МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

на курсовую работу

по дисциплине «Алгоритмы и структуры данных»

Тема Шашки Артамонова

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Р.02069337. 22/2405 16 ТЗ-03

Листов <количество листов документа>

Руководитель разработки:

Шишкин Вадим

Викторинович

« » 2023 г.

Исполнитель:

студент гр. ИСТбд-21

Савельев Артём

Серегеевич

« » 2023 г.

2023

Содержание

Аннотация……...…………………………………………………………3

Техническое задание…………………………………………………...4

Пояснительная записка...……………………………………………...9

Руководство пользователя……………………………….…………...16

Текст программы…..…………………………………………………...22

**АННОТАЦИЯ**

Тема курсового проекта: Компьютерная логическая игра «Шашки Артамонова»

Исполнитель: студент гр. ИСТбд-21 Савельев Артём Сергеевич

Руководитель разработки: Шишкин Вадим Викторинович

Работа состоит из технического задания, пояснительной записки,

руководства программиста и текста программы.

В техническом задании описаны общие компьютерной логической игры «Шашки Артамонова», условия выигрыша и проигрыша, начальные позиции фигур. Приведены основания для разработки, функциональное назначение, основные требования к функциональным характеристикам, надежности, информационной и программной совместимости, хранению, транспортировке, программной документации.

В пояснительной записке указываются задачи, математические методы, архитектура и алгоритмы, тестирование и источники, использованные при разработке.

В руководстве программиста приводятся назначение и функции, выполняемые приложением, условия использования, характеристики приложения, особенности реализации приложения, обращение к программе, сообщения, выдаваемые по результатам контроля корректности ввода/вывода

Текст программы представляет собой полный код программы,

реализующий компьютерную игру эндшпиль компьютерную логическую игру «Шашки Артамонова».

Ключевые слова: компьютерная игра, шашки, Артамонов, python.

Всего листов 32

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на курсовую работу

по дисциплине «Алгоритмы и структуры данных»

Тема “Шашки Артамонова”

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Р.02069337. 22/2405 16 ТЗ-03

Листов 6

Исполнитель:

студент гр. ИСТбд-21

Савельев Артём Сергеевич

« » 2023 г.

2023

**Введение**

Компьютерная логическая игра «Шашки Артамонова» ведется на Шашечная доска состоит из 64 одинаковых квадратов, попеременно светлых (белых) и тёмных (светло-коричневых) полей.

**Правила:**

Игра ведется только по тёмным полям. Шашечная доска располагается между партнёрами таким образом, чтобы в нижнем углу слева от играющего находилось тёмное угловое поле. Каждый игрок может в свою очередь хода, либо выполнить ход от себя, любой шашкой, здесь все шашки общие, либо выполнить рубку по правилам русских шашек. Если простая шашка находится на одной диагонали рядом с шашкой соперника, за которой имеется свободное поле, она должна быть перенесена через эту шашку на свободное поле. Это называется взятие, а на сленге — «рубить». Шашка соперника в этом случае снимается с доски. Если дамка находится на одной диагонали рядом или на расстоянии от шашки соперника, за которой имеется одно или несколько свободных полей, она должна быть перенесена через эту шашку на любое свободное поле. Шашка соперника в этом случае снимается с доски. Взятие считается одним сыгранным ходом. Взятие должно быть четко обозначено и проводится в определенном порядке. Оно считается законченным после снятия шашки соперника с доски. Если в процессе взятия шашкой она вновь оказывается на одной диагонали рядом с другой шашкой соперника, за которой имеется свободное поле, она должна быть перенесена через вторую, третью и т.д. шашку. Если в процессе взятия дамкой она снова оказывается на одной диагонали рядом или на расстоянии от шашки соперника, за которой находится одно или несколько свободных полей, дамка должна быть обязательно перенесена через вторую, третью и т.д. шашку и занять любое свободное поле на той же диагонали за последней взятой шашкой. После завершения взятия (согласно двух предыдущих пунктов) взятые шашки соперника снимаются с доски в порядке их взятия. Это называется последовательным взятием. В процессе последовательного взятия запрещается переносить шашки через свои собственные. В процессе последовательного взятия разрешается проходить несколько раз через одно и то же поле, но запрещается переносить свою шашку через одну и ту же шашку соперника более одного раза. Взятие должно быть четко обозначено, шашка за шашкой, ставя берущую шашку на промежуточное, а затем и на конечное поле. Не обозначение взятия считается неспортивным и некорректным поведением. Последовательное взятие считается законченным, если игрок по окончании переноса шашки отнял от нее руку. Взятые шашки могут быть сняты с доски только после завершения последовательного взятия и в том же порядке, в каком проходило их взятие. Снятие взятых шашек с доски считается законченным, когда игрок снял последнюю взятую шашку. При возможности взятия по двум и более направлениям дамкой или шашкой выбор, вне зависимости от количества или качества снимаемых шашек (дамки или простой), предоставляется берущему. Если простая шашка при взятии достигает последнего (восьмого от себя) горизонтального ряда и если ей предоставляется возможность дальнейшего взятия шашек, то она обязана тем же ходом продолжать бой, но уже на правах дамки. Если же простая шашка достигает последнего горизонтального ряда без взятия и ей после этого предоставляется возможность боя, то она должна бить (если эта возможность сохранится) лишь следующим ходом на правах дамки.

Правила рубки дамкой также сохранены. Если в процессе игры шашка достигает одного из полей последнего, восьмого (считая от себя) горизонтального ряда, она превращается в дамку, получая новые права. На доске у каждой стороны одновременно может быть по несколько дамок. Дамки обозначаются сдвоенными шашками, т.е. поставленными одна на другую. Дамка, в отличие от простой, ходит на любое из свободных полей по диагонали в любом направлении (как вперед, так и назад), но становиться может, как и простая, лишь на не занятые другими шашками поля, причем через свои шашки она перескакивать не может.

**Нюансы:**

Есть только три нюанса. Нельзя ходить той шашкой, которой только выполнил ход противник. И если шашка в процессе рубки достигла последней горизонтали (крайней от игрока), то она не продолжает рубку, даже если это еще возможно, а ход передается противнику, сама шашка превращается в дамку. Третий нюанс заключается в необязательности рубки. По причине того, что наши ходы ограничены и нам нужно сделать так чтобы за одну рубку съелось максимальное количество шашек.

**Цель:**

Cрубить как можно больше шашек. **Игра завершается** если в течении длительной серии ходов, ни один игрок не смог выполнить рубку. Серия в 15 ходов. После объявления о завершении игры, считаются срубленные шашки, кто срубил больше, тот и победил**.**

**Функциональные возможности:**

В приложении должно быть реализовано меню с настройками. В самой игре у игрока должна быть возможность выбирать шашку и ей ходить. При окончании игры, высвечивается результат и счёт.

**1. Основания для разработки**

Учебный план направления 09.03.02 «Информационные системы и технологии» и распоряжение по факультету № 12-р.

**2. Требования к программе или программному изделию**

**2.1. Функциональное назначение**

Требуется разработать однопользовательское десктопное приложение по компьютерной логической игре «Шашки Артамонова» с графическим интерфейсом в среде Windows.

**2.2 Требования к функциональным характеристикам**

2.2.1 Требования к структуре приложения

Приложение должно быть разработано в виде одного модуля с дополнительными информационными файлами при необходимости.

2.2.2 Требования к составу функций приложения

В приложении должны быть реализованы в графическом режиме следующие основные функции:

- регистрация/авторизация пользователя;

- отрисовка игрового поля;

- взаимодействие с пользователем;

- интерактивные прием, проверка правильности и отрисовка хода пользователя;

- проверка окончания игры;

- вычисление, проверка правильности и отрисовка хода компьютер;

- информирование пользователя об окончании игры и победителе.

2.2.3 Требования к организации информационного обеспечения, входных и выходных данных

В приложении должен быть реализован графический интерфейс взаимодействия с пользователем. Изображения шашек могут храниться в отдельных графических файлах. Логин и пароль пользователя должны вводиться с клавиатуры. Логины и пароли зарегистрированных пользователей должны храниться в отдельном файле или базе данных в зашифрованном виде. Пояснительные информационные сообщения для пользователя должны выводиться внизу игрового поля по ходу игры.

**2.3 Требования к надёжности**

Приложение должно чётко выполнять команды игрока и в случае ошибки сообщить игроку о ней и попытается исправить.

**2.4 Требования к информационной и программной совместимости**

Версия операционной системы Windows 7,8,8.1,10,11. Используемые библиотеки: tkinter, itertool, random. Язык Python. Среда разработки PyCharm.

**2.5 Требования к маркировке и упаковке**

Определяются заданием на курсовую работу.

**2.6 Требования к транспортированию и хранению**

2.6.1 Условия транспортирования

Требования к условиям транспортирования не предъявляются.

2.6 2 Условия хранения

Требования к условиям хранения не предъявляются.

2.6 3 Сроки хранения

Срок хранения – до июля 2023 года.

**3. Требования к программной документации**

Определяются заданием на курсовую работу.

**4. Стадии и этапы разработки**

Определяются заданием на курсовую работу.

**5. Порядок контроля и приёмки**

Определяются заданием на курсовую работу.

**ШАБЛОН ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.**

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Измерительно-вычислительные комплексы»

**Курсовая работа**

**По дисциплине «Алгоритмы и структуры данных»**

**Тема “Шашки Артамонова**

**Пояснительная записка**

Р.02069337. 22/2405 16 ТЗ-03

Листов 7

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Исполнитель:

студент гр. ИСТбд-21

Савельев Артём Сергеевич

« » 2023 г.

2023

**Введение**

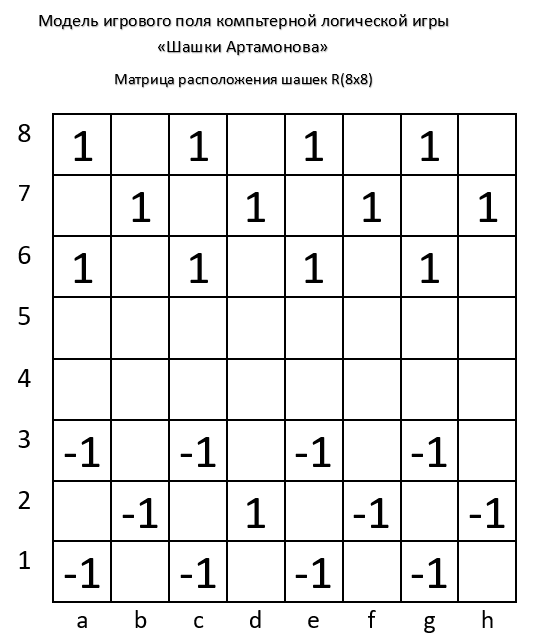
Одноцветные шашки, Компьютерная логическая игра «Шашки Артамонова». В приложении есть меню регистрации, главное меню и сама игра.

**1. Проектная часть**

**1.1 Постановка задачи на разработку приложения**

Определяется заданием на курсовую работу. Детализируется в разработанном техническом задании (приложение 1).

