МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | Руководство программиста  на лабораторную работу №9  по дисциплине «Алгоритмы и структуры данных»  Тема «Разработка компьютерной игры Крестики Нолики» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21  Вражкин Н.А.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024  **1. Назначение и условия применения программы**  **1.1. Назначение программы**  Программа представляет собой реализацию игры "Крестики-нолики" с графическим интерфейсом. Она разработана на языке Python с использованием библиотеки tkinter. Игрок сражается против компьютера, который использует алгоритм минимакс для принятия оптимальных решений.  **1.2. Функции программы**   * Реализация игры "Крестики-нолики" на поле размером 3x3. * Ходы игрока (крестики) и компьютера (нолики). * Использование алгоритма минимакс для выбора хода компьютера. * Проверка условий победы, ничьей или заполнения игрового поля. * Блокировка кнопок игрового поля после окончания игры. * Функция сброса игры для начала новой партии.   **1.3. Условия применения**  Для корректной работы программы необходимы:   * Python версии 3.x. * Библиотека tkinter, входящая в стандартную библиотеку Python.   **2. Характеристика программы**  **2.1. Общие сведения**  Программа разработана с использованием процедурного подхода. Все функции логически разделены по задачам:   * Обработка ходов игрока и компьютера. * Проверка состояния игры (победа, ничья, заполненность поля). * Управление графическим интерфейсом (кнопки, сообщения, сброс игры).   **2.2. Основные характеристики**   * **Графический интерфейс:** Программа использует библиотеку tkinter для создания интерфейса. Клетки игрового поля представлены кнопками, которые обновляются в зависимости от действий игрока и компьютера. * **Алгоритм минимакс:** Используется для обеспечения оптимальных ходов компьютера, что делает игру сложной для пользователя. * **Гибкость:** Пользователь может перезапустить игру в любой момент.   **3. Обращение к программе**  **3.1. Основные функции программы**   1. **check\_winner(board, player)**    * **Назначение:** Проверяет, есть ли выигрышная комбинация для указанного игрока (X или O).    * **Описание:** Проверяет строки, столбцы и диагонали игрового поля. Возвращает True, если игрок выиграл, иначе False. 2. **is\_full(board)**    * **Назначение:** Проверяет, заполнено ли игровое поле.    * **Описание:** Возвращает True, если все клетки заполнены, иначе False. 3. **minimax(board, depth, is\_maximizing)**    * **Назначение:** Реализует алгоритм минимакс для вычисления лучшего хода компьютера.    * **Описание:** Рекурсивно оценивает все возможные ходы, чтобы найти оптимальное решение. Возвращает числовое значение, которое соответствует результату (1 для победы компьютера, -1 для победы игрока, 0 для ничьей). 4. **computer\_move()**    * **Назначение:** Выполняет ход компьютера.    * **Описание:** Использует алгоритм минимакс для выбора наилучшего хода. После выполнения хода обновляет интерфейс и проверяет условия победы или ничьей. 5. **player\_move(row, col)**    * **Назначение:** Обрабатывает ход игрока.    * **Описание:** Записывает ход игрока в игровое поле, обновляет интерфейс и проверяет условия победы или ничьей. Если игра не окончена, передает ход компьютеру. 6. **disable\_all\_buttons()**    * **Назначение:** Блокирует все кнопки игрового поля.    * **Описание:** Используется после завершения игры для предотвращения дальнейших ходов. 7. **reset\_game()**    * **Назначение:** Сбрасывает игру для начала новой партии.    * **Описание:** Очищает игровое поле, разблокирует кнопки и обновляет интерфейс.   **4. Входные и выходные данные**  **4.1. Входные данные**   * Действия пользователя:   + Нажатие на кнопки игрового поля для выполнения хода.   + Нажатие кнопки "Сброс" для начала новой игры.   **4.2. Выходные данные**   * Визуальное отображение:   + Игровое поле с текущими ходами (X или O).   + Сообщение о результате игры (победа игрока, победа компьютера, ничья).   **5. Сообщения**  Программа отображает следующие сообщения:   * **"Игрок выиграл!"** — если пользователь составил выигрышную комбинацию. * **"Компьютер выиграл!"** — если компьютер составил выигрышную комбинацию. * **"Ничья!"** — если игровое поле полностью заполнено и победителя нет.   **6. Используемые технические средства**  **Библиотеки:**   * **tkinter** — для создания графического интерфейса. * **random** — для генерации случайного выбора (если алгоритм минимакс не используется).   **7. Особенности реализации**   * **Алгоритм минимакс:** Используется для выбора оптимального хода компьютера. Алгоритм анализирует все возможные варианты ходов, чтобы обеспечить максимально сложную игру для пользователя. * **Графический интерфейс:** Реализован на основе виджетов tkinter. Игровое поле представлено кнопками 3x3. Кнопки блокируются после окончания игры. * **Процедурная архитектура:** Программа написана с использованием функций, каждая из которых решает отдельную задачу.   **8. Пример использования**   1. Запустите программу. 2. Выберите клетку для хода, кликнув на соответствующую кнопку (игрок всегда ходит первым). 3. Следите за ходами компьютера, которые появляются автоматически. 4. После завершения игры появится сообщение о результате. 5. Для начала новой игры нажмите кнопку "Сброс". | | | | | | |