**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования   
**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**Самолетостроительный факультет

Кафедра «Информационные технологии и общенаучные дисциплины»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

На лабораторную работу

По дисциплине «Алгоритмы и структуры данных»

ТемаРазработка логических компьютерных игр с графическим интерфейсом в среде Питон

Листов 16

Руководитель разработки:  
кандидат технических наук, доцент кафедры  
Шишкин В.В.   
«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024г.

Исполнитель:  
студент гр. АИСТбд-21  
Парфенова Я.И.  
«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024г.

2024г.

**Содержание**

Техническое задание..............................................................................................3

Пояснительная записка ........................................................................................ 8

Руководство программиста .................................................................................13

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Информационные технологии и общенаучные дисциплины»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на лабораторную работу

по дисциплине «Алгоритмы и структуры данных»

Тема Разработка логических компьютерных игр с графическим интерфейсом в среде Питон

Листов 4

Исполнитель:

студент гр. АИСТбд-21

Парфенова Я.И.

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024г.

2024

**Введение**

Крестики-нолики — это простая логическая игра на двоих участников. Игровое поле представляет собой сетку размером 3x3. Один из игроков играет крестиками (X), а другой — ноликами (O). Игроки поочередно делают ходы, заполняя одну из пустых клеток своим символом. Цель игры – первому выстроить три своих символа в ряд по горизонтали, вертикали или диагонали на игровом поле. Если все клетки заполнены, а ни одна из сторон не собрала три символа в ряд, то игра заканчивается вничью. Для нашей игры реализуется вариант игры с приложением, которое никогда не проигрывает.

**1. Основания для разработки**

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**

Приложение "Крестики-нолики" предназначено для реализации популярной логической игры, в которую играют двое участников — человек против компьютера (бота). Программа позволяет пользователю выбрать, какую сторону он хочет играть ("Крестики" или "Нолики"), и противостоит ему искусственный приложение (бот), использующий алгоритм минимакса для принятия оптимальных решений.

Перечень автоматизируемых процессов:

1. Инициализация и открытие пользовательского интерфейса.
2. Выбор стороны (крестики или нолики) пользователем.
3. Отображение игрового поля 3x3 и обновление интерфейса в реальном времени.
4. Определение победителя и окончание игры при достижении одной из выигрышных комбинаций или ничьей.
5. Автоматическое определение и выполнение ходов за бота.

Группы пользователей:

1. Обычные пользователи (игроки), желающие играть против компьютера.

**2.2. Требования к функциональным характеристикам**

**2.2.1. Требования к структуре приложения**

Приложение должно состоять из нескольких модулей/методов, каждый из которых отвечает за определенный функционал, таких как инициализация интерфейса, обработка ходов игрока и бота, проверка победителя, и окончание игры.

Должно быть четко выделено деление на функции управления игровым процессом, пользовательским интерфейсом и логикой искусственного интеллекта.

**2.2.2. Требования к составу функций приложения**

Инициализация и выбор стороны: обеспечить выбор стороны игроком, отображение интерфейса для выбора.

Игровой процесс: обеспечение работы как для хода игрока, так и для хода бота, включая проверку условий победы и ничьей.

Интеллект бота: реализация бота, который будет играть с игроком, используя алгоритм минимакса для определения оптимального хода.

Окончание игры: обнаружение условий завершения игры и информирование пользователя о результате через диалоговые окна.

**2.2.3. Требования к организации информационного обеспечения, входных и выходных данных**

Пользовательский интерфейс: Использование tkinter для создания простого и понятного интерфейса. Обеспечение централизованного размещения игрового окна на экране.

Входные данные: Обработка кликов пользователя на кнопках.

Выходные данные: Отображение состояния игры в режиме реального времени, сообщение об окончании игры (победа, поражение или ничья).

**2.3. Требования к надежности**

Приложение должно корректно обрабатывать все возможные сценарии ввода данных от пользователя (например, повторный выбор одной и той же ячейки) без сбоев.

Индикация победы должна быть четкой и своевременной, предоставляя пользователю четкое понимание, что игра завершена.

При возникновении сбоев приложение должно обеспечивать возможность перезапуска игры без необходимости перезапуска программы.

**2.4. Требования к информационной и программной совместимости**

Операционная система: приложение должно работать на всех основных операционных системах, поддерживающих Python и tkinter (Windows, macOS, Linux).

Используемые библиотеки: требуется наличие стандартной библиотеки tkinter, которая является частью Python.

Язык программирования: программа написана на Python (рекомендуемая версия 3.x).

Среда разработки: любая интегрированная среда разработки, поддерживающая Python, может быть использована для разработки и тестирования приложения.

**2.5. Требования к маркировке и упаковке**

Определяются заданием на лабораторную работу.

**2.6. Требования к транспортированию и хранению**

**2.6.1 Условия транспортирования**

Требования к условиям транспортирования не предъявляются.

**2.6.2 Условия хранения**

Обеспечение свободного доступа к проекту в репозитории до окончания срока учебы.

**2.6.3 Сроки хранения**

Срок хранения – до окончания срока учебы.

**3. Требования к программной документации**

Определяются заданием на лабораторную работу.