**1.2 Математические методы**

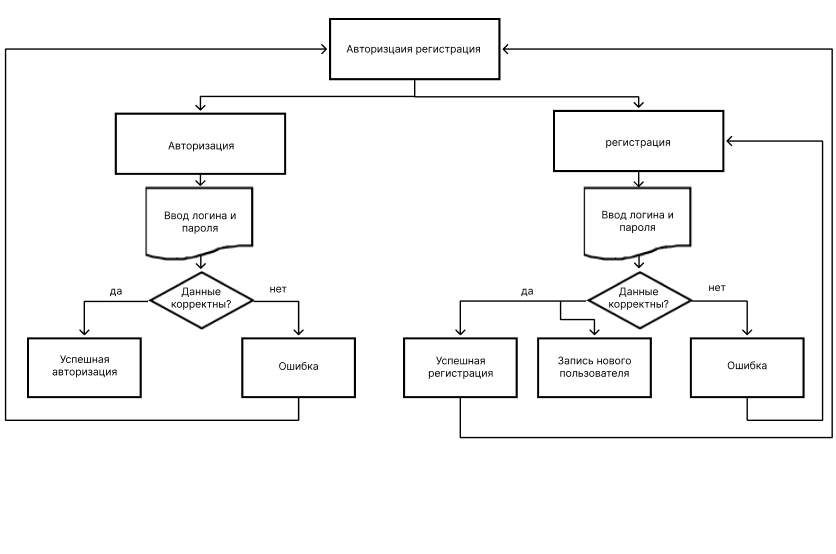
****

**1.3 Архитектура и алгоритмы**

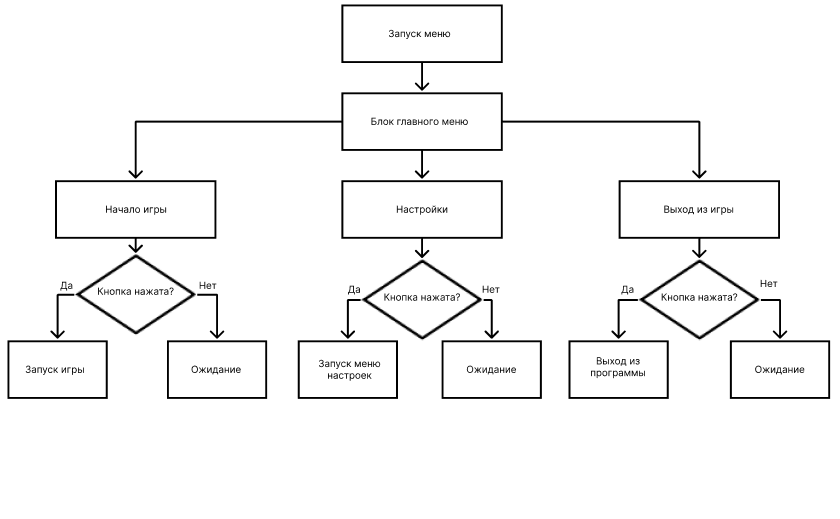
1.3.1. Архитектура



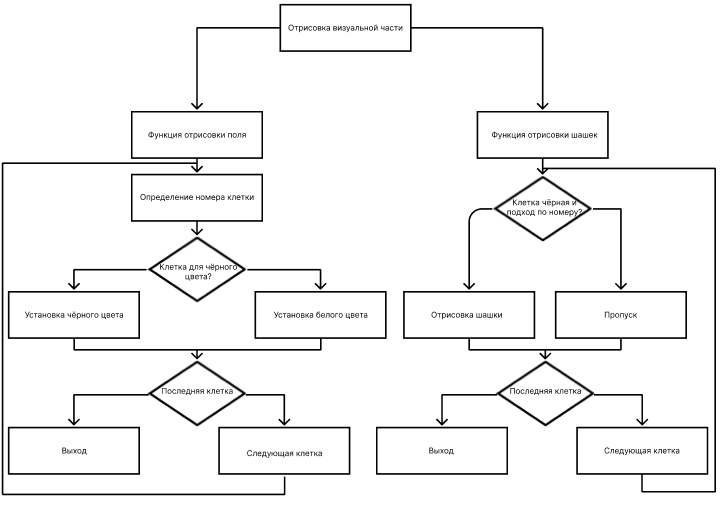
1.3.2. Алгоритм авторизации/регистрации



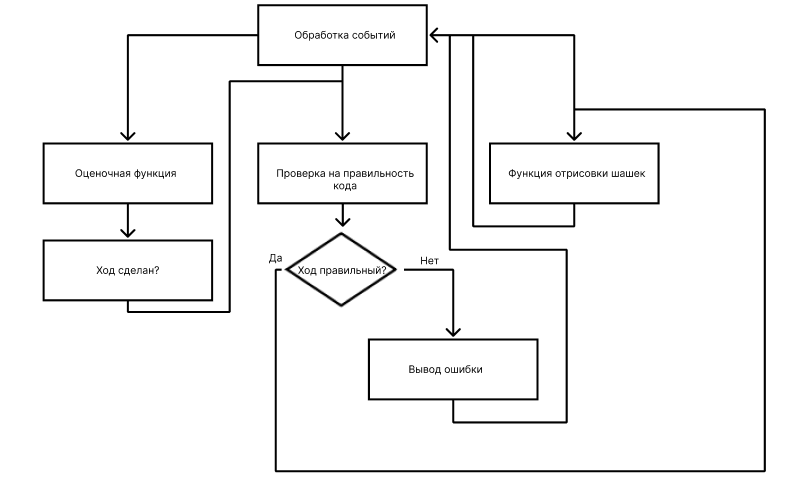
1.3.2 Алгоритм запуска меню



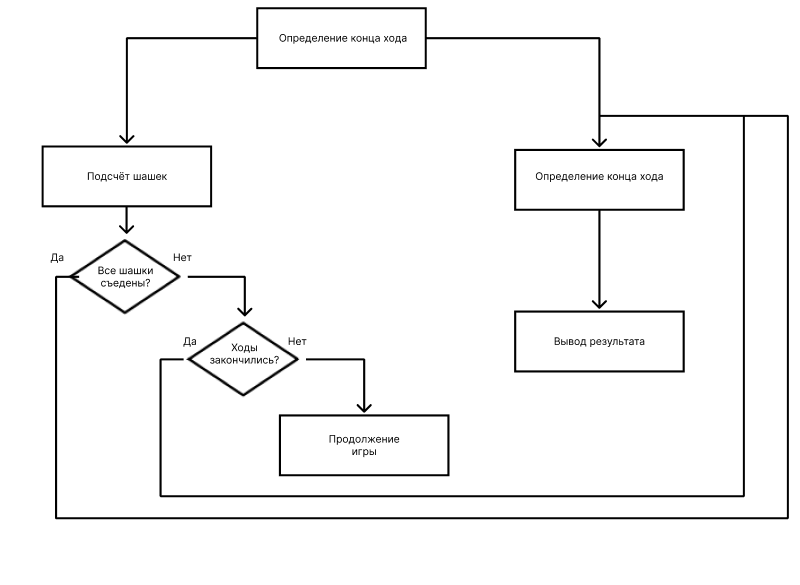
1.3.2 Алгоритм отрисовки



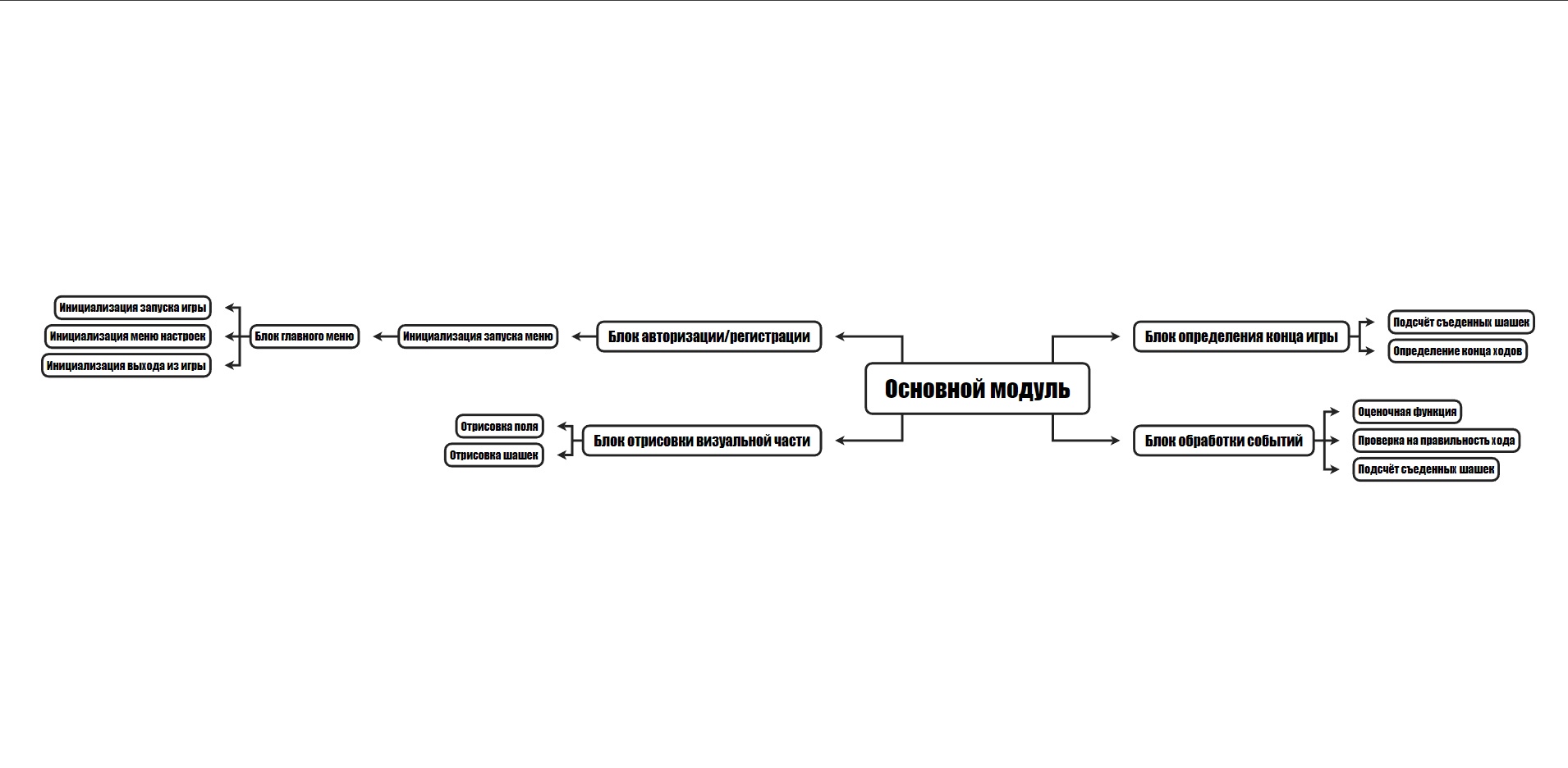
1.3.2 Алгоритм обработки событий



1.3.2 Алгоритм определение конца хода



**1.4 Тестирование**



1.4.1 Описание отчета о тестировании

В данном отчете представлены результаты тестирования программы на основе статического тестирования документации и программного кода. Описаны проведенные тесты, их результаты и обнаруженные дефекты.

1.4.2 Цель тестирования

Целью тестирования является проверка соответствия ПО предъявляемым требованиям, а также выявление возможных багов. По результатам тестирования следует исправление выявленных багов.

1.4.3 Методика тестирования

Тестирование проводилось с использованием следующих методов:

– Статическое тестирование: анализ и проверка кода без его запуска, выявление ошибок в технической документации.

– Ручное тестирование: запуск пользовательских сценариев программы с различными входными данными и проверка корректности полученных результатов.

1.4.4 Проведенные тесты

В ходе тестирования были проведены следующие тесты: 4.1) Статическое тестирование

Количество обнаруженных и исправленных ошибок в документации: 2.

Количество обнаруженных и исправленных ошибок в программном коде: 5.

4.2) Ручное тестирование.

Написаны и проведены следующие тест-кейсы и чек-листы: ТК1. Отработка авторизации. Предварительные шаги:

Зарегистрироваться с логином: TestTest и паролем 12345678.

Шаги:

1. Запустить приложение.

2. В окне регистрации, в поле «логин» ввести TestTest, а в поле «пароль» − 12345678.

3. Нажать кнопку «Авторизоваться».

Ожидаемый результат: Пользователь начнет игру.

Фактический результат: Пользователь начал новую игру.

ТК2. Прохождение регистрации односимвольными логином и паролем.

Шаги:

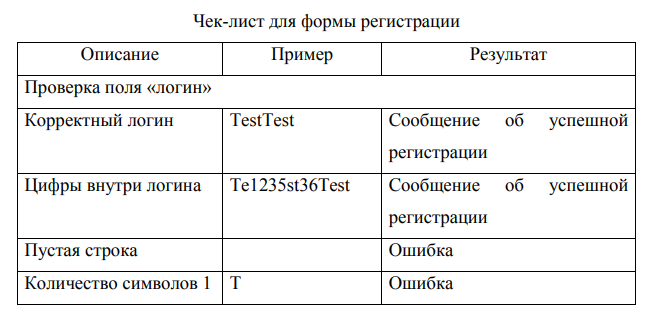
1. Запустить приложение.

2. В окне регистрации, в поле «логин» ввести 1, а в поле «пароль» − 1.

3. Нажать на кнопку «Зарегистрироваться/ начать игру».

Ожидаемый результат: Пользователь получит сообщение об ошибке.

Фактический результат: Пользователь зарегистрировался.



1.4.5 Вывод

На основе проведенных тестов сделаны следующие выводы:

– Программа успешно прошла все тесты и работает корректно.

– Обнаружены и исправлены следующие дефекты: некорректное название переменных, неправильное название игры, неправильное определение хода и дальнейший расклад игры

– Рекомендации по дальнейшему развитию программы: звукового сопровождения, таблицы лидеров, улучшение условий окончания игры

**2. Источники, использованные при разработке**

1. Шишкин, В.В. Разработка логических компьютерных игр с

графическим интерфейсом в среде питон / В.В. Шишкин, Д.С.

