**Руководство программиста.**

**Руководство программиста на разработку программы "Крестики-нолики".**

**1. Общее описание**

Данное руководство предназначено для программистов, работающих с кодом игры "Крестики-нолики", реализованной на языке Python с использованием библиотеки Tkinter. Руководство содержит информацию о структуре программы, алгоритмах, используемых в ней, и описание ключевых функций.

**2. Структура программы**

Программа состоит из одного файла, содержащего класс TicTacToe, который реализует логику игры и графический интерфейс.

* Класс TicTacToe:
  + init(self, master): Инициализирует графический интерфейс игры, создавая игровое поле, кнопки для хода, индикатор текущего игрока и кнопку "Сброс".
* Создает матрицу self.board 3x3, которая хранит состояние игрового поля.
* Инициализирует переменные self.current\_player (текущий игрок), self.game\_over (флаг окончания игры).
* Создает матрицу self.buttons 3x3, где каждый элемент - это кнопка Tkinter, представляющая ячейку на игровом поле.
* Создает метку self.label для отображения хода текущего игрока.
* Создает кнопку "Сброс" для начала новой игры.
  + click\_button(self, row, col): Обрабатывает ход пользователя.
* Проверяет, не закончилась ли игра.
* Проверяет, пуста ли выбранная ячейка.
* Ставит знак текущего игрока на выбранную ячейку.
* Обновляет текст кнопки, соответствующей ячейке.
* Проверяет победу текущего игрока.
* Проверяет ничью.
* Переключает ход на другого игрока.
* Вызывает функцию bot\_move(), если ход перешел к компьютеру.
  + bot\_move(self): Реализует ход компьютера с использованием алгоритма Minimax.
* Ищет оптимальный ход на основе рекурсивного алгоритма Minimax, который оценивает все возможные ходы и выбирает наиболее выгодный для компьютера.
* Вызывает функцию click\_button(), чтобы сделать выбранный ход на игровом поле.
  + minimax(self, board, is\_maximizing\_player): Рекурсивная функция Minimax, которая находит оптимальный ход для компьютера.
* Принимает на вход состояние игрового поля board и флаг is\_maximizing\_player, указывающий, является ли текущий ход максимизирующим (компьютер) или минимизирующим (игрок).
* Рекурсивно просматривает все возможные ходы, оценивая их с помощью функции check\_win().
* Возвращает оценку хода, которая позволяет определить, насколько он выгодный для компьютера.
  + check\_win(self, player): Проверяет, победил ли игрок player.
* Проверяет все возможные линии (горизонтальные, вертикальные и диагональные) на наличие трех знаков игрока.
* Возвращает True, если игрок победил, иначе False.
  + check\_draw(self): Проверяет, закончилась ли игра ничьей.
* Проверяет, есть ли свободные ячейки на игровом поле.
* Возвращает True, если все ячейки заняты, иначе False.
  + show\_winner(self, winner): Отображает сообщение о победителе или ничьей с помощью диалогового окна.
  + reset\_game(self): Сбрасывает игру, очищая игровое поле и устанавливая начальные значения переменных.

**3. Алгоритмы**

Алгоритм Minimax:

* + Алгоритм Minimax используется для принятия решения о ходе компьютера.
  + Алгоритм рекурсивно просматривает все возможные ходы на игровом поле, оценивая их с точки зрения компьютера.
  + Каждый ход оценивается по тому, насколько он приближает компьютера к победе или насколько он ухудшает положение игрока.
  + Алгоритм выбирает ход с наибольшей оценкой для компьютера, то есть наиболее выгодный для него.

**4. Рекомендации программисту**

* + Программа хорошо структурирована и документирована, что упрощает ее модификацию и отладку.
  + Алгоритм Minimax реализован с использованием рекурсии, что обеспечивает его эффективность.
  + Графический интерфейс простой и интуитивно понятный, что делает программу удобной в использовании.

**Заключение**

Данная программа предоставляет пользователям возможность насладиться классической игрой "Крестики-нолики", и благодаря использованию алгоритма Minimax, компьютер способен предоставлять достойного соперника. Программа может быть расширена дополнительными функциями, такими как выбор уровня сложности и применение различных стратегий игры.