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  Камчаров М.С.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**Техническое задание.**

**Техническое задание на разработку программы "Крестики-нолики".**

**1. Наименование**

Программное обеспечение игры "Крестики-нолики".

**2. Назначение**

Разработка программного обеспечения игры "Крестики-нолики" для двух игроков, один из которых - искусственный интеллект.

**3. Функциональные требования**

* Игровой процесс:
* Игра должна быть реализована на основе классических правил "Крестики-нолики" для поля 3x3.
* Игрок "X" делает первый ход.
* Игрок "O" (искусственный интеллект) делает ходы автоматически после каждого хода игрока "X".
* Программа должна определять победителя (первый, кто построил ряд из трех своих символов по горизонтали, вертикали или диагонали) или ничью (если все клетки заполнены, а победителя нет).
* Интерфейс:
* Программа должна иметь графический интерфейс, реализованный с помощью библиотеки Tkinter.
* Интерфейс должен содержать игровое поле с 9 кнопками для размещения символов "X" и "O".
* Интерфейс должен отображать информацию о текущем игроке.
* Интерфейс должен отображать результат игры (победа, ничья).
* Интерфейс должен иметь кнопку "Сброс" для начала новой игры.
* Искусственный интеллект:
* Искусственный интеллект должен реализовывать алгоритм Minimax для определения оптимального хода.
* Алгоритм Minimax должен быть реализован таким образом, чтобы ИИ мог играть на уровне сложности, сопоставимом с человеком (возможность выбора уровня сложности: простой, средний, сложный).

**4. Нефункциональные требования**

* Надежность:
* Программа должна быть устойчива к ошибкам пользователя.
* Программа не должна зависать или выдавать ошибки при некорректном вводе пользователя (например, при нажатии на уже занятую клетку).
* Программа должна иметь механизм обработки исключительных ситуаций.
* Эффективность:
* Программа должна работать быстро и без задержек, особенно при выборе хода ИИ.
* Алгоритм Minimax должен быть оптимизирован для быстрого поиска хода.
* Переносимость:
* Программа должна быть совместима с операционной системой Windows.
* Пользовательский интерфейс:
* Интерфейс должен быть интуитивно понятным и удобным для пользователя.
* Интерфейс должен быть визуально привлекательным и соответствовать стандартам дизайна.

**5. Требования к документации**

* Программа должна быть снабжена комментариями, поясняющими логику работы кода.

**6. Сроки выполнения**

* Срок выполнения: 1 неделя

**7. Ответственность**

* Ответственный за разработку: Камчаров Максим Сергеевич
* Заказчик: Шишкин Вадим Викторинович