Афонин. - Ульяновск: УлГТУ, 2023. – 89 с. – URL: chrome-

extension://efaidnbmnnnibpcajpcglclefindmkaj/http://lib.ulstu.ru/venec/di

sk/2023/112.pdf. – Текст: электронный

1. Руководство по Tkinter . – URL: <https://metanit.com/python/tkinter/>

**Приложение 5.**

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Измерительно-вычислительные комплексы»

**Курсовая работа**

**По дисциплине «Алгоритмы и структуры данных»**

**Тема “Шашки Артамонова**

**Руководство программиста**

Р.02069337. 22/2405 16 ТЗ-03

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Листов

Исполнитель:

студент гр. ИСТбд-21

Савельев Артём Сергеевич

« » 2023 г.

2023

**1. Назначение и условия применения программы**

**1.1 Назначение и функции, выполняемые приложением**

Однопользовательское десктопное приложение по игре

с графическим интерфейсом в среде

Windows.

**1.2 Условия, необходимые для использования приложения**

Версия операционной системы: Windows 7, 8, 8.1, 10, 11

Язык программирования: Python

Среда разработки: PyCharm Community Edition 2023.1.1

Используемые библиотеки: графическая библиотека Tk

**2. Характеристики программы**

**2.1 Характеристики приложения**

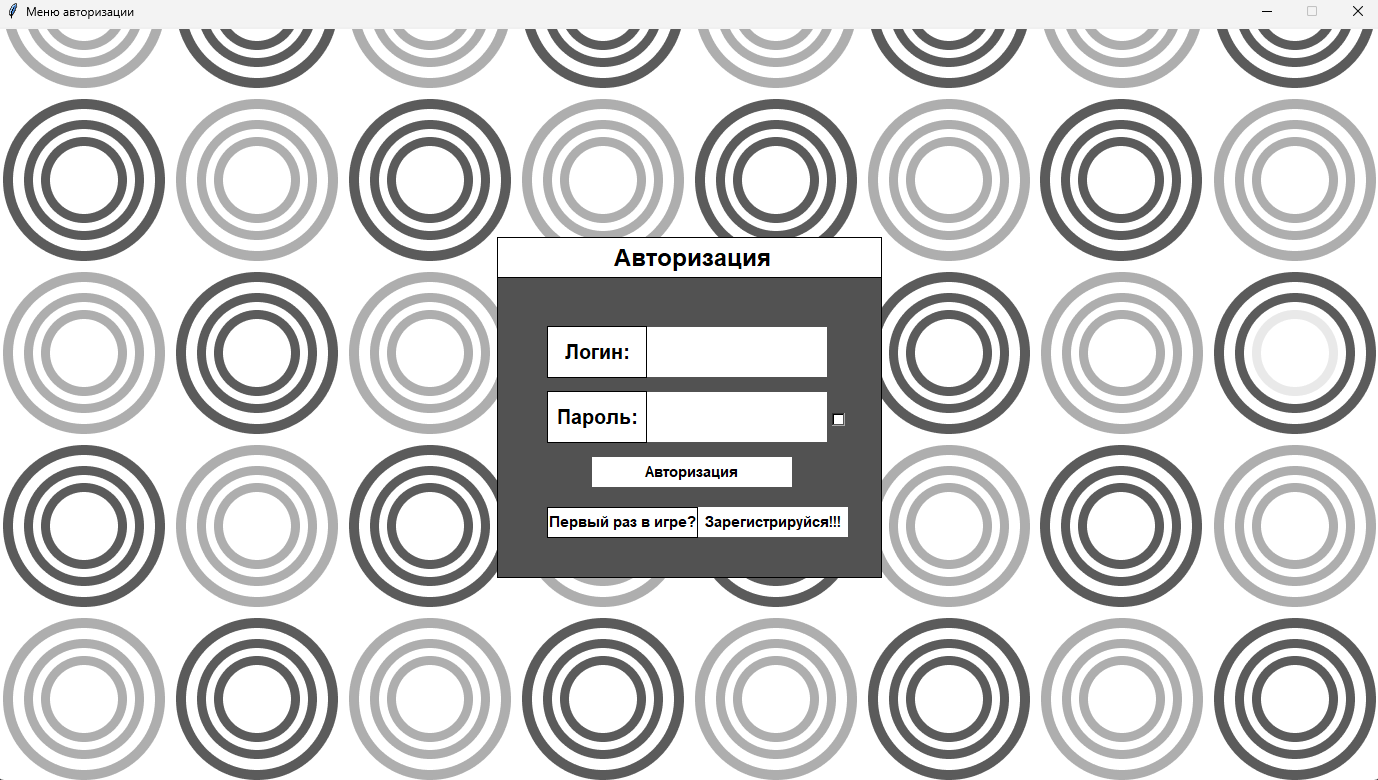
Указывается количество значимых (т.е. выполняющих какие-то действия) строк программного кода. Количество структур данных, алгоритмов.

Описываются используемые библиотеки.

1. Окно регистрации и авторизации

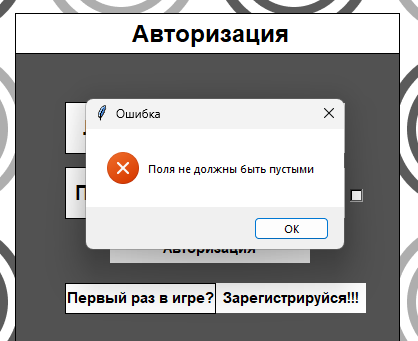
1.1. Для начала игры необходимо зарегистрироваться и

авторизоваться

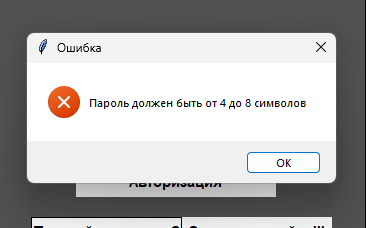


1.2. Если никаких данных введено не было или были введены

пробелы, возникнет сообщение об ошибке:

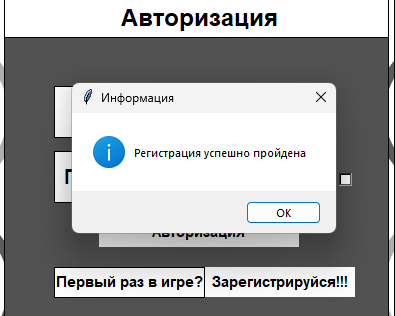


1.3. Если введенные при регистрации пароль и логин меньше четырёх символов и больше восьми, возникнет сообщение об ошибке:



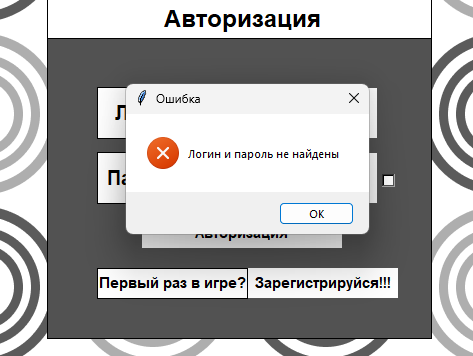
1.4. При введении корректных данных возникнет сообщение об

успешной регистрации



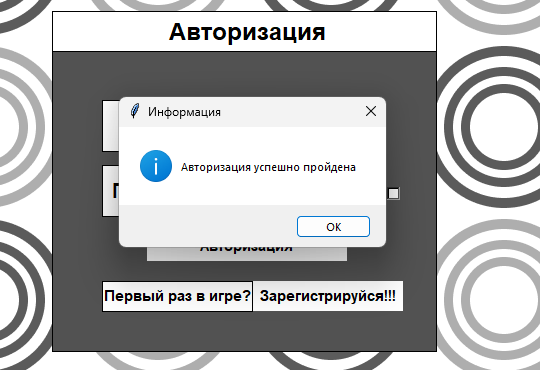
1.5. При вводе неправильного логина или пароля возникнет

сообщение об ошибке:



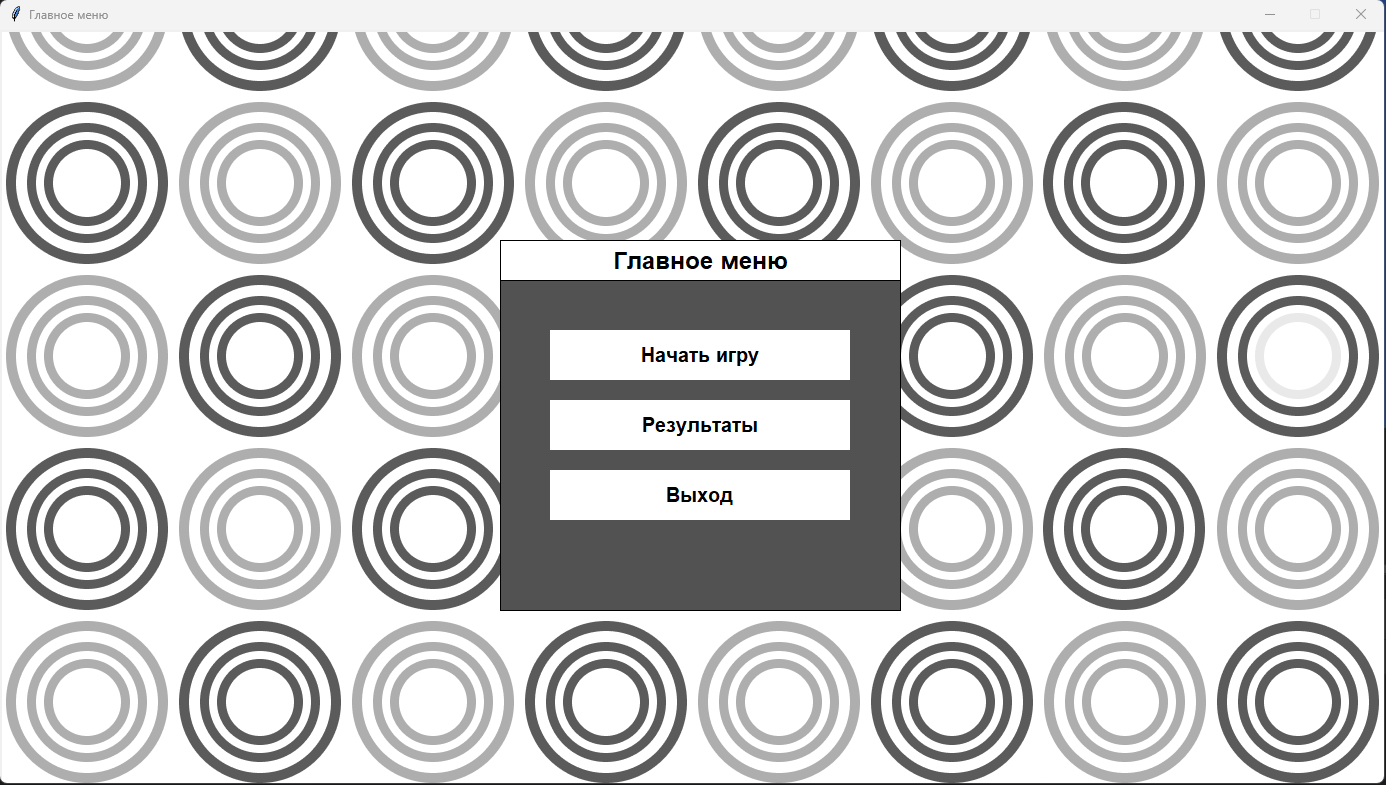
1.6. При вводе корректных данных появится новое окно с

сообщением об успешной авторизации и кнопкой начала игры:



2. Главное меню

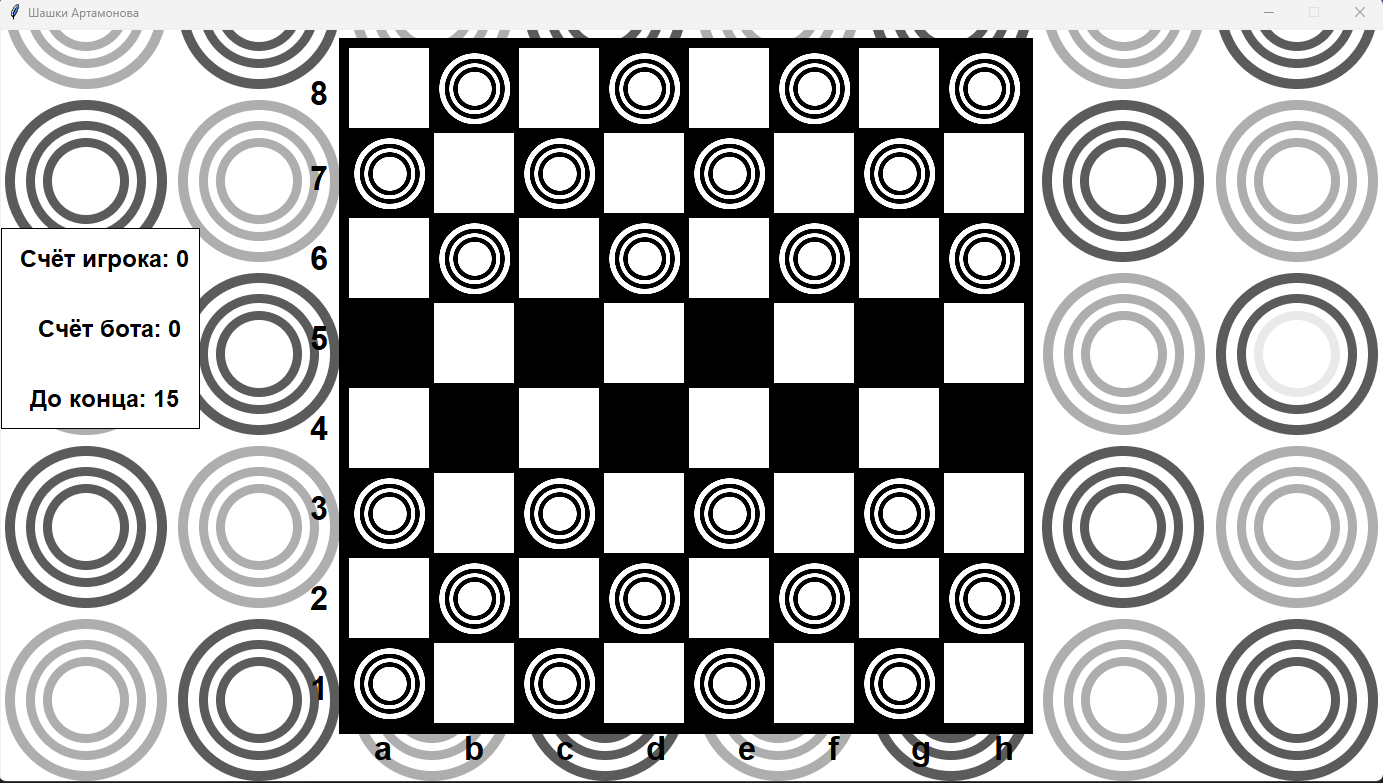
2.1. После авторизации появляется главное меню

****

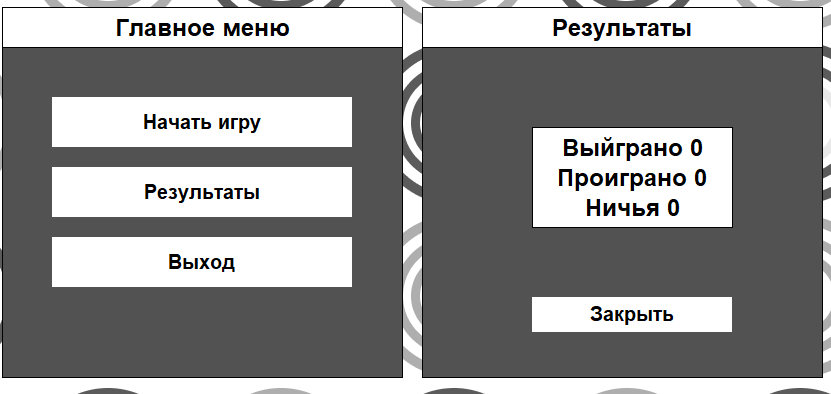
2.2. После нажатия на кнопку начала игры запускается отрисовка

игрового поля и шахматных фигур. Шахматные фигуры

расставляются случайным образом:

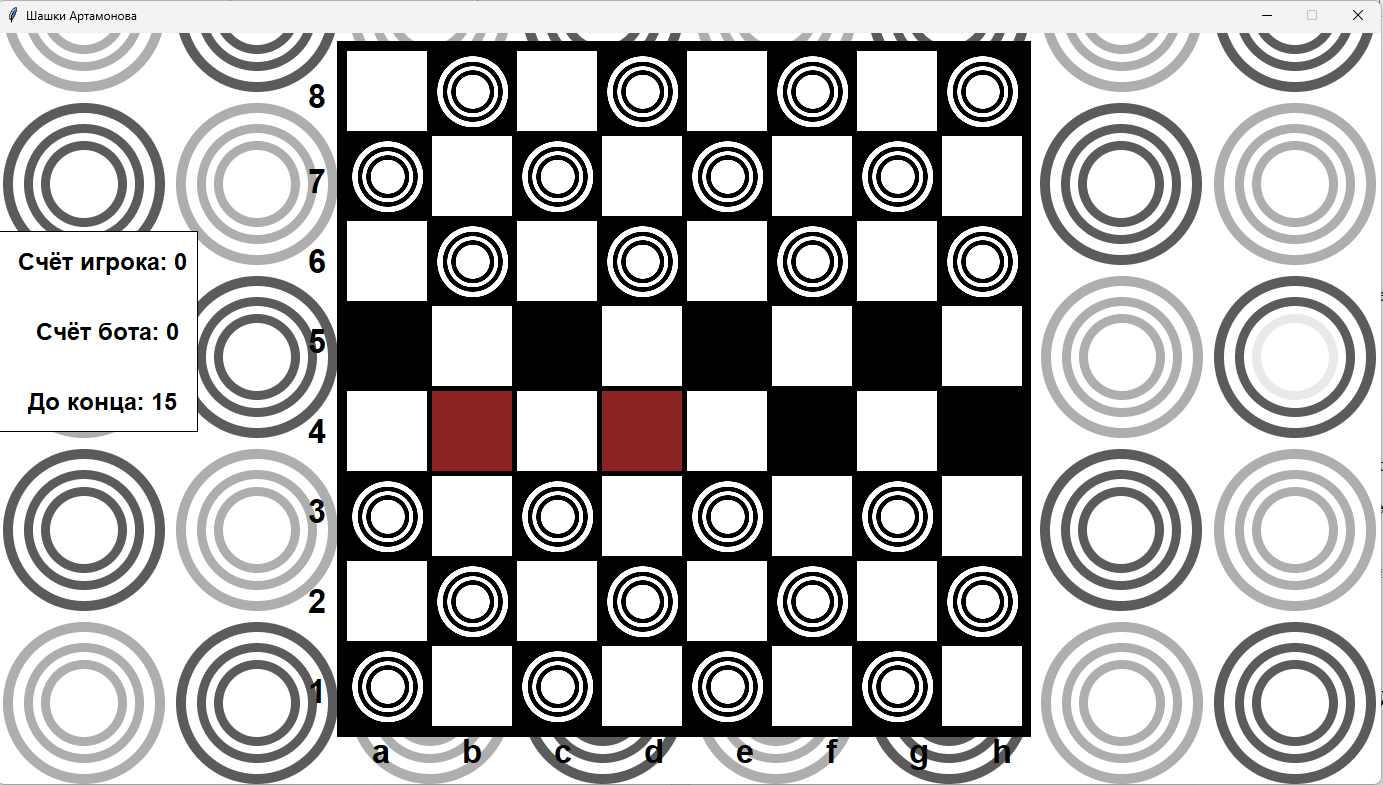


* 1. После нажатия на кнопку резултат запускается окно резултьтатов где показывается сколько раз игрок выиграл, проиграл и сыграл в ничью

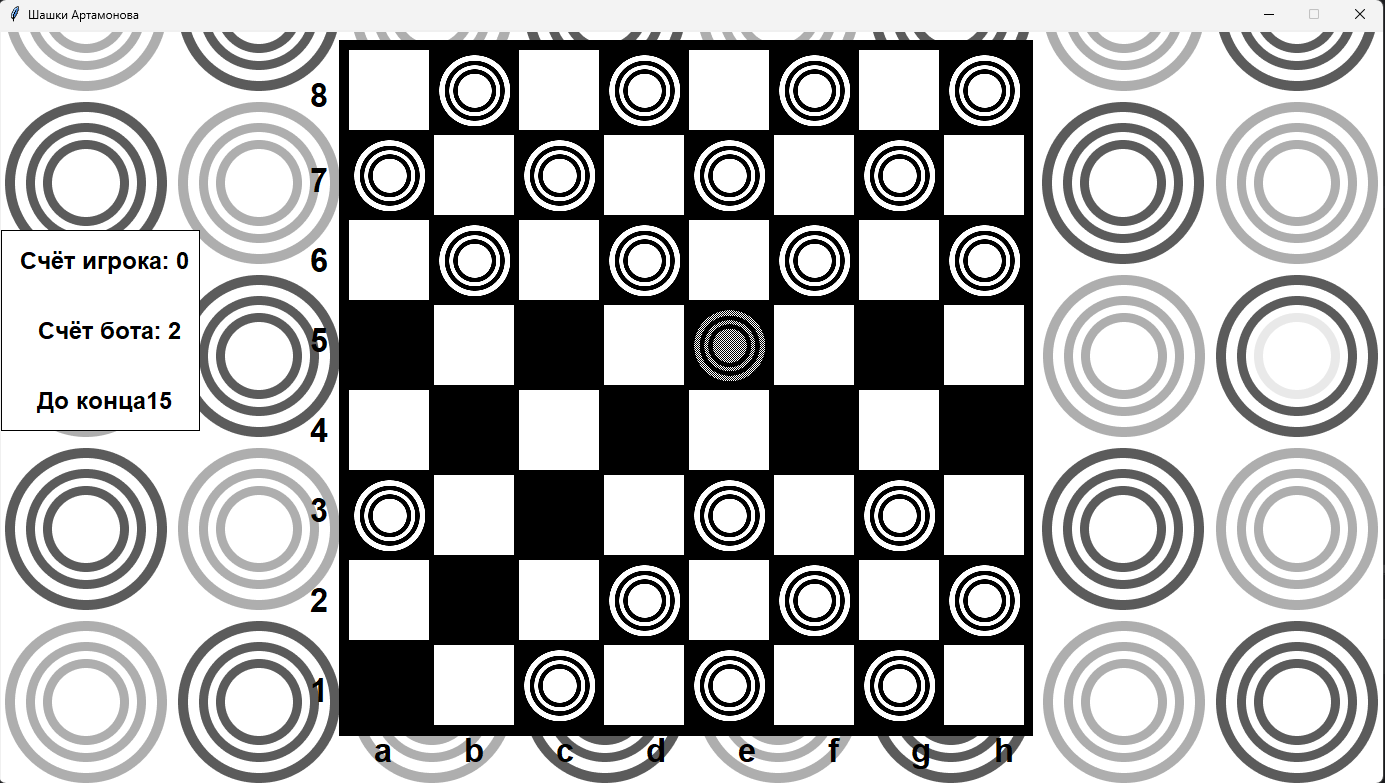


1. Игра

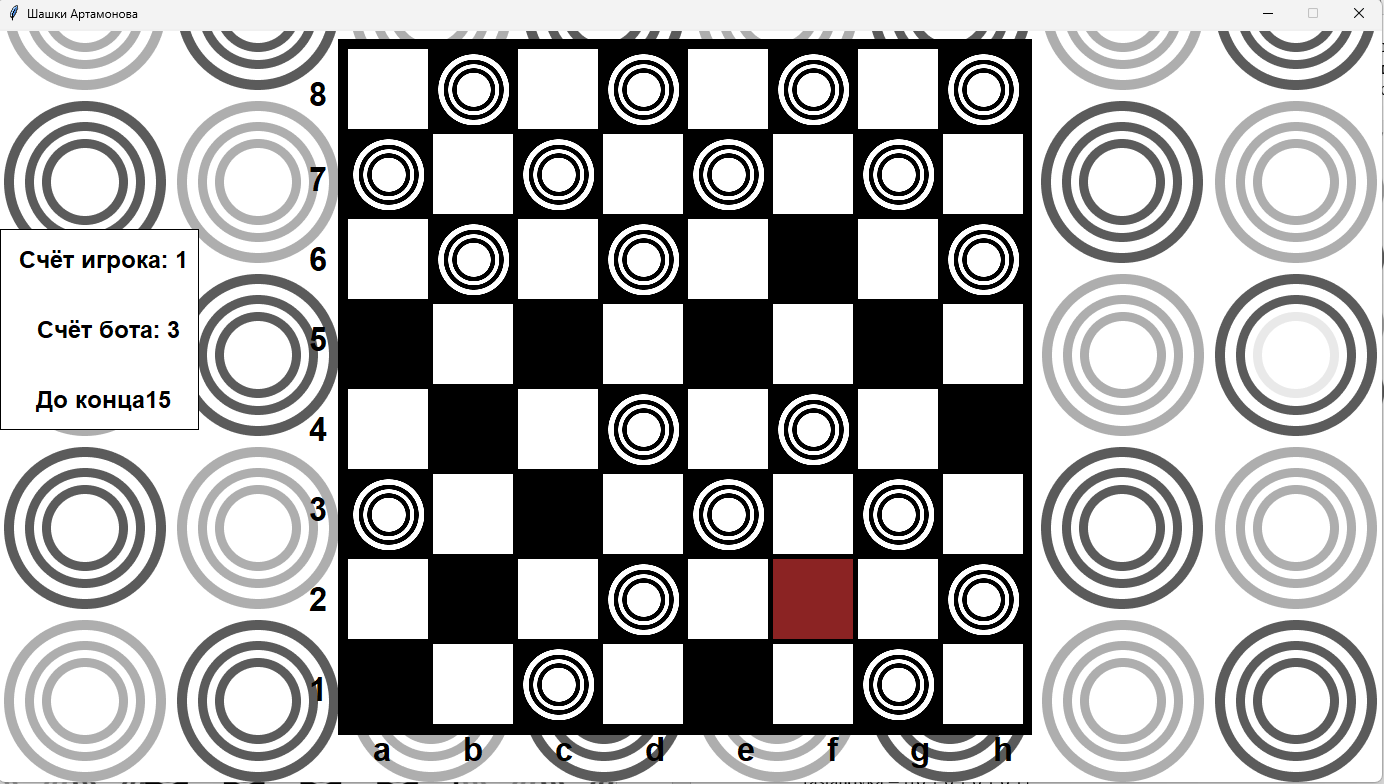
3.1. При нажатии на шашку показываются доступные ходы, при повторном нажатии на ту же шашку, её выбор отменятся.