**4. Стадии и этапы разработки**

Определяются заданием на лабораторную работу.

**5. Порядок контроля и приемки**

Определяются заданием на лабораторную работу.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования   
**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**Самолетостроительный факультет

Кафедра «Информационные технологии и общенаучные дисциплины»

**Лабораторная работа**

**По дисциплине «Алгоритмы и структуры**

**данных»**

**Тема** Разработка логических компьютерных игр с графическим интерфейсом в среде Питон

**Пояснительная записка**

Листов 4

Исполнитель:  
студент гр. АИСТбд-21  
Парфенова Я.И.  
«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024г.

2024

**Введение**

Разрабатываемое приложение носит название «Крестики-нолики». Его условное обозначение для целей проектирования и документации будет «ТТТ» (от английского «Tic-Tac-Toe»).

Игра, реализованная в данном приложении, известна как «Крестики-нолики», одна из самых популярных и простых логических игр, широко используемая и известная по всему миру.

Игра «Крестики-нолики» была выбрана для реализации в рамках данного проекта из-за ее простоты, популярности и образовательного потенциала в обучении основам программирования и алгоритмизации. Из-за фиксированной структуры и ограниченного числа возможных конфигураций игрового поля, данная игра позволяет эффективно использовать и внедрять алгоритмические стратегии, такие как минимакс, для создания непобедимого приложения с высокой степенью сложности игры. Это позволяет развивать навыки анализа и оптимизации алгоритмов, а также проектирования устойчивых программных решений.

Разработанное приложение представляет собой игру в «Крестики-нолики» с возможностью играть против компьютера. Приложение включает в себя интуитивно понятный интерфейс пользователя, управление исходами игры (победа, поражение, ничья), а также реализацию механизма оптимальной игры. Поддерживаются дополнительные функции перезапуска сессий и смены сторон игроком.

**1. Проектная часть**

**1.1. Постановка задачи на разработку приложения**

Определяется заданием на лабораторную работу. Детализируется в разработанном техническом задании.

**1.2. Математические методы**

В реализации приложения используется алгоритм минимакса для управления стратегией игры приложения. Алгоритм минимакс представляет собой деревоподобную структуру, где каждый узел соответствует состоянию игры, а его дочерние узлы представляют возможные следующие ходы. Алгоритм проводит полный перебор всех возможных ходов, чтобы найти оптимальный путь, минимизируя максимальные потери — данная стратегия гарантирует, что приложение никогда не проиграет, если игрок не сделает ошибки.

Обоснование выбора: минимакс идеально подходит для игр с конечным и относительно малым числом возможных ходов, что делает его подходящим для задания «Крестики-нолики».

**1.3. Архитектура и алгоритмы**

**1.3.1 Архитектура**

Архитектура приложения основывается на модульном подходе, включающем:

Модуль графического интерфейса (GUI), отвечающий за визуализацию игрового процесса.

Модуль игровой логики, содержащий основные правила и механики игры.

Модуль искусственного интеллекта, реализующий алгоритм минимакса.

Модуль управления состоянием, отвечающий за состояние игры: регистрация ходов, определение победителя, перезапуск игры.

Взаимодействие между модулями происходит через четко определенные интерфейсы и сигналы, обеспечивающие стабильность и расширяемость приложения.

**1.3.2 Алгоритм минимакс**

Основной алгоритм минимакс реализуется через рекурсию и включает в себя следующие ключевые шаги:

Генерация всех возможных состояний игрового поля после текущего хода.

Рекурсивный расчет оценки каждого состояния с учетом возможных ответных ходов противника.

Выбор хода, который минимизирует наихудший исход, рассматриваемый как максимальная потенциальная потеря.

**1.4. Тестирование**

**1.4.1 Описание отчета о тестировании**

Отчет о тестировании служит документированием процесса проверки корректности и надежности работы приложения «Крестики-нолики». Он содержит информацию о проведенных тестах, описания тестовых сценариев, и оценку их результатов.

**1.4.2 Цель тестирования**

Целями тестирования являются: подтверждение правильного выполнения игровых правил, проверка надежности и устойчивости приложения к некорректным действиям пользователя, а также оценка продуктивности алгоритма минимакса в условиях ограниченного времени на ход.

**1.4.3 Методика тестирования**

Используются как ручные, так и автоматизированные методы тестирования. Ручные тесты включают тестирование всех возможных исходов игры, включая крайние случаи. Автоматизированные тесты проверяют корректность алгоритмов с использованием юнит-тестирования.

**1.4.4 Проведенные тесты**

Тестовые сценарии включают:

Проведение игры до достижения всех возможных исходов (победа X/O, ничья).

Проверка некорректного или дублирующегося ввода.

Производительность и время ответа при различных стадиях игры.

**1.4.5 Выводы**

Тестирование показало, что приложение стабильно и корректно функционирует в большинстве сценариев использования. Однако выявлены некоторые области для улучшения, такие как оптимизация графического интерфейса для снижения нагрузки на ресурсы системы и улучшение визуальных уведомлений для пользователя.

