**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 «Компьютерная логическая игра «Русские 80-клеточные шашки»** 

**Руководство программиста**

Р.02069337. 21/849-01 РП-01

Листов 7

**Исполнитель**:

студент гр. ИСТбд-22

*Абляев Глеб Сергеевич*

« » 2024 г.

**2024**

**1. Назначение и условия применения программы**

**1.1 Назначение и функции, выполняемые приложением**

Игра “Русские 80-клеточные шашки”

80-Клеточные шашки ("Шашки на 80 квадратов" - 80-клеточные шашки), также известные как Советские шашки ("Советские шашки" - советские шашки), представляют собой современную разновидность [русских шашек](https://translated.turbopages.org/proxy_u/en-ru.ru.743e42dc-65f6f238-b07b2d34-74722d776562/https/checkers.fandom.com/wiki/Russkie_Shashki), которая была изобретена в России около 1930 года Николаем Спанцирети (1916-1991).

Функциональные возможности приложения: авторизация, регистрация, игра в киммерийские шашки.

**1.2 Условия, необходимые для использования приложения**

1. ОС: Windows 10 21H2.
2. Среда разработки: PyCharm Community Edition Version: 2022.3.
3. Версия языка: Python 3.10.
4. Библиотеки: pygame 2.1.2, tkinter, checkers.

В качестве БД использовать файл.

**2. Характеристики программы**

**2.1 Характеристики приложения**

Количество значимых строк кода – 667.

Количество алгоритмов – 28.

Количество методов 24.

Запуск программы производится двойным щелчком мыши на файле «main.exe» или на его ярлыке.

При запуске приложения появляется окно регистрации/авторизации пользователя и выхода (Рис. 1). Логин и пароль должны состоять из набора латинских букв и цифр, с длиной символов от 1 до 16. В случае введения некорректных данных на окне появляются замечания.

Появляется окно регистрации, в котором требуется ввести пароль и имя пользователя.(Рис.1)

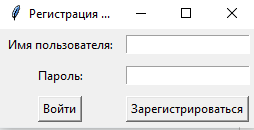


Рис.1

После по нажатию кнопки зарегистрироваться произойдет регистрация пользователя с записью данных в текстовый файл, также появится окно об успешной регистрации. (Рис.2)

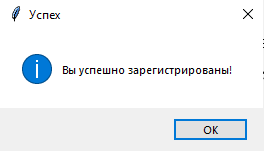


Рис.2

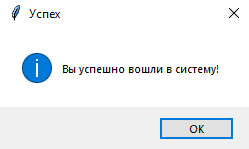
После этого пользователь вводит свои данные в окно еще раз и нажимает войти, появляется окно об успешном заходе в систему. (Рис.3) 

Рис.3

На рис. 4 представлен геймплей игры. Пользователь по нажатию на свои шашки может совершать ходы в соответствии с правилами игры.