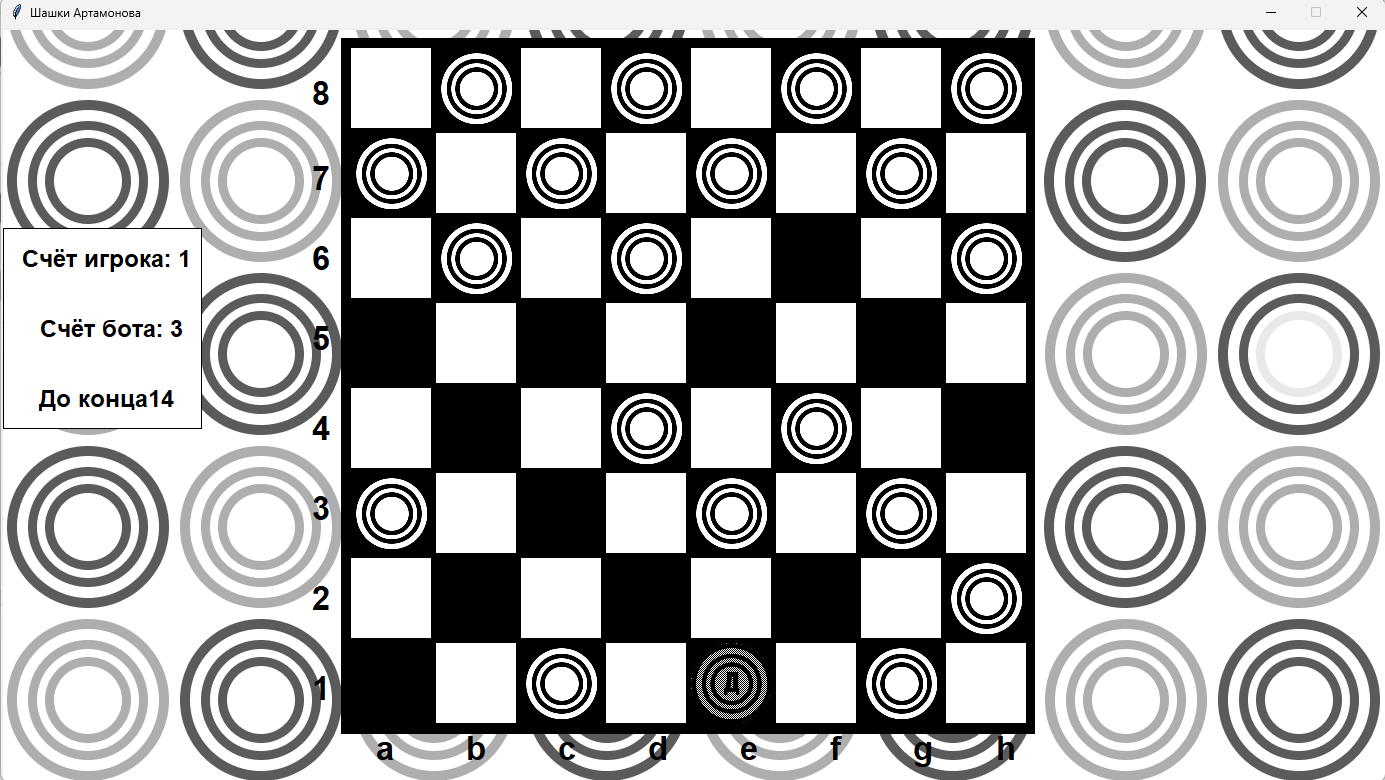
3.2. При нажатии на клетку с красной подсветкой, шашка передвигается и происходит ход бота. Выбранная ранее шашка выключается и не может быть использована. Если на пути хода была шашка, то она убирается с поля и прибавляется к результату.



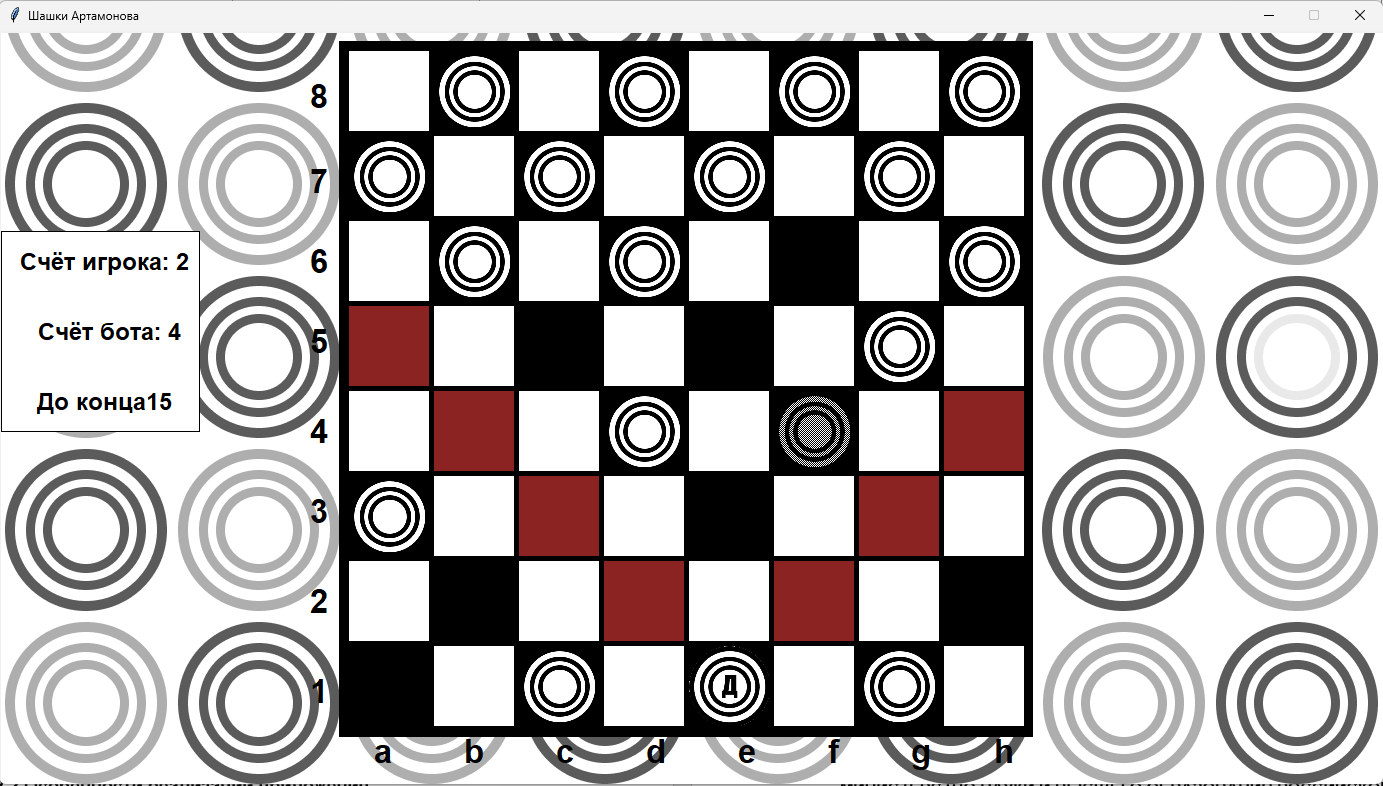
* 1. При съедении шашки, если есть ещё один ход, при котором можно съесть, ход продолжается и есть выбор есть дальше. Если игрок не хочет, то при повторном нажатии на выбранную ранее шашку, ход заканчивается.



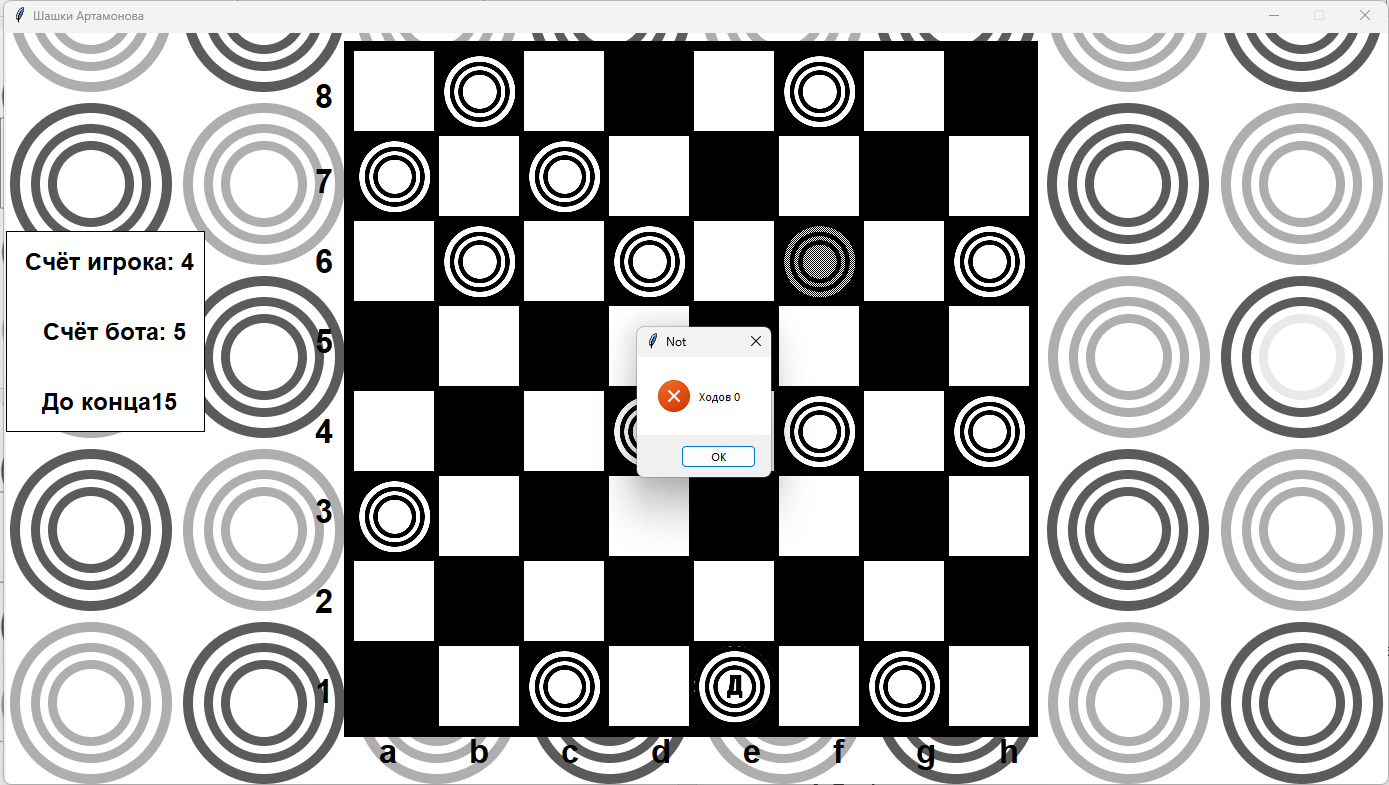
* 1. При достижении шашки другого края поля, она превращается в дамку.

