1. **Структуры данных**

Основные структуры данных, используемые в игре "Крестики-нолики", включают:

**Игровое поле (self.board):**

- Двумерный список 3x3

- Каждая ячейка может содержать:

  "" - пустая клетка

  "X" - крестик

  "O" - нолик

Пример состояния поля:

[

    ["X", "", "O"],

    ["", "X", ""],

    ["O", "", ""]

]

Где:

* ""— пустая клетка,
* 'X' — клетка, занятая крестиком,
* 'O' — клетка, занятая ноликом.
* **Игроки**: Игроки представлены в виде строковых значений, обозначающих, какой символ каждый игрок использует.

player1 = 'X'

player2 = 'O'

* **Очередь хода**: Переменная, отвечающая за текущего игрока, будет меняться после каждого хода.

current\_player = player1

1. **Алгоритмы**

Для реализации механики игры потребуются следующие ключевые алгоритмы:

**2.1. Алгоритм хода игрока**

Каждый игрок может поставить свой символ ('X' или 'O') в одну из пустых клеток 2.1 Алгоритм хода игрока (make\_move):

1. Проверка возможности хода

2. Установка символа игрока

3. Проверка победы/ничьей

4. Смена текущего игрока

**2.2 Алгоритм ИИ (minimax):**

1. Оценка текущей позиции

2. Рекурсивный перебор возможных ходов

3. Выбор оптимального хода

**2.3 Алгоритм проверки победы:**

- Проверка всех строк

- Проверка всех столбцов

- Проверка диагоналей

**2.4 Алгоритм проверки ничьей:**

- Проверка заполненности всех клеток

1. **Дополнительные элементы модели**

* **Графический интерфейс**: Для отображения игры может быть использован простой графический интерфейс (например, с помощью библиотеки tkinter). Игровое поле будет отображаться в виде кнопок, на которые игроки могут нажимать, чтобы сделать ход.
* **Отображение статуса игры**: На экране будет отображаться информация о текущем игроке, победителе или ничьей.

1. **Итеративность проектирования**

Проектирование игры может включать несколько этапов тестирования и доработок. На каждом этапе могут быть добавлены новые функции (например, возможность отменить ход, подсказки для игрока), а также оптимизированы существующие алгоритмы.

Модель будет дополняться по мере выявления потребностей в ходе тестирования и разработки новых игровых механик.