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 №10  по дисциплине «Алгоритмы и структуры данных»  Тема «Разработка компьютерной игры "Крестики-Нолики"» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-22  Гончаров Т. Е  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**1. Назначение и условия применения программы**

**1.1. Назначение и функции, выполняемые приложением**

**Назначение:**

Приложение "Крестики-нолики" представляет собой компьютерную реализацию классической настольной игры, известной как "Крестики-нолики". Оно позволяет игрокам (человеку или компьютеру) соревноваться в стратегическом построении своих символов на игровом поле.

**Функции:**

* **Создание новой игры:** Инициализация нового игрового поля 3x3, очистка предыдущих данных игры.
* **Ход игрока:** Игрок выбирает клетку на поле, чтобы разместить свой символ (крестик).
* **Ход компьютера:** Компьютер выбирает клетку для определения наиболее подходящей клетки для размещения символа (нолика).
* **Проверка победы:** Проверяет каждую строку, столбец и диагональ на наличие трех одинаковых символов, состоящих из крестиков или ноликов.
* **Проверка ничьей:** Проверяет, заполнены все клетки на поле символами (крестиками и ноликами).
* **Отображение результата:** Вывод сообщения о победе компьютера или о ничьей.

**1.2. Условия, необходимые для использования приложения**

* **Операционная система:** Windows, macOS, Linux.
* **Язык программирования:** Python.
* **Инструментальная среда:** Любая среда разработки Python (например, PyCharm, Visual Studio Code).
* **Библиотеки:**
  + tkinter (для создания графического интерфейса).

**2. Характеристики программы**

**2.1. Характеристики приложения**

* **Количество строк кода:** Примерно 133 строк кода.
* **Количество структур данных:**
  + **Игровое поле:** Двумерный массив 3x3 для хранения состояния игры.
  + **Текущий игрок:** Переменная для хранения текущего игрока или компьютера (крестик или нолик).
  + **Результат игры:** Переменная для хранения результата (выигрыш, ничья).
* **Количество алгоритмов:**
  + **Алгоритм проверки на выигрыш:** Проверка всех строк, столбцов и диагоналей на наличие выигрышной комбинации (трех символов крестиков или ноликов на игровом поле).
  + **Алгоритм проверки на ничью:** Проверка, заполнены ли все клетки на поле.
  + **Алгоритм хода компьютера:** использован алгоритм minimax, для определения хода компьютера.
  + **Алгоритм отрисовки игрового поля:** Отрисовка игрового поля и символов на экране.
  + **Алгоритмы обработки событий:** Обработка действий пользователя, таких как нажатие кнопок или выбор клетки на поле.
* **Используемые библиотеки:** tkinter (для создания графического интерфейса).
* **Работа приложения:**
  + Игровое поле инициализируется и отображается на экране.
  + Игрок и компьютер делают ходы по очереди, выбирая клетки на поле.
  + После каждого хода проверяется наличие выигрыша или ничьей.
  + В случае победы или ничьей выводится соответствующее сообщение.
  + Игроку предлагается начать новую игру.
* **Внешний вид приложения:** Приложение имеет простой графический интерфейс с полем размером 3 на 3 и кнопки для новой игры.
* **Средства контроля корректности ввода/вывода:**
  + **Проверка выбора клетки:** Проверка, является ли выбранная клетка пустой.
  + **Проверка границ поля:** Проверка, находится ли выбранная клетка в пределах игрового поля (3x3).
  + **Проверка окончания игры:** Проверка, не завершена ли уже игра (выигрыш или ничья).
  + **Обработка ошибок:** Вывод соответствующих сообщений в случае некорректного ввода.

**2.2. Особенности реализации приложения**

* **Структуры данных:**
  + **Двумерный массив:** Используется для представления игрового поля, где каждый элемент массива соответствует клетке на поле и хранит информацию о его состоянии (пусто, крестик, нолик).
  + **Переменные:** Используются для хранения текущего игрока или компьютера и результата игры.
* **Обоснование выбора:**
  + **Двумерный массив:** Удобен для хранения состояния игрового поля, так как обеспечивает доступ к клеткам по индексу.
  + **Переменные:** Необходимы для хранения данных, которые меняются в течение игры.
  + **Список или стек:** Удобны для реализации функций отмены хода, просмотра истории игры, повтора игры или анализа хода.
* **Заимствованные решения:**
  + **Алгоритм хода компьютера:**
    - **Minimax:**

1. Проверка на победу:   
   Алгоритм сначала проверяет, есть ли возможность для компьютера выиграть на следующем ходу. Он перебирает все свободные клетки и ставит в них свой символ "O". Если после этого вызывается функция `self.check\_win()`, которая возвращает `True`, то компьютер делает этот ход и выигрывает. Если нет возможности выиграть, то алгоритм проверяет, есть ли возможность заблокировать победу противника (игрока).
2. Проверка на блокирование:

Аналогично предыдущему пункту, компьютер перебирает все свободные клетки, ставит в них символ противника "X" и проверяет, выигрывает ли противник. Если да, то компьютер ставит "O" в эту клетку, чтобы заблокировать ход противника.

1. Случайный ход:

Если нет ни возможности выиграть, ни заблокировать победу противника, то компьютер делает случайный ход. Это не идеальный алгоритм, но он гарантирует, что компьютер не проиграет, если у игрока нет возможности выиграть..

* + **Графический интерфейс:**
    - Если используются элементы графического интерфейса из сторонних библиотек, то необходимо указать источник этих элементов (библиотека, документация).

**3. Обращение к программе**

* **Методы:**
  + **new\_game():** Инициализация нового игрового поля, выбор режима игры, очистка предыдущих результатов.
  + **make\_move (x, y):** Размещение символа игрока в выбранную клетку двумерного массива.
  + **computer\_move ():** Выбор клетки для размещения символа компьютера.
  + **check\_win ():** Проверка игрового поля на наличие выигрышной комбинации (в ряд построенных 3 символа из крестиков или ноликов)
  + **check\_draw ():** Проверка, заполнены ли все клетки на поле без выигрышной комбинации.
  + **show\_result (message):** Вывод сообщения о победе компьютера или ничьей.

**4. Сообщения**

По результатам выполнения кода могут быть следующие результаты:

1. Победа компьютера, если собраны 3 в ряд символа.
2. Ничья, если игровое поле занято, но не собраны 3 в ряд символа.