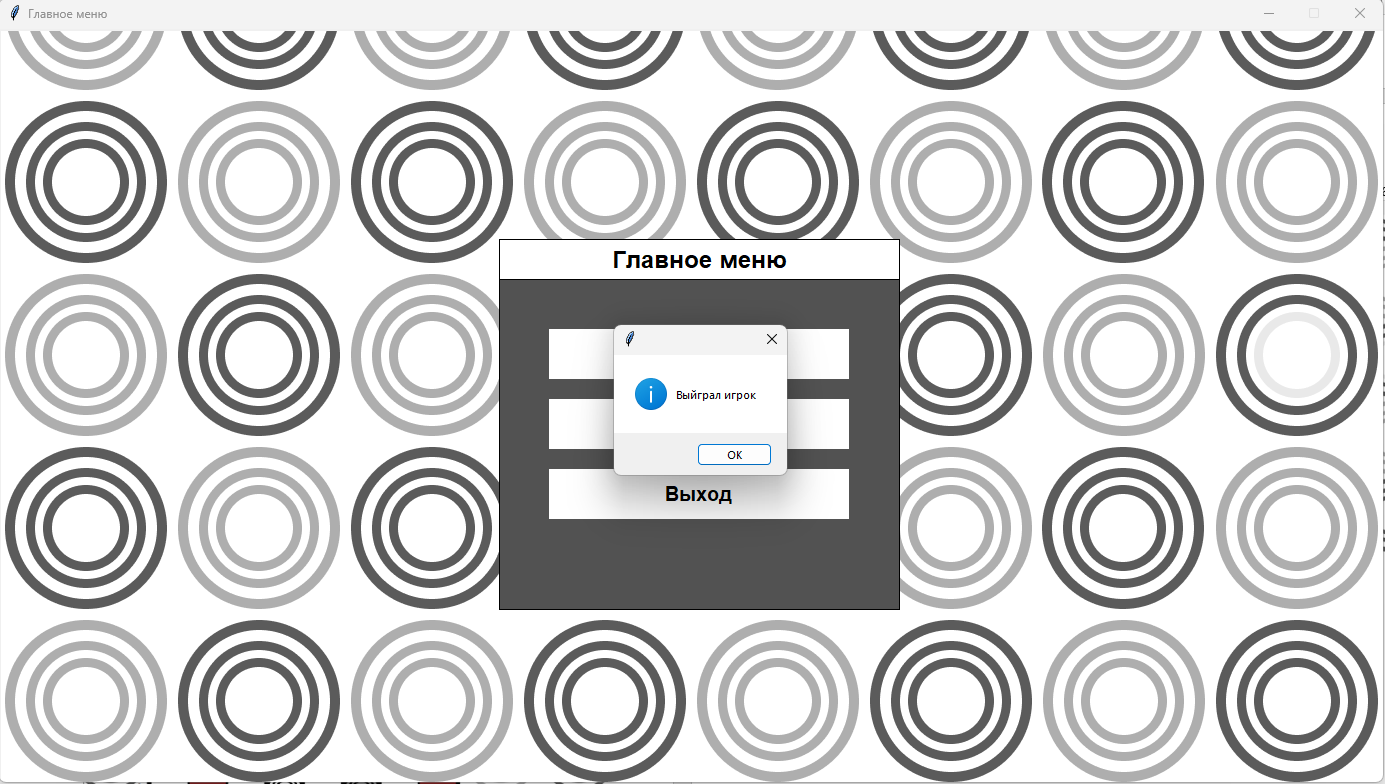


* 1. При выборе дамки, подсвечиваются ходы по диагонали.



* 1. При выборе шашки у которой не ходов, появляется ошибка.

 3.7. При достижении ситуации на поле, при которой смысла ходов нет или ходов нет, то игра заканчивается. После этого подсчитываются очки и выводится результат. Так же открывается главное меню.



Средства контроля ввода/вывода:

* Проверка на пустое поле
* Проверка на уже существующие логин и пароль
* Проверка на длину логина и пароля
* Проверка на авторизацию

**2.2 Особенности реализации приложения**

Массив поля:

rastanovka = [[0,1,0,1,0,1,0,1],  
 [1,0,1,0,1,0,1,0],  
 [0,1,0,1,0,1,0,1],  
 [0,0,0,0,0,0,0,0],  
 [0,0,0,0,0,0,0,0],  
 [-1,0,-1,0,-1,0,-1,0],  
 [0,-1,0,-1,0,-1,0,-1],  
 [-1,0,-1,0,-1,0,-1,0]]

Массив веса:

ves = [[1,2,3,4,5,6,7,8],  
 [9,10,11,12,13,14,15,16],  
 [17,18,19,20,21,22,23,24],  
 [25,26,27,28,29,30,31,32],  
 [33,34,35,36,37,38,39,40],  
 [41,42,43,44,45,46,47,48],  
 [49,50,51,52,53,54,55,56],  
 [57,58,59,60,61,62,63,64]]

**3. Обращение к программе**

Tkinter – это кросс-платформенная событийно-ориентированная графическая библиотека Python, предназначенная для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя (GUI).

Модуль tkinter.messagebox, предоставляющий доступ к готовым диалоговым окнам.

**4. Сообщения**

"Ошибка", "Поля не должны быть пустыми" при проверке на пустое

поле;

"Ошибка", "Логин и пароль уже существуют" при проверке на уже

существующие логин и пароль;

"Ошибка", "Пароль должен быть от 4 до 8" при регистрации нового пароля если он не подходит под условия.

"Информация", "Регистрация успешно пройдена” при успешной регистрации;

"Информация", "Авторизация успешно пройдена” при успешной авторизации;

"Ошибка", "Логин и пароль не найдены" при проверке на

авторизацию;

"Ошибка", "Запрещено" при выборе неправильного хода;

“Ошибка”, "Ходов 0" при ситуации когда ходов для выбранной шашки нет;

"Информация", "Выиграл игрок” при большем количестве очков у игрока;

"Информация", "Выиграл бот” при большем количестве очков у бота;

"Информация", "Ничья” при равном количестве очков у обоих;

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Измерительно-вычислительные комплексы»

**Курсовая работа**

**По дисциплине «Алгоритмы и структуры данных»**

**Тема “Шашки Артамонова**

**Текст программы**

Р.02069337. 22/2405 16 ТЗ-03

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Листов 10

Исполнитель:

студент гр. ИСТбд-21

Савельев Артём Сергеевич

« » 2023 г.

2023

from tkinter import \*

from tkinter.messagebox import showerror, showwarning, showinfo

root= Tk()

root.title("Меню авторизации")

w = root.winfo\_screenwidth();h = root.winfo\_screenheight()

wh = (w // 2) - 700;hh = (h // 2) - 400

root.geometry(f"1384x753+{wh}+{hh}")

root.resizable(False, False)

canvas = Canvas(root, bg='white', width=1384, height=753)

canvas.place(x=0, y=0)

img = PhotoImage(file='background.png')

enabled = IntVar()

x0, y0 = 50, 50

x1, y1 = 150, 150

x\_pred = 77;y\_pred = 77

log\_registr = StringVar();passwr\_register = StringVar()

login = StringVar();passw = StringVar()

ngo = 1

score\_igrok = 0;score\_bot = 0;chet\_do\_konca = 15

hodim = -1

i\_last\_last = 0;x\_last\_last = 0;y\_last\_last = 0

check\_dal = 0;check\_bliz = 0;check\_hod\_mat = 0;ch = 0

global rastanovka

rastanovka = []

global mat\_knopok,ves

mat\_knopok = []

ves = [[1,2,3,4,5,6,7,8],

[9,10,11,12,13,14,15,16],

[17,18,19,20,21,22,23,24],

[25,26,27,28,29,30,31,32],

[33,34,35,36,37,38,39,40],

[41,42,43,44,45,46,47,48],

[49,50,51,52,53,54,55,56],

[57,58,59,60,61,62,63,64]]

suma\_dal = 0;suma\_bliz = 0;poz\_bot\_x = 0;poz\_bot\_y = 0

max\_poz = 0;win\_igr = 0;win\_bot = 0;nich = 0

def proverka\_na\_conec(a, b):

global win\_igr,win\_bot,nich,hodim,score\_bot,score\_igrok,chet\_do\_konca

suma= 0

chek\_tup= 0

kol\_mat = 0

for i in range(8):

for j in range(8):

if rastanovka[i][j] == 1 or rastanovka[i][j] == -1:

suma += 1

if i == 0:

chek\_tup += 1

else:

chek\_tup= 0

if (suma == 1) or (suma == 0) or(suma > 0 and chek\_tup >0 and chek\_tup<5) or chet\_do\_konca == 0:

if a > b:

igr.destroy()

glavmenu.deiconify()

showinfo(title="", message="Выйграл игрок")

win\_igr +=1

elif a < b:

igr.destroy()

glavmenu.deiconify()

showinfo(title="", message="Выйграл бот")

win\_bot += 1

else:

igr.destroy()

glavmenu.deiconify()

showinfo(title="", message="Ничья")

nich +=1

hodim = -1

score\_bot = 0

score\_igrok = 0

chet\_do\_konca = 15

def smena\_hoda():

#-------------------------------смена номера хода-------------------------------

global hodim

if hodim == 1:

hodim = -1

else:

hodim = 1

def bot():

#-------------------------------бот-------------------------------

global return\_check

for io in range(8):

for jo in range(8):

if mat\_knopok[io][jo]['state'] == NORMAL and mat\_knopok[io][jo]['image'] != '' and mat\_knopok[io][jo]['bg'] != 'white':

if return\_check == 1:

return\_check = 0

break

elif return\_check == 0:

pokaz\_dostup\_hodov(io, jo, 0, 1, rastanovka[io][jo])

if return\_check == 1:

break

click\_check(0, poz\_bot\_x, poz\_bot\_y, 0);click\_check(0, poz\_bot\_x\_next, poz\_bot\_y\_next, 1)

def pokaz\_dostup\_hodov(x, y, clear, bot, k):

global ch,suma\_bliz,suma\_dal,poz\_bot\_x\_next ,poz\_bot\_y\_next ,max\_poz,poz\_bot\_x,poz\_bot\_y,fash,score,suma\_hod\_mat,return\_check

fash = 0;score = 0

chek\_M\_1 = 0;chek\_M\_2 = 0;chek\_M\_3 = 0;chek\_M\_4 = 0

if clear == 1:

max\_poz = 0

#-------------------------------очистка клеток от красного у шашки-------------------------------

if k == -1 or k == 1:

for i in range(((x + x\_last) // 2) - 1, ((x + x\_last) // 2) + 2):

for j in range(((y + y\_last) // 2) - 1, ((y + y\_last) // 2) + 2):

if i >= 0 and j >= 0 and i<8 and j<8 and i != x and j !=y and mat\_knopok[i][j]['bg'] == '#8B2323':

mat\_knopok[i][j]['bg'] = 'black'

elif i >= 0 and j>=0 and i<8 and j<8 and i != x and j !=y and mat\_knopok[i][j]['image'] != '' and mat\_knopok[i][j]['bg'] == 'black':

if (i + (i - x) >= 0) and (j + (j - y) >= 0) and (i + (i - x)) < 8 and (j + (j - y)) < 8 and i != x and j != y and mat\_knopok[i + (i - x)][j + (j - y)]['bg'] == '#8B2323':

mat\_knopok[i + (i - x)][j + (j - y)]['bg'] = 'black'

elif k == 2:

#-------------------------------очистка клеток от красного у дамки + удаление съеденых-------------------------------

for i in range(7, 0,-1):

if x + i >= 0 and y + i >= 0 and x + i < 8 and y + i < 8:

if mat\_knopok[x + i][y + i]['image'] == str(shash\_M) and chek\_M\_1 == 0:

chek\_M\_1 = 1

elif mat\_knopok[x + i][y + i]['image'] == str(shash\_M) and chek\_M\_1 == 1:

mat\_knopok[x + i][y + i]['image'] = ''

rastanovka[x - i][y + i] = 0

score += 1

if (mat\_knopok[x + i][y + i]['image'] == str(shash)) and chek\_M\_1 == 1:

mat\_knopok[x + i][y + i]['image'] = ''

rastanovka[x + i][y + i] = 0

score += 1

mat\_knopok[x + i][y + i]['bg'] = 'black'

if x - i >= 0 and y + i >= 0 and x - i < 8 and y + i < 8:

if mat\_knopok[x - i][y + i]['image'] == str(shash\_M) and chek\_M\_2 == 0:

chek\_M\_2 = 1

elif mat\_knopok[x - i][y + i]['image'] == str(shash\_M) and chek\_M\_2 == 1:

mat\_knopok[x - i][y + i]['image'] = ''

rastanovka[x - i][y + i] = 0

score += 1

if (mat\_knopok[x - i][y + i]['image'] == str(shash)) and chek\_M\_2 == 1:

mat\_knopok[x - i][y + i]['image'] = ''

rastanovka[x - i][y + i] = 0

score += 1

mat\_knopok[x - i][y + i]['bg'] = 'black'

if x + i >= 0 and y - i >= 0 and x + i < 8 and y - i < 8:

if mat\_knopok[x + i][y - i]['image'] == str(shash\_M) and chek\_M\_3 == 0:

chek\_M\_3 = 1

elif mat\_knopok[x + i][y - i]['image'] == str(shash\_M) and chek\_M\_3 == 1:

mat\_knopok[x + i][y - i]['image'] = ''

rastanovka[x + i][y - i] = 0

score += 1

if (mat\_knopok[x + i][y - i]['image'] == str(shash)) and chek\_M\_3 == 1:

mat\_knopok[x + i][y - i]['image'] = ''

rastanovka[x + i][y - i] = 0

score += 1

mat\_knopok[x + i][y - i]['bg'] = 'black'

if x - i >= 0 and y - i >= 0 and x - i < 8 and y - i < 8:

if mat\_knopok[x - i][y - i]['image'] == str(shash\_M):

chek\_M\_4 = 1

elif mat\_knopok[x - i][y - i]['image'] == str(shash\_M) and chek\_M\_4 == 1:

mat\_knopok[x - i][y - i]['image'] = ''

rastanovka[x - i][y - i] = 0

score += 1

if (mat\_knopok[x - i][y - i]['image'] == str(shash))and chek\_M\_4 == 1:

mat\_knopok[x - i][y - i]['image'] = ''

rastanovka[x - i][y - i] = 0

score+=1

mat\_knopok[x - i][y - i]['bg'] = 'black'

elif clear == 0:

