**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 “Компьютерная логическая игра «Эндшпиль - король, 2 пешки-король, ладья»”

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Р.02069337.22/2410-14 ТЗ-02

Листов 6

**Исполнитель**:

студент гр. ИСТбд-21

*Лашманов Н.Н.*

« » 2023 г.

**2023**

**Введение**

Компьютерная логическая игра «Эндшпиль - король, 2 пешки-король, ладья»

Приложение должно соответствовать правилам игры, приведенным ниже.

**Поле и игроки.** Играют двое игроков. Игра ведется на шахматном поле. Шахматная доска состоит из 64 (8х8) одинаковых клеток, попеременно белых и чёрных.

**Шахматные фигуры.** В шахматы играют следующими фигурами разного цвета: пешка, ладья и король. Перед началом игры игрокам предоставляется: король и две пешки одному игроку, король и ладья другому.

**Ходы.** Ходом считается передвижение одной фигуры на другую клетку. Фигуры не могут ходить на место другой фигуры своего цвета, или чужого, если фигура не может совершить взятие вражеской фигуры.

Пешка может ходить на одну или две клетки вперёд, ход на две клетки возможен только, если пешкой ранее не был сделан ход. Пешку можно превратить в ладью, если пешка дошла до конца шахматного поля напротив себя.

Ладья может пойти на любое поле по вертикали или горизонтали, на которых она стоит.

Король может ходить на любые восемь клеток вокруг себя, если они не «под угрозой». «Под угрозой» клетка считается, если фигура противника может ходить по ней. Король может совершить рокировку с ладьёй. Рокировка - это ход короля и одной из ладей того же цвета на одной и той же горизонтали, считающийся как единственный ход короля и выполняемый следующим образом: король перемещается со своего начального места на два поля по направлению к ладье, затем эта ладья перемещается на поле, которое король только что пересек. Рокировка невозможна если король или ладья уже ходили.

Белые фигуры всегда ходят первыми, после них чёрные и далее по очереди.

**Стартовая позиция.** Шахматные фигуры расставляются следующим образом: начальная позиция одного игрока в виде двух пешек на клетках d2 и e2 и короля на e1, и начальная позиция второго игрока в виде ладьи на h8 и короля на e8.

**Взятие.** Пешка может взять фигуру противника, если она находится по диагонали вплотную перед ней.

Ладья может взять другую фигуру противника, если эта фигура находится по горизонтали или по вертикали от ладьи.

Король может взять любую фигуру противника, находящуюся вплотную к нему. Короля нельзя взять, но можно поставить ему шах. Под шахом понимается ситуация, когда фигура противника следующим ходом может совершить взятие короля. В таком случае король под шахом будет вынужден уйти от атаки, либо требуется поставить фигуру на пути атакующей фигуры или совершить взятие угрожающей фигуры.

**Выигрыш партии**. Выигравшим партию признается тот, кто поставит мат королю противника. Матом считается ситуация, когда король не имеет защиты и не может уйти от атаки.

**Ничья.** Ничьёй или патом считается ситуация, если невозможно поставить мат королю противника, например, если у сторон не осталось фигур или возникла одинаковая позиция по крайней мере три раза.

**1. Основания для разработки**

В качестве оснований для разработки указывается учебный план направления 09.03.02 «Информационные системы и технологии» и распоряжение по факультету.

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

**2.1. Функциональное назначение**

Требуется разработать однопользовательское десктопное приложение по игре в шашки 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 нескольких модулей, взаимодействующих между собой с использованием дополнительных информационных файлов.

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 компьютера;

- информирование пользователя об окончании игры и победителе.

2.2.2 Требования к организации информационного обеспечения, входных и выходных данных

В приложении должен быть реализован графический интерфейс взаимодействия с пользователем. Отдельно выделены папки под графические файлы, шрифт, заготовку объектов и карты, аудио эффектов, а также для самого кода. Логин и пароль пользователя должны вводиться с клавиатуры. Логины и пароли зарегистрированных пользователей должны храниться в отдельном файле или базе данных в зашифрованном виде.

**2.3 Требования к надёжности**

Поддержка непрерывной и стабильной работы компьютера.

**2.4 Требования к информационной и программной совместимости**

Рекомендуется к использованию на Windows 7.

При создании программы используются встроенные библиотеки “random”, “json”, “copy ”, “itertools”, “math”, “base64”.

И сторонние библиотеки “tkinter 8.6.”.

Разработка ведётся в “PyCharm community edition 2022.3” на версии языка программирования Python 3.9.

**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

В хранилище репозиториев GitHub.

2.6 3 Сроки хранения

Срок хранения – до июля 2026 года**.**

**3. Требования к программной документации**

1. «Техническое задание» на реализуемое приложение должно соответствовать ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению»;
2. «Пояснительная записка» должна соответствовать ГОСТ 19.404-79 «Пояснительная записка. Требования к содержанию и оформлению»;
3. «Руководство программиста» должна соответствовать ГОСТ 19.504-79 «Руководство программиста. Требования к содержанию и оформлению»;
4. Оформление программного кода приложения должно быть в соответствии с ГОСТ 19.401-79 «Текст программы. Требования к содержанию и оформлению».

**4. Стадии и этапы разработки**

1) Написание технического задания;

2) Написание пояснительной записки;

3) Разработка и отладка прототипа приложения;

4) Разработка и отладка окончательного варианта приложения;

5) Написание руководства программиста;

6) Оформление пояснительной записки курсовой работы;

7) Подготовка презентации.

8) Защита курсовой работы.

**5. Порядок контроля и приёмки**

Созданный программный продукт предоставляется студентом руководителю с помощью репозитория GitHub, как в исходном виде, так и преобразованный в исполняемый файл «exe». В случае наличия каких-либо замечаний, студент может доработать свой программный продукт, записав новую версию по той же ссылке. Программные документы также размещаются в репозитории GitHub. Окончательная (принятая) версия документации предоставляется в печатном виде.

График защиты курсовых работ составляется руководителем и доводится до сведения студентов. Студент обязан убедиться в корректности функционирования приложения до защиты. Проблемы, обнаруженные непосредственно при защите, являются основанием для её переноса (в соответствии с графиком).

В начале защиты студент сообщает название приложения, кратко формулирует его назначение и указывает основные особенности. В процессе защиты – демонстрирует используемую(мые) структуры данных и алгоритмы, акцентируя внимание на наиболее важных и интересных, демонстрирует работоспособность самого приложения.

Оценка работы осуществляется руководителем с учётом качества её выполнения, включая качество документации, полноты учёта предъявляемых требований, выступления с презентацией и ответов на вопросы в ходе защиты, а также соблюдения сроков выполнения и защиты курсовой работы, включая работу на практических занятиях.

Основаниями для получения неудовлетворительной оценки могут являться серьёзное несоответствие программного продукта предъявляемым требованиям, неработоспособность приложения, наличие существенных элементов заимствования из чужих работ как в программном коде или интерфейсе приложения, так и в документации, а также слабая ориентация студента в представляемой работе. При неудовлетворительной оценке руководитель определяет направления и объём доработки программного продукта.