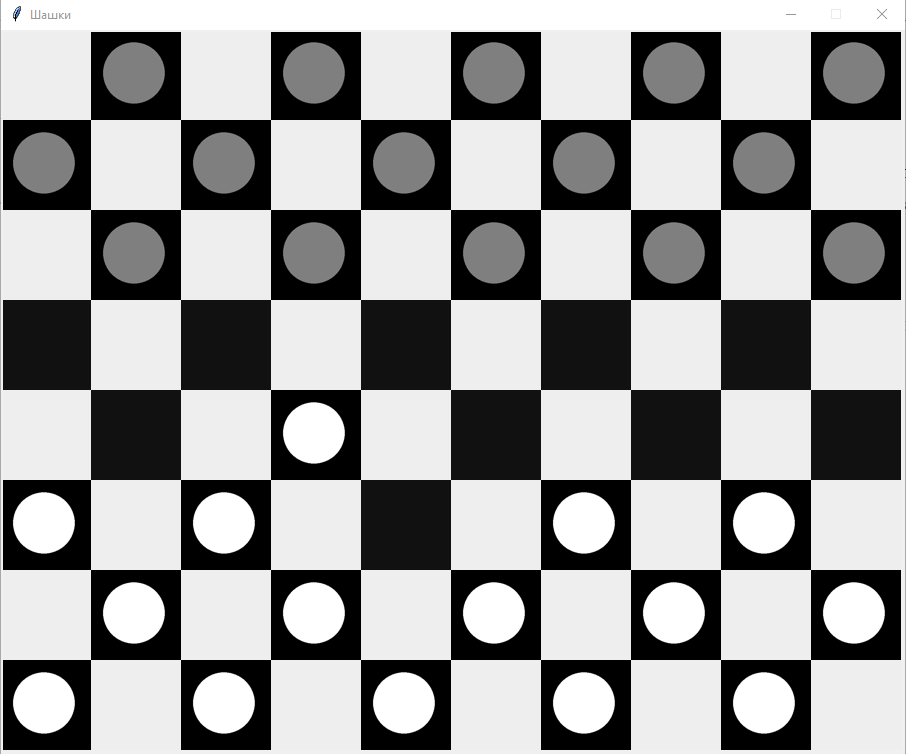


Рис.4

Результат игры можно увидеть на рис.5. По окончанию игры появляется окно с надписью о победе черных или белых.

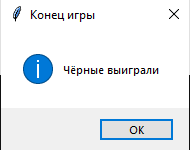


Рис.5

**2.2 Особенности реализации приложения**

В программе используются массивы, отвечающие за наименование координат игрового поля и для проверки победителя, правильности ходов, работы компьютера.

Приложение используются библиотеки:

* tkinter-использовалась для отрисовки шахматной доски и меню.

**3. Обращение к программе**

В классе Game:

\_\_init\_\_() : Конструктор класса, инициализирует игровое поле, изображения шашек, устанавливает очередь игрока. Если игрок играет за черных, совершает ход противника.

\_\_init\_images(self): Инициализирует изображения шашек, загружая их из файлов.

\_\_animate\_move(self, move: Move): Анимирует ход шашки на игровом поле.

\_\_draw(self): Очищает холст и перерисовывает игровое поле и шашки.

\_\_draw\_field\_grid(self): Рисует сетку игрового поля.

\_\_draw\_checkers(self): Рисует шашки на игровом поле.

\_\_handle\_player\_turn(self, move: Move): Обрабатывает ход игрока.

\_\_handle\_enemy\_turn(self): Обрабатывает ход противника.

\_\_check\_for\_game\_over(self): Проверяет, не завершилась ли игра, и выводит сообщение о победе одной из сторон.

Класс Checker:

\_\_init\_\_(): Конструктор класса. Принимает размеры игрового поля по оси X и по оси Y и инициализирует поле.

x\_size(self) -> int: Свойство, возвращающее размер игрового поля по оси X.

y\_size(self) -> int: Свойство, возвращающее размер игрового поля по оси Y.

size(self) -> int: Метод, возвращающий максимальный размер игрового поля измеренный по оси X или Y.

copy( ): Метод класса, создающий копию объекта игрового поля.

\_\_generate(self): Приватный метод, генерирующий начальную расстановку шашек на игровом поле.

type\_at( ) -> CheckerType: Метод, возвращающий тип шашки по указанным координатам.

at( ) -> Checker: Метод, возвращающий объект шашки по указанным координатам.

is\_within( ) -> bool: Метод, проверяющий, находится ли точка с указанными координатами в пределах игрового поля.

white\_checkers\_count(self) -> int: Метод, возвращающий количество белых шашек на поле.

black\_checkers\_count(self) -> int: Метод, возвращающий количество черных шашек на поле.

white\_score(self) -> int: Метод, возвращающий сумму очков белых шашек на поле.

black\_score(self) -> int: Метод, возвращающий сумму очков черных шашек на поле.

Библиотеки:

1. tkinter - использовалась для создания шахматной доски и меню.
2. checkers - для проверки ходов и перемещение шашек.

В виде БД обычные файлы txt.

**4. Сообщения**

При победе программа выводит окно с сообщениями: “черные выиграли” или “белые выиграли”

При некорректном вводе логина и пароля отображается замечание: “неправильное имя пользователя или пароль”