#-------------------------------подсветка доступных ходов-------------------------------

suma\_hod\_mat = 0;suma\_bliz = 0;suma\_dal = 0;max\_poz = 0;return\_check = 0

if k == -1 or k == 1:

#-------------------------------для шашки-------------------------------

for i in range(x-1,x+2):

for j in range(y-1,y+2):

if (i >= 0 and j >= 0 and i != x and j != y and i < 8 and j < 8 and mat\_knopok[i][j]['image'] == '' and mat\_knopok[i][j]['bg'] == 'black') and check\_dal == 0 and check\_bliz == 0:

if (x - i) != hodim:

if bot == 0:

mat\_knopok[i][j]['bg'] = '#8B2323'

suma\_bliz += 1

poz\_bot\_x\_next = i

poz\_bot\_y\_next = j

else:

suma\_bliz += 1

if max\_poz < ves[i][j] and suma\_bliz > 0 and mat\_knopok[i][j]['state'] == NORMAL and mat\_knopok[x][y]['state'] == NORMAL:

max\_poz = ves[i][j]

poz\_bot\_x = x

poz\_bot\_y = y

poz\_bot\_x\_next = i

poz\_bot\_y\_next = j

elif i >= 0 and j>=0 and i != x and j !=y and i<8 and j<8 and mat\_knopok[i][j]['image'] != '' and mat\_knopok[i][j]['bg'] == 'black':

if ((i+(i-x) >= 0) and (j+(j-y)>=0) and (i+(i-x)) < 8 and (j+(j-y)) < 8 and i != x and j != y and mat\_knopok[i + (i - x)][j + (j - y)]['image'] == '') and check\_bliz == 0:

if bot == 0:

mat\_knopok[i + (i - x)][j + (j - y)]['bg'] = '#8B2323'

suma\_dal += 1

poz\_bot\_x\_next = i + (i - x)

poz\_bot\_y\_next = j + (j - y)

else:

suma\_dal += 1

if suma\_dal>0 and mat\_knopok[i][j]['state'] == NORMAL and mat\_knopok[x][y]['state'] == NORMAL:

max\_poz = ves[i+(i-x)][j+(j-y)]

poz\_bot\_x = x

poz\_bot\_y = y

poz\_bot\_x\_next = i+(i-x)

poz\_bot\_y\_next = j+(j-y)

return\_check =1

elif k == 2:

#-------------------------------для дамки-------------------------------

for i in range(1, 7):

if x + i >= 0 and y + i >= 0 and x + i < 8 and y + i < 8:

if mat\_knopok[x + i][y + i]['image'] == '':

fash = 0

if bot == 1:

max\_poz = 9999

poz\_bot\_x = x;poz\_bot\_y = y;poz\_bot\_x\_next = x + i;poz\_bot\_y\_next = y + i

else:

mat\_knopok[x + i][y + i]['bg'] = '#8B2323'

if mat\_knopok[x + i][y + i]['image'] != '' and fash == 0:

fash += 1

suma\_hod\_mat +=1

elif mat\_knopok[x + i][y + i]['image'] != '' and fash == 1:

suma\_hod\_mat = 0

break

fash = 0

for i in range(1, 7):

if x - i >= 0 and y + i >= 0 and x - i < 8 and y + i < 8:

if mat\_knopok[x - i][y + i]['image'] == '':

fash = 0

if bot == 1:

max\_poz = ves[x - i][y + i];poz\_bot\_x = x;poz\_bot\_y = y

poz\_bot\_x\_next = x - i;poz\_bot\_y\_next = y + i

else:

mat\_knopok[x - i][y + i]['bg'] = '#8B2323'

if mat\_knopok[x - i][y + i]['image'] != '' and fash == 0:

fash += 1

suma\_hod\_mat += 1

elif mat\_knopok[x - i][y + i]['image'] != '' and fash == 1:

fash = 0

suma\_hod\_mat = 0

break

fash = 0

for i in range(1, 7):

if x + i >= 0 and y - i >= 0 and x + i < 8 and y - i < 8:

if mat\_knopok[x + i][y - i]['image'] == '':

fash = 0

if bot == 1:

max\_poz = ves[x + i][y - i]

poz\_bot\_x = x;poz\_bot\_y = y

poz\_bot\_x\_next = x + i;poz\_bot\_y\_next = y - i

else:

mat\_knopok[x + i][y - i]['bg'] = '#8B2323'

if mat\_knopok[x + i][y - i]['image'] != '' and fash == 0:

fash += 1

suma\_hod\_mat += 1

elif mat\_knopok[x + i][y - i]['image'] != '' and fash == 1:

fash = 0

suma\_hod\_mat = 0

break

fash = 0

for i in range(1, 7):

if x - i >= 0 and y - i >= 0 and x - i < 8 and y - i < 8:

if mat\_knopok[x - i][y - i]['image'] == '':

fash = 0

if bot == 1:

max\_poz = ves[x - i][y - i]

poz\_bot\_x = x;poz\_bot\_y = y

poz\_bot\_x\_next = x - i;poz\_bot\_y\_next = y - i

else:

mat\_knopok[x - i][y - i]['bg'] = '#8B2323'

if mat\_knopok[x - i][y - i]['image'] != '' and fash == 0:

fash += 1

suma\_hod\_mat += 1

elif mat\_knopok[x - i][y - i]['image'] != '' and fash == 1:

fash = 0

suma\_hod\_mat = 0

break

def click\_check(i, x, y, boti):

global ngo, i\_last,x\_last,y\_last,score\_igrok,score\_bot,chet\_do\_konca,i\_last\_last,hodim,x\_last\_last,y\_last\_last,ch,suma\_dal,suma\_bliz,igr,check\_dal,check\_bliz,check\_hod\_mat

#-------------------------------первое нажатие-------------------------------

if ngo == 1 and mat\_knopok[x][y]['image'] != '':

ngo = 2

i\_last = i

x\_last = x

y\_last = y

pokaz\_dostup\_hodov(x, y, 0, 0, rastanovka[x][y])

if suma\_dal == 0 and suma\_bliz == 0 and rastanovka[x][y] != 2:

showerror(title="Not", message="Ходов 0")

ngo = 1

ch = 0

return

if boti == 1:

ngo = 2

click\_check(i, poz\_bot\_x\_next, poz\_bot\_y\_next, 1)

#-------------------------------второе нажатие-------------------------------

elif ngo == 2 and mat\_knopok[x\_last][y\_last]['image'] != '':

#-------------------------------если ход на две клетки-------------------------------

if abs(y - y\_last) == 2 and mat\_knopok[(x + x\_last) // 2][(y + y\_last) // 2]['image'] != '' and mat\_knopok[x][y]['image'] == '':

mat\_knopok[(x + x\_last) // 2][(y + y\_last) // 2]['image'] = ''

rastanovka[(x + x\_last) // 2][(y + y\_last) // 2] = 0

check\_dal += 1

#-------------------------------добавление к результату-------------------------------

if hodim == -1:

score\_igrok += 1

canvas\_igr.itemconfigure(sc\_1, text=f"Cчёт первого игрока: {score\_igrok}")

chet\_do\_konca = 16

canvas\_igr.itemconfigure(hodi, text=f'До конца{chet\_do\_konca}')

elif hodim == 1:

score\_bot += 1

canvas\_igr.itemconfigure(sc\_2, text=f"Cчёт второго игрока: {score\_bot}")

chet\_do\_konca = 16

canvas\_igr.itemconfigure(hodi, text=f'До конца{chet\_do\_konca}')

#-------------------------------если ход на более двух клеток-------------------------------

if abs(y-y\_last) > 2 and mat\_knopok[x][y]['image'] == '':

if hodim == -1:

score\_igrok += score

canvas\_igr.itemconfigure(sc\_1, text=f"Cчёт первого игрока: {score\_igrok}")

chet\_do\_konca = 16

canvas\_igr.itemconfigure(hodi, text=f'До конца{chet\_do\_konca}')

elif hodim == 1:

score\_bot += score

canvas\_igr.itemconfigure(sc\_2, text=f"Cчёт второго игрока: {score\_bot}")

chet\_do\_konca = 16

canvas\_igr.itemconfigure(hodi, text=f'До конца{chet\_do\_konca}')

check\_hod\_mat += 1

#-------------------------------если нажата даже клетка-------------------------------

if x == x\_last and y == y\_last:

mat\_knopok[x\_last][y\_last]['bg'] = 'black'

pokaz\_dostup\_hodov(x\_last, y\_last, 1, 0, rastanovka[x][y])

if check\_dal != 0:

smena\_hoda()

ngo = 1

ch = 0

check\_dal = 0

check\_bliz = 0

x\_last = 0

y\_last = 0

bot()

return

else:

ngo = 1

return

#-------------------------------если нажата кнопка с подсветкой-------------------------------

if mat\_knopok[x][y]['bg'] == '#8B2323':

if abs(x-x\_last) == 1:

check\_bliz =1

chet\_do\_konca -=1

canvas\_igr.itemconfigure(hodi, text=f'До конца{chet\_do\_konca}')

mat\_knopok[x\_last\_last][y\_last\_last]['state'] = NORMAL

if rastanovka[x\_last][y\_last] == 2:

mat\_knopok[x][y]['image'] = shash\_M

else:

mat\_knopok[x][y]['image'] = shash

mat\_knopok[x][y]['bg'] = 'black'

mat\_knopok[x\_last][y\_last]['image']= ''

rastanovka[x][y] = rastanovka[x\_last][y\_last]

rastanovka[x\_last][y\_last] = 0

pokaz\_dostup\_hodov(x\_last, y\_last, 1, 0, rastanovka[x][y])

if rastanovka[x][y] != 2:

pokaz\_dostup\_hodov(x, y, 0, 0, rastanovka[x][y])

if suma\_dal == 0 and suma\_bliz == 0 or (check\_hod\_mat == 1 and suma\_hod\_mat == 1):

x\_last\_last = x;y\_last\_last = y

smena\_hoda()

ngo = 1

ch = 0

pokaz\_dostup\_hodov(x\_last, y\_last, 1, 0, rastanovka[x][y])

check\_dal = 0;check\_bliz = 0

x\_last = 0;y\_last = 0

if y in [0, 1, 2, 3, 4, 5, 6, 7] and ((x == 0) or (x == 7)):

mat\_knopok[x][y]['image'] = shash\_M

rastanovka[x][y] = 2

mat\_knopok[x][y]['state'] = DISABLED

if boti != 1:

bot()

proverka\_na\_conec(score\_igrok, score\_bot)

elif ((suma\_dal > 0 and suma\_bliz == 0) or check\_hod\_mat == 0) and boti == 0:

ngo = 1

click\_check(i, x, y, 0)

elif suma\_dal > 0 and suma\_bliz == 0 and boti == 1:

ngo = 1

click\_check(i, x, y, 1)

