классы выполняют определенную функцию в соответствии с названием. Папка Data предназначена для хранения изображений шахматных фигур в формате png. Эти изображения загружаются в память с помощью класса GraphicsHandler и далее используются программой. Приложение с использованием Windows Forms было создано типовым образом при помощи соответствующих стандартных средств программы Visual Studio 2017. При этом был автоматически создан класс Program, впоследствии он не претерпел никаких изменений. Он содержит единственный метод Main, который запускает приложение и делает его главную форму (окно приложения) активной и видимой.

Файл Common содержит определения основных перечислений, структур и интерфейсов, которые используются в остальных частях программы.

Файл `GraphicsHandler` содержит определение единственного одноименного класса, предназначением которого является загрузка и хранение изображений шахматных фигур в формате `png` для последующего использования в других частях программы.

Файл `ChessBoard` содержит определение единственного одноименного класса, который, в свою очередь, является определением игрового поля в виде шахматной доски. Класс имеет несколько статических полей, недоступных извне (с модификатором `private`).

Файл `LegalMoveSet` содержит определение единственного одноименного статического класса, основной функцией которого является реализация механизма перемещения шахматных фигур по игровому полю. Класс содержит единственное статическое свойство `UseSimpleRules` типа `bool`. Это свойство доступно извне как для чтения, так и для записи. При установке значения `true` часть правил игры перестает учитываться при определении доступных ходов шахматных фигур. Так, слон получает возможность атаковать фигуры при перемещении на одну клетку из центральной зоны ходом ладьи. Кроме того, слон и ладья получают возможность перемещаться на неограниченное расстояние при условии, что они не пересекают центральную зону дважды.

Файл `AI` содержит определение единственного одноименного статического класса, основной функцией которого является выполнение операций, связанных с работой компьютерного противника. Данный класс реализует алгоритм `MiniMax` с альфа-бета-отсечением.

Файл `GameHandler` содержит определение единственного одноименного класса, основной функцией которого является выполнение некоторых внутренних игровых операций, не связанных напрямую с главной формой (окном) программы.



## **ЗАКЛЮЧЕНИЕ**

В результате данной бакалаврской работы были детально изучены средства и особенности языка C#, а также интерфейса программирования приложений Windows Forms. Была разработана полная версия игры-симулятора шахмат «Гала», включающая в себя весь запланированный функциональный набор, игровые режимы и компьютерного противника. Реализованы самостоятельные классы, которые с минимальными изменениями могут быть использованы при разработке иных приложений. Различные аспекты игры могут быть добавлены, улучшены и изменены. Это позволяет продолжить разработку нового функционала и дополнений к текущему проекту в будущем.

### **Основные источники информации:**

- Computer Chess Engines. [Электронный ресурс]. URL: <https://www.chess.com/article/view/computer-chess-engines> (Дата обращения 20.04.2021). Загл. с экр. Яз. Англ.
- How Stockfish Works: An Evaluation of the Databases Behind the Top Open-Source Chess Engine. [Электронный ресурс]. URL: <http://rin.io/chess-engine/> (Дата обращения 24.04.2021). Загл. с экр. Яз. Англ.
- Садлер М. – Game Changer AlphaZero’s Groundbreaking Chess Strategies and the Promise of AI. / М. Садлер, Н. Риган, Г. Каспаров – AMS, New in Chess, 2019. - 414 с.
- A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. [Электронный ресурс]. URL: <https://science.sciencemag.org/content/362/6419/1140> (Дата обращения 29.04.2021). Загл. с экр. Яз. Англ.
- Шилдт Г. – Полный справочник по C# / Г. Шилдт – М. Издательский дом «Вильямс», 2008. - 752 с.
- Климов А. П. – Советы программистам / А. П. Климов – СПб. БХБ-Петербург, 2008. - 544 с.
- Троелсен Э. – C# и платформа .NET. Библиотека программиста / Э. Троелсен – СПб. Питер, 2007. - 796 с.
- Троелсен Э. – Язык программирования C# 7 и платформы .NET и .NET Core. / Э. Троелсен, Ф. Джепикс - 8-е изд. – СПб. Диалектика, 2018. - 1328 с.