**2. Источники, использованные при разработке**

1. Шишкин, В. В., & Афонин, Д. С. (2023). Разработка логических компьютерных игр с графическим интерфейсом в среде Питон. УлГТУ. [Электронный ресурс].
2. Воронина, В. В., Шишкин, В. В. (2023). Компьютерная графика. УлГТУ. [Электронный ресурс].
3. Стандарты ЕСПД [Электронный ресурс] <https://www.swrit.ru/gost-espd.html>
4. СПД (ГОСТ 19) Единая система программной [Электронный ресурс] <https://www.standards.ru/collection.aspx?control=40&id=868075&catalogid=temat-sbor>
5. Назина, О. (2022). Что такое тестирование. Курс молодого бойца. Книга для начинающих тестировщиков. СПб.: БХВ – Петербург. [Электронный ресурс]. http://testbase.ru/book-beginner

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования   
**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**Самолетостроительный факультет

Кафедра «Информационные технологии и общенаучные дисциплины»

**Лабораторная работа**

**По дисциплине «Алгоритмы и структуры**

**данных»**

**Тема** Разработка логических компьютерных игр с графическим интерфейсом в среде Питон

**Руководство программиста**

Листов 3

Исполнитель:  
студент гр. АИСТбд-21  
Парфенова Я.И.  
«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024г.

2024

**1. Назначение и условия применения программы**

**1.1 Назначение и функции, выполняемые приложением**

Приложение "Крестики-нолики" реализует одноименную логическую игру для двух участников, где пользователь играет против компьютера. Цель игры — достичь непрерывной линии из трех крестиков или ноликов на сетке размером 3x3 перед оппонентом. Эта игра служит тренировкой для развития стратегического мышления и помогает расслабиться.

Функции, выполняемые приложением:

1. Выбор игроком стороны: "Крестики" или "Нолики".
2. Отрисовка и обновление игрового поля в интерактивном графическом интерфейсе.
3. Позволяет игроку сделать.
4. Генерация ходов компьютером с использованием алгоритма минимакса для максимальной сложности игры против бота.
5. Определение победителя игры (или ничьи) и отображение соответствующего сообщения.
6. Обеспечение функции перезапуска игры для начала новой партии.

Свод правил игры:

Игроки ходят по очереди, заполняя клетки на поле 3x3 своими символами.

Победителем становится тот игрок, который первым выстроит три своих символа в ряд — горизонтально, вертикально или по диагонали.

Если все клетки заполнены, но победитель не определен, итог игры — ничья.

**1.2 Условия, необходимые для использования приложения**

Операционная система: приложение разработано для платформ, поддерживающих Python и библиотеку tkinter, таких как Windows, macOS и Linux.

Платформа: используется распространенная платформа Python версии 3.6 и выше.

Инструментальная среда и библиотеки:

Python: версии 3.6 или новее.

tkinter: стандартная библиотека для создания графического интерфейса в Python, обычно входит в состав Python на всех поддерживаемых платформах.

**2. Характеристики программы**

**2.1 Характеристики приложения**

Количество строк кода: примерно 100 строк кода.

Основные структуры данных: двумерный список (матрица) для представления игрового поля.

Алгоритмы: минимакс для принятия решений ботом.

Используемые библиотеки: tkinter для создания графического интерфейса.

Стандартная библиотека Python для логики игры и алгоритма минимакса.

Работа приложения и внешний вид:

При запуске открывается окно с выбором за кого будет делать ход пользователь. Далее открывается графический интерфейс с сеткой 3x3.

Пользователь может кликом мыши выбрать клетку для размещения своего символа.

После каждого хода интерфейс обновляется, показывая текущее состояние игрового поля.

По окончании игры отображается сообщение с результатом (победа, проигрыш или ничья).

Далее программа возвращается в окно выбора роли.

Контроль корректности ввода:

Контроль за вводом осуществляется предотвращением выбора уже занятой клетки, нажатия на неё игнорируются.

**2.2 Особенности реализации приложения**

Использована двумерная структура данных (список списков) для представления состояния игрового поля, что позволяет эффективно управлять содержимым клеток и проверять выигрышные комбинации.

Альтернативы, такие как использование одномерного списка с математической адресацией (индексированием), были отклонены из-за повышенной сложности и уменьшения читаемости кода.

Заимствование программных решений:

Алгоритм минимакса является общеизвестной методикой для создания ботов, играющих в стратегические игры. Источники и примеры таких реализаций легко найти в учебниках и интернет ресурсах.

**3. Обращение к программе**

Методы и алгоритмы:

Основные методы: player\_move(self, i, j), bot\_move(self), check\_winner(), is\_draw(), restart\_game().

Алгоритм минимакса реализован в виде рекурсивной функции: minimax(board, depth, is\_maximizing).

Используемые библиотеки:

tkinter: для отрисовки и управления пользовательским интерфейсом.

Другие вспомогательные компоненты стандартной библиотеки Python.

**4. Сообщения**

Сообщения контроля корректности ввода/вывода:

"Игра окончена", "Победили X/O!" — по завершении игры с победой одного из участников.

"Игра окончена", "Это ничья!"— при ничейном результате.

"Игра окончена", "Выиграл бот!" — при начале новой игры.