#-------------------------------если нажата кнопка без подсветка-------------------------------

elif mat\_knopok[x][y]['bg'] != '#8B2323':

showerror(title="Not", message="Запрещено")

ngo = 2

return

#-------------------------------проверка на конец-------------------------------

def otris(canvas\_igr,igr):

global rastanovka

rastanovka = [[0, 1, 0, 1, 0, 1, 0, 1],

[1, 0, 1, 0, 1, 0, 1, 0],

[0, 1, 0, 1, 0, 1, 0, 1],

[0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0],

[-1, 0, -1, 0, -1, 0, -1, 0],

[0, -1, 0, -1, 0, -1, 0, -1],

[-1, 0, -1, 0, -1, 0, -1, 0]]

for i in range(8):

mat\_knopok.append([0] \* 8)

global sc\_1,sc\_2,hodi

#-------------------------------отрисовка фона поля и текста-------------------------------

canvas\_igr.create\_rectangle(2, 200, 200, 400, fill="white", outline="black")

sc\_1 = canvas\_igr.create\_text(105, 230, text=f"Cчёт игрока: {score\_igrok}", font=("Compact 18 bold"), fill="black")

sc\_2 = canvas\_igr.create\_text(110, 300, text=f"Cчёт бота: {score\_bot}", font=("Compact 18 bold"), fill="black")

hodi = canvas\_igr.create\_text(105, 370, text=f'До конца: {chet\_do\_konca}', font=("Compact 18 bold"), fill="black")

canvas\_igr.create\_rectangle(340, 10, 1033, 705, fill="black", outline="black")

canvas\_igr.create\_text(695, 720, text=f'a b c d e f g h', font=("Compact 25 bold"), fill="black")

canvas\_igr.create\_text(320, 660, text=f'1', font=("Compact 25 bold"), fill="black");canvas\_igr.create\_text(320, 570, text=f'2', font=("Compact 25 bold"), fill="black")

canvas\_igr.create\_text(320, 480, text=f'3', font=("Compact 25 bold"), fill="black");canvas\_igr.create\_text(320, 400, text=f'4', font=("Compact 25 bold"), fill="black")

canvas\_igr.create\_text(320, 310, text=f'5', font=("Compact 25 bold"), fill="black");canvas\_igr.create\_text(320, 230, text=f'6', font=("Compact 25 bold"), fill="black")

canvas\_igr.create\_text(320, 150, text=f'7', font=("Compact 25 bold"), fill="black");canvas\_igr.create\_text(320, 65, text=f'8', font=("Compact 25 bold"), fill="black")

n = 350;w = 20

color = 'white'

#-------------------------------отрисовка поля и раставление шашек-------------------------------

for i1 in range(8):

for j1 in range(8):

i = i1 \* 8 + j1

if i1 % 2 != 0:

if i % 2 != 0:

color = "white"

else:

color = "black"

elif i1 % 2 == 0:

if i % 2 == 0:

color = "white"

else:

color = "black"

if rastanovka[i1][j1] == 1 or rastanovka[i1][j1] == -1:

mat\_knopok[i1][j1] = (Button(igr, bg=color, fg="black", activebackground="white", relief="flat", command=lambda i =i, i1=i1, j1=j1: click\_check(i, i1, j1, 0)))

else:

mat\_knopok[i1][j1] = (Button(igr, bg=color, fg="white", activebackground="white", relief="flat", command=lambda i=i, i1=i1, j1=j1: click\_check(i, i1, j1, 0)))

if mat\_knopok[i1][j1]['bg'] == 'white':

mat\_knopok[i1][j1]['state'] = DISABLED

canvas\_igr.create\_window(n, w, anchor=NW, window=mat\_knopok[i1][j1], width=80, height=80)

if i in [1,3,5,7,8,10,12,14,17,19,21,23,40,42,44,46,49,51,53,55,56,58,58, 65,60,62,62, 69]:

mat\_knopok[i1][j1].config(image=shash)

n += 85

n = 350;w += 85

def igra(glavmenu):

#-------------------------------создание окна игры-------------------------------

global color,igr

glavmenu.withdraw()

igr = Tk()

igr.title("Шашки Артамонова")

igr.geometry(f"1384x753+{wh}+{hh}")

igr.resizable(False, False)

global canvas\_igr

canvas\_igr = Canvas(igr, bg='white', width=1384, height=753)

canvas\_igr.place(x=0, y=0)

img\_igr = PhotoImage(master=canvas\_igr, file='background.png')

global shash,shash\_M

shash = PhotoImage(master=canvas\_igr, file='shashka.png')

shash = shash.subsample(2, 2)

shash\_M = PhotoImage(master=canvas\_igr, file='shashka\_damka.png')

shash\_M = shash\_M.subsample(2, 2)

canvas\_igr.create\_image(0, 0, anchor='nw', image=img\_igr)

otris(canvas\_igr,igr)

igr.mainloop()

def rezul(canvas\_glav,igr):

#-------------------------------окно результатов-------------------------------

canvas\_glav.create\_rectangle(920, 210, 1320, 250, fill="white", outline="black",tags='nastr')

canvas\_glav.create\_text(1120, 230, text="Результаты", font=("Compact 18 bold"), fill="black",tags='nastr')

canvas\_glav.create\_rectangle(920, 250, 1320, 580, fill="#525252", outline="black",tags='nastr')

canvas\_glav.create\_rectangle(1030, 330, 1230, 430, fill="white", outline="black", tags='nastr')

canvas\_glav.create\_text(1130, 350, text=f"Выйграно {win\_igr}", font=("Compact 18 bold"), fill="black", tags='nastr')

canvas\_glav.create\_text(1130, 380, text=f"Проиграно {win\_bot}", font=("Compact 18 bold"), fill="black", tags='nastr')

canvas\_glav.create\_text(1130, 410, text=f"Ничья {nich}", font=("Compact 18 bold"), fill="black", tags='nastr')

but\_zak = Button(igr,text="Закрыть", bg="white", fg="black", activebackground="white",

font=("Compact 15 bold"), relief='flat', cursor="hand2",command= lambda: canvas\_glav.delete('nastr'))

canvas\_glav.create\_window(1030, 500, anchor=NW, window=but\_zak, width=200, height=35,tags='nastr')

def vih():

exit()

def glav\_menu():

#-------------------------------создание окна-------------------------------

global glavmenu

glavmenu = Tk()

glavmenu.title("Главное меню")

glavmenu.geometry(f"1384x753+{wh}+{hh}")

glavmenu.resizable(False, False)

canvas\_glav\_menu = Canvas(glavmenu, bg='white', width=1384, height=753)

canvas\_glav\_menu.place(x=0, y=0)

# -------------------------------фон-------------------------------

img\_menu = PhotoImage(master = canvas\_glav\_menu,file='background.png')

canvas\_glav\_menu.create\_image(0, 0, anchor='nw', image=img\_menu)

# -------------------------------размещение блоков-------------------------------

canvas\_glav\_menu.create\_rectangle(500, 210, 900, 250, fill="white", outline="black")

canvas\_glav\_menu.create\_text(700, 230, text="Главное меню", font=("Compact 18 bold"), fill="black")

canvas\_glav\_menu.create\_rectangle(500, 250, 900, 580, fill="#525252", outline="black")

# -------------------------------размещение кнопок-------------------------------

but\_nach = Button(glavmenu, text="Начать игру", bg="white", fg="black", activebackground="white",

font=("Compact 15 bold"), relief='flat', cursor="hand2", command= lambda:igra(glavmenu)) # начло игры

canvas\_glav\_menu.create\_window(550, 300, anchor=NW, window=but\_nach, width=300, height=50)

but\_nastr = Button(glavmenu, text="Результаты", bg="white", fg="black", activebackground="white",

font=("Compact 15 bold"), relief='flat', cursor="hand2", command= lambda: rezul(canvas\_glav\_menu, glavmenu)) # настройки

canvas\_glav\_menu.create\_window(550, 370, anchor=NW, window=but\_nastr, width=300, height=50)

but\_vih = Button(glavmenu, text="Выход", bg="white", fg="black", activebackground="white",

font=("Compact 15 bold"), relief='flat', cursor="hand2", command=vih) # выход

canvas\_glav\_menu.create\_window(550, 440, anchor=NW, window=but\_vih, width=300, height=50)

glavmenu.mainloop()

def paf(par\_ent):

if enabled.get() == 1:

par\_ent['show'] = ''

if enabled.get() == 0:

par\_ent['show'] = '\*'

def ok():

if (len(passwr\_register.get()) >= 4 and len(passwr\_register.get()) <=10) and (log\_registr.get().count(' ') == 0 and passwr\_register.get().count(' ') == 0):

showinfo(title="Информация", message="Регистрация успешно пройдена")

f = open('base.txt', 'a+')

canvas.delete("reg")

f.write(log\_registr.get() +' '+ passwr\_register.get()+'\n')

f.close()

log\_registr.set('')

passwr\_register.set('')

else:

showerror(title="Ошибка", message="Пароль должен быть от 4 до 8 символов")

def avtor():

if (len(login.get()) == 0 or len(passw.get()) == 0) or (len(login.get()) == 0 and len(passw.get()) == 0):

showerror(title="Ошибка", message="Поля не должны быть пустыми")

else:

f = open('base.txt', 'r')

while TRUE:

line = f.readline()

line = line.split()

if len(line) == 0:

f.close()

showerror(title="Ошибка", message="Логин и пароль не найдены")

break

elif line[0] == login.get() and line[1] == passw.get():

f.close()

showinfo(title="Информация", message="Авторизация успешно пройдена")

root.destroy()

glav\_menu()

break

def registr():

block\_naz = canvas.create\_rectangle(890, 210, 1200, 250, fill="white", outline="black",tags= "reg")

block\_one = canvas.create\_rectangle(890, 250, 1200, 550, fill="#525252", outline="black",tags= "reg")

leib1 = canvas.create\_text(1050, 230, text="Регистрация", font=("Compact 18 bold"), fill="black",tags= "reg")

log\_block = canvas.create\_rectangle(950, 280, 1150, 320, fill="white", outline="black",tags= "reg")

log\_reg = canvas.create\_text(1050, 300, text="Введите логин\n без пробелов", font=("Compact 13 bold"), fill="black",tags= "reg")

login\_reg = Entry(canvas, font=("Compact 18 bold"), relief='flat',textvariable = log\_registr)

canvas.create\_window(950, 325, anchor=NW, window=login\_reg, width=201, height=30,tags= "reg")

passw\_block = canvas.create\_rectangle(950, 370, 1150, 410, fill="white", outline="black", tags="reg")

passw\_reg = canvas.create\_text(1050, 390, text=" Введите пароль\n от 4 до 8 символов", font=("Compact 13 bold"), fill="black", tags="reg")

passwor\_reg = Entry(canvas, font=("Compact 18 bold"), relief='flat',textvariable = passwr\_register)

canvas.create\_window(950, 415, anchor=NW, window=passwor\_reg, width=201, height=30, tags="reg")

but\_two = Button(root, text="Зарегистрироваться", bg="white", fg="black", activebackground="white",

font=("Compact 11 bold"),

relief='flat', command= ok, cursor="hand2")

canvas.create\_window(980, 460, anchor=NW, window=but\_two, width=150, height=30, tags="reg")

def reg(login,passw):

image = canvas.create\_image(0, 0, anchor='nw', image=img)

canvas.create\_rectangle(500, 210, 884, 250, fill="white", outline="black")

leib = canvas.create\_text(695, 230, text="Авторизация", font=("Compact 18 bold"), fill="black")

block\_one = canvas.create\_rectangle(500, 250, 884, 550, fill="#525252", outline="black")

# -------------------------------Окно логина-------------------------------

login\_text = canvas.create\_text(600, 325, text="Логин:", font=("Compact 15 bold"), fill="black")

log\_block = canvas.create\_rectangle(550, 299, 649, 350, fill="white", outline="black")

canvas.tag\_lower(log\_block,login\_text)

login = Entry(canvas, font=("Compact 18 bold"),relief = 'flat',textvariable = login)

canvas.create\_window(650, 300, anchor=NW, window=login, width=180, height=50)

# -------------------------------Окно пароля-------------------------------

passw\_text = canvas.create\_text(600, 390, text="Пароль:", font=("Compact 15 bold"), fill="black")

passw\_block = canvas.create\_rectangle(550, 364, 649, 415, fill="white", outline="black")

canvas.tag\_lower(passw\_block, passw\_text)

passw = Entry(canvas, font=("Compact 18 bold"),relief = 'flat',textvariable = passw,show = '\*')

canvas.create\_window(650, 365, anchor=NW, window=passw, width=180, height=50)

enabled\_checkbutton = Checkbutton(text="", bg="#525252", activebackground="#525252", variable=enabled,

command=lambda: paf(passw))

canvas.create\_window(830, 380, anchor=NW, window=enabled\_checkbutton, width=0, height=0)

# -------------------------------Кнопка авторизации-------------------------------

but\_one = Button(root,text="Авторизация",bg ="white",fg="black",activebackground="white",font=("Compact 11 bold"),relief = 'flat', cursor="hand2",command=avtor)

canvas.create\_window(595, 430, anchor=NW, window=but\_one, width=200, height=30)

# -------------------------------Кнопка регистрации-------------------------------

log\_block = canvas.create\_rectangle(550, 480, 700, 510, fill="white", outline="black")

login\_text = canvas.create\_text(625, 495, text="Первый раз в игре?", font=("Compact 11 bold"), fill="black")

but\_two = Button(root, text="Зарегистрируйся!!!", bg="white", fg="black", activebackground="white", font=("Compact 11 bold"),

relief='flat',command=registr, cursor="hand2")

canvas.create\_window(701, 480, anchor=NW, window=but\_two, width=150, height=30)

reg(login,passw)

root.mainloop()