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 №8  по дисциплине «Алгоритмы и структуры данных»  Тема «Управление договорами на аренду автомобилей» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21  Пирогов П.Е.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**Введение**

Целью данной лабораторной работы является создание графического приложения для игры в "Крестики-нолики" с использованием языка программирования Python и библиотеки tkinter. Программа должна предоставить пользователю два режима игры: "Игрок против игрока" и "Игрок против компьютера", при этом компьютер будет использовать искусственный интеллект для принятия решений и выбора оптимального хода.

**Основания для разработки**

Данный проект разработан в рамках учебного плана по направлению "Информационные системы и технологии".

**Основные требования к функциональности**

**1. Основные функции приложения:**

* Два режима игры:
  + Игрок против игрока.
  + Игрок против компьютера (с применением алгоритма Minimax для оптимизации выбора хода компьютера).
* Возможность сброса текущей игры и начала новой партии.
* Отображение текущего игрока ("X" или "O").
* Оповещение о победе одного из игроков или ничьей.

**2. Организация пользовательского интерфейса:**

* Игровое поле размером 3x3, представлено кнопками, каждая из которых соответствует ячейке на поле.
* Кнопки для выбора режима игры: "Игрок против Игрока" и "Игрок против Компьютера".
* Кнопка для сброса игры и старта новой партии.
* Информирование пользователя о результате игры (победа, проигрыш или ничья) через всплывающие окна.

**3. Взаимодействие с пользователем:**

* Кнопки игрового поля будут интерактивными, что позволит игрокам делать ходы по клику.
* При выборе ячейки на игровом поле будет выполняться ход игрока.
* При игре против компьютера, после хода игрока, программа автоматически выполнит ход компьютера.

**Требования к программной реализации**

**1. Язык программирования и библиотеки:**

* Программа должна быть реализована на языке Python версии 3.8 или выше.
* Для создания графического интерфейса будет использоваться встроенная библиотека tkinter.

**2. Логика игры:**

* После каждого хода необходимо проверять состояние поля, чтобы определить победителя или ничью.
* Для вычисления хода компьютера будет применяться алгоритм Minimax, который оптимизирует выбор позиции.

**Требования к надежности**

* Программа должна корректно обрабатывать возможные ошибки (например, предотвращать попытки сделать ход в уже занятую ячейку).
* После окончания игры (победа, проигрыш или ничья) программа должна блокировать возможность дальнейших ходов до начала новой партии.

**Состав и параметры технических средств**

Программа должна работать на устройствах с установленным Python версии 3.8 и выше.

**Этапы разработки**

1. **Проектирование интерфейса**: Разработка структуры окна и размещение элементов.
2. **Разработка основной логики игры**: Реализация функций для обработки ходов игроков и проверки победы или ничьей.
3. **Реализация алгоритма Minimax**: Создание функции для вычисления оптимального хода компьютера.
4. **Тестирование**: Проверка всех функций программы, включая взаимодействие с интерфейсом и логику игры.

**Порядок контроля и приемки**

Контроль осуществляется через тестирование функциональности программы, проверку корректности интерфейса и точность логики игры.