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*** |  | |  | | ТЕХНИЧЕСКОЕ ЗАДАНИЕ  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 | | | | | | |

**Введение**

**Наименование приложения:** Компьютерная игра "Крестики-нолики"

Приложение представляет собой реализацию классической игры "Крестики-нолики" для двух игроков. Игра ведется на поле 3x3, игроки по очереди ставят свои символы (крестик или нолик) в пустые клетки. Цель игры - построить ряд из трех символов (крестиков или ноликов) по горизонтали, вертикали или диагонали.

**Функциональные возможности приложения:**

* **Создание новой игры:** Инициализация игрового поля с пустыми клетками.
* **Ход игрока:** Выбор игроком клетки на поле для размещения символа (крестика).
* **Ход компьютера:** В случае игры против компьютера, выбор компьютером клетки для размещения символа (нолика).
* **Проверка победы:** Проверка после каждого хода на наличие построенных в ряд 3 символа (нолика) для компьютера.
* **Проверка ничьей:** Проверка после каждого хода на заполнение всех клеток без построения в ряд 3 символа (нолика).
* **Отображение результата:** Отображение сообщения о победе или ничьей.
* **Возможность повтора игры:** Предложение начать новую игру после завершения текущей.

**1. Основания для разработки**

Учебная программа по дисциплине «Алгоритмы и структуры данных» по специальности 09.03.02 «Информационные системы и технологии».

**2. Требования к программе**

**2.1. Функциональное назначение**

Компьютерная игра "Крестики-нолики" предназначена для реализации классической настольной игры "Крестики-нолики" с возможностью игры против искусственного интеллекта.

Автоматизируемые процессы в игре "Крестики-нолики"

1. Инициализация игры:

* Создание игрового поля:
  + Создание пустого массива 3x3, представляющего игровое поле.
  + Заполнение массива пустыми значениями (например, " "), представляющими пустые клетки.

1. Ход игрока:

* Выбор клетки:
  + Пользователь выбирает клетку на поле, кликая по ней.
* Проверка:
  + Клетка должна быть пустой.
  + Клетка должна находиться в пределах игрового поля (от 0 до 2).
* Размещение символа:
  + В выбранную клетку размещается символ игрока (крестик).
  + Массив игрового поля обновляется и записывается символ (крестик), выбранным игроком клетку.
  + После успешного размещения символа переключается ход компьютеру.

1. Ход компьютера:

* Выбор клетки:
  + Использование алгоритма для выбора клетки.
* Алгоритм должен учитывать:
  + Доступность клетки.
  + Возможность выиграть в следующий ход.
  + Возможность предотвратить выигрыш противника в следующий ход.
  + Различные стратегии, такие как: выбор центральной клетки, если она свободна, или выбор угловых клеток.
* Размещение символа:
  + В выбранную клетку размещается символ компьютера.
  + Массив игрового поля обновляется и записывается символ (нолик), выбранным компьютеру клетку.

1. Проверка на победу:

* Проверка строк:
  + Проверка каждой строки на наличие трех одинаковых символов (крестиков или ноликов).
* Проверка столбцов:
  + Проверка каждого столбца на наличие трех одинаковых символов (крестиков или ноликов).
* Проверка диагоналей:
  + Проверка главной и побочной диагоналей на наличие трех одинаковых символов (крестиков или ноликов).

1. Проверка на ничью:

* Проверка заполнения поля:
  + Проверка, заполнены все клетки на поле.

1. Отображение результата:

* Выведение сообщения о победе:
  + Если победил компьютер, то вывести сообщение о его победе, например, "Компьютер победил!".
* Выведение сообщения о ничьей:
  + Если ничья, вывести сообщение о ничьей, например, "Ничья!".

1. Начало новой игры:

* Очистка игрового поля:
  + Заполнить все клетки массива игрового поля пустыми значениями.  
    Группы пользователей: студенты, занимающиеся изучением разработкой игр.

**2.2. Требования к функциональным характеристикам**

**2.2.1 Требования к структуре приложения**

Приложение будет состоять из следующих модулей:

* **Модуль "Игровое поле":**
  + Создание игрового поля 3x3.
  + Отображение текущего состояния поля (пустые клетки, символы игрока и компьютера).
  + Обновление состояния поля после хода игрока и компьютера.
* **Модуль "Ход игрока":**
  + Принимает выбор игрока (клик по клетке).
  + Проверяет корректность выбора (клетка должна быть пуста).
  + Размещает символ игрока в выбранной клетке.
* **Модуль "Ход компьютера":**
  + Выбирает клетку для размещения своего символа.
  + Проверяет корректность выбора (клетка должна быть пуста).
  + Размещает символ компьютера в выбранной клетке.
* **Модуль "Проверка победы":**
  + Проверяет после каждого хода на наличие выигрышной комбинации (3 символа в ряд – нолика или крестика).
  + Возвращает результат: True (есть победа) или False (победы нет).
* **Модуль "Проверка ничьей":**
  + Проверяет после каждого хода на заполнение всех клеток без выигрышной комбинации.
  + Возвращает результат: True (ничья) или False (ничьей нет).
* **Модуль "Отображение результата":**
  + Отображает сообщение о победе или ничьей.
* **Модуль "Новая игра":**
  + Очищает игровое поле.
  + Начинают новую игру по нажатию кнопки “Новая игра”.

**2.2.2 Требования к составу функций приложения**

* **Функции для игрока:**
  + new\_game(): Начать новую игру.
  + make\_move(x, y): Сделать ход, разместив свой символ в клетку с координатами (x, y).
* **Функции для компьютера:**
  + computer\_move(): Сделать ход, разместив свой символ в клетку, выбранную компьютером.
* **Функции для проверки игры:**
  + check\_win(): Проверить, есть ли победа.
  + check\_draw(): Проверить, есть ли ничья.
* **Функции для вывода результата:**
  + show\_result(): Вывод результат игры – победа компьютера или ничья.

**2.2.3 Требования к организации информационного обеспечения**

* **Пользовательский интерфейс:**
  + Графический интерфейс с визуальным отображением игрового поля 3x3.
  + Кнопка "Новая игра".
  + Поле для вывода сообщения о победе или ничьей.
* **Файлы:**
  + Нет необходимости в сохранении данных в файлы.

**2.3. Требования к надежности**

Приложение должно быть стабильным и не должно вызывать ошибок во время работы.

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

* **Операционная система:** Windows (любая современная версия).
* **Язык программирования:** Python.
* **Среда разработки:** Любая среда разработки Python (например, PyCharm, Visual Studio Code).
* **Библиотеки:**
  + tkinter (для создания графического интерфейса).

**2.5. Требования к маркировке и упаковке**

Не предъявляются.

**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**

Программная документация должна включать:

* Техническое задание.
* Пояснительную записку.
* Исходный код приложения.
* Отчет о тестировании.

**4. Стадии и этапы разработки**

Разработка приложения будет проходить в несколько этапов:

* **Этап 1:** Проектирование. Разработка архитектуры приложения, определение основных модулей и функций.
* **Этап 2:** Реализация. Написание кода приложения, реализация графического интерфейса.
* **Этап 3:** Тестирование. Проверка функциональности приложения, выявление и устранение ошибок.
* **Этап 4:** Документирование. Подготовка документации по разработанному приложению.

**5. Порядок контроля и приемки**

* **Контроль:** Контроль качества приложения осуществляется преподавателем в ходе выполнения работы.
* **Приемка:** Приемка приложения осуществляется преподавателем после завершения всех этапов разработки.