**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 игра «шашки-Чапаев»»

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Р.02069337.22/2372-36 ТЗ-01

Листов 6

**Исполнитель**:

студент гр. ИСТбд-23

*Безбородов Егор Юрьевич*

« » 2023 г.

**2023**

**Введение**

Компьютерная игра «шашки-Чапаев»

Приложение должно соответствовать правилам игры, приведенным ниже.

**Поле и игроки.** Играют двое игроков. Игра ведется на шашечной / шахматной доске. Доска состоит из поля - 64 (8х8) клеток, попеременно светлых и тёмных, а также окантовки доски - на ней могут быть указаны индексы полей. Доска располагается между игроками таким образом, чтобы на поле в нижнем углу слева от играющего находилась темная клетка.

**Шашки.** Шашки подразделяются на чёрные и белые. Перед началом игры игрокам предоставляется по 8 простых шашек: одному — светлых, другому — темных.

**Стартовая позиция.** Шашки расставляются на ближайших к игрокам горизонталям.

**Ходы.** Ходом является удар по шашке. Удар направлен в геометрической плоскости (щелбан в реальной жизни).

Первый ход всегда делает игрок, победивший в прошлом раунде. Если это начало матча, или прошлый раунд был ничьёй, первыми ходят белые.

Игрок продолжает свой ход до тех пор, пока за каждый свой удар, у противника влетают с поля его шашки. Если после удара, ни одна шашка у противника не слетела, то ход переходит ему. Это продолжается до тех пор, пока у обоих сторон остаются шашки на доске.

**Раунды.** В игре происходит несколько раундов. Противники ставят по 8 шашек своего цвета в один ряд перед собой. Первый раунд начинается с крайних рядов доски, а в следующем победитель двигает свои шашки на ряд вперед, сокращая тем самым расстояние до своего соперника. Если два ряда столкнутся между собой, проигравший в раунде отступает на 1 линию в свою сторону, а победитель занимает его место. В случае ничьи, игроки остаются на прежних местах. Также возможен вариант, когда победитель не потерял ни одной шашки. В этом случае движение происходит на 2 линии вперед.

### **Выигрыш партии**. Выигравшим партию признается тот, чьи шашки прижали шашки врага к его крайней линии.

**1. Основания для разработки**

В качестве оснований для разработки указывается 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 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”, “os”, “pygame”, “pygame-gui”, “hashlib”, “numpy”.

Разработка ведётся в “PyCharm community edition 2023.5” 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

Диск CD-R должен храниться при комнатной температуре, в диапазоне от 20°C до 25°C. Рекомендуется хранить диск в условиях с относительной влажностью воздуха от 20% до 50%. Диск CD-R должен храниться в темном месте, защищенном от прямых солнечных лучей и других источников яркого света. Для предотвращения повреждения диска CD-R рекомендуется хранить его в специальных пластиковых коробках или футлярах, предназначенных для хранения CD-дисков.

2.6 3 Сроки хранения

Срок хранения – до июля 2024 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.