**Введение**

Настоящее техническое задание описывает разработку приложения “Крестики-нолики с ИИ”. Приложение реализует классическую игру “Крестики-нолики” с возможностью игры против искусственного интеллекта.

**Правила игры:** Два игрока по очереди ставят на игровом поле 3x3 знаки (крестики или нолики). Выигрывает игрок, первым выстроивший три своих знака в ряд (горизонтально, вертикально или по диагонали). Если все клетки заполнены, а ни один игрок не победил, объявляется ничья.

Приложение предоставляет следующие функциональные возможности:

* Игра против искусственного интеллекта, использующего алгоритм Minimax с альфа-бета отсечением.
* Графический интерфейс пользователя (GUI) для интерактивной игры.
* Проверка корректности хода игрока и информирование об ошибках.
* Отображение текущего игрока, результата игры и хода игры.
* Возможность начать новую игру.

**1. Основания для разработки**

Учебный план направления 09.03.02 «Информационные системы и технологии»

**2. Требования к программе или программному изделию**

**2.1. Функциональное назначение**

Программа “Крестики-нолики с ИИ” предназначена для игры в классические “Крестики-нолики” против искусственного интеллекта. Программа автоматизирует процесс генерации ходов ИИ, проверки условий победы/поражения/ничьей, и отображения игрового процесса. Целевыми пользователями являются студенты, изучающие программирование ИИ, и все желающие поиграть в “Крестики-нолики” с сильным оппонентом.

**2.2. Требования к функциональным характеристикам**

2.2.1 Требования к структуре приложения

Программа имеет модульную структуру.

Модуль графического интерфейса: отвечает за визуальное отображение игрового поля и взаимодействие с пользователем.

Модуль игровой логики: реализует правила игры “Крестики-нолики”, определяет победителя и ничью.

Модуль ИИ: содержит алгоритм Minimax с альфа-бета-отсечением для генерации ходов ИИ.

2.2.2 Требования к составу функций приложения

Инициализация игры: Создание нового игрового поля. Выбор пользователя (X или O).

2.2.3 Требования к организации информационного обеспечения, входных и выходных данных

Пользовательский интерфейс: Должен быть интуитивным и удобным для использования. Игровое поле должно быть визуально понятно. Отображение текущего игрока, результатов игры и сообщений об ошибках.

Входные данные: Выбор пользователя (X или O), координаты хода игрока (строка, столбец).

Выходные данные: Визуальное отображение игрового поля, сообщения о ходе игры, результат игры.

Обработка хода человека: Прием координат хода от игрока, проверка корректности, обновление игрового поля.

Генерация хода ИИ: Использование алгоритма Minimax для выбора оптимального хода.

Проверка окончания игры: Определение победы, поражения или ничьей после каждого хода.

Отображение результатов: Информирование пользователя о результате игры (победа, поражение, ничья) на игровом поле или в отдельном окне.

Перезапуск игры: Опция для начала новой игры.

**2.3. Требования к надежности**

Программа должна быть устойчива к ошибкам ввода пользователя, обрабатывая некорректный ввод с соответствующими сообщениями. Программа не должна вызывать сбои в системе.

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

* Версия операционной системы: Windows 10
* Используемые библиотеки: tkinter
* Язык Python: 3.12.5
* Используемая среда разработки: Visual Studio Code

**2.5. Требования к маркировке и упаковке**

Определяются заданием на лабораторную работу

**2.6. Требования к транспортированию и хранению**

2.6.1 Условия транспортирования

Требования к условиям транспортирования не предъявляются

2.6.2 Условия хранения

Требования к условиям транспортирования не предъявляются

2.6.3 Сроки хранения

Срок хранения – до окончания срока учебы

**3. Требования к программной документации**

Определяются заданием на лабораторную работу

**4. Стадии и этапы разработки**

Определяются заданием на лабораторную работу

**5. Порядок контроля и приемки**

Определяются заданием на лабораторную работу