МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное

учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

Лабораторная работа № 10

По дисциплине «Алгоритмы и структуры данных»

Тема программа «Крестики-нолики»

Руководство программиста

Р.02069337.№23/690-№20 РП-01

Листов 8

Исполнитель:

студент гр. ИСТбд-22

Минибаева Е.Р.

«»_____2024 г.

2024 г.

1. Обзор программы

Программа представляет собой графический интерфейс (GUI) для игры в крестики-нолики с компьютером. Она позволяет:

- Визуально представлять состояние игрового поля в графическом виде.
- Реализовывать алгоритм игры для компьютера: В режиме игры против компьютера, программа должна использовать алгоритм, позволяющий компьютеру делать ходы, стремясь к выигрышу или, как минимум, к ничьей.
- Вводить ходы: Возможность пользователю делать ходы, устанавливая свой символ "X" на свободные клетки игрового поля.
- Предоставлять информацию о результате игры: Оповещать пользователя о победе, поражении или ничьей.

2. Структура программы

Программа состоит из одного класса:

1. TicTacToe: Представляет собой полноценную программу крестики-нолики.

Имеет следующие атрибуты:

- master: Родительский виджет.

Имеет следующие методы:

- __init__(self, master): создает GUI
- start_game(): Запускает закрытие меню, запускает создание игровой доски.
- hide_menu(self): Закрывет меню
- create_game_board(self): Создает игровую доску, создает кнопку назад
- back_to_menu(self): Закрывает окно игрового процесса, запускает показ меню
- show_menu(self): Показывает меню
- make_move(self, x, y): обрабатывает логику ходов, запускает проверку на победу компьютера, запускает проверку на ничью.
- check_winner(self, player): Проверяет состояние игрового поля на наличие победной ситуации.
- is_draw(self): Проверяет состояние игрового поля на ничью.
- reset_game(self):Обнуляет состояние игрового поля.
- check_line(self, sum_O, sum_X): Проверяет линию на количество занятых клеток, если они соответствуют входным значениям возвращается значение координаты поля.

- AI(self): Выявляет с помощью функции check_line() лучший в данной ситуации ход и выполняет его.

3. Алгоритм работы программы

1. Инициализация:

- При запуске программы создается экземпляр класса TicTacToe
- TicTacToe создает GUI программы.

2. Взаимодействие пользователя:

- Пользователь может нажать на кнопки "Против компьютера" и "Выйти из игры".
- При нажатии на кнопку "Выйти из игры" приложение закрывается.
- При нажатии на кнопку "Против компьютера" меню закрывается, открывается окно игрового процесса, на котором отображается игровое поле и кнопка "Назад".
- При нажатии на кнопку "Назад" окно игрового процесса закрывается и открывается меню.
- Пользователь может нажать на любую клетку игрового поля в начале игры, далее только на пустые клетки.

3. Обработка данных:

- При нажатии на любую из кнопок на игровом поле (представляющих собой клетки игрового поля) вызывается метод make_move(x, y), где x и y - координаты нажатой клетки.
- Метод make_move(x, y):
    - Проверяет, является ли клетка свободной.
    - Обновляет текст кнопки (ставит “X” или “O”).
    - Обновляет состояние игрового поля в списке maps.
    - Вызывает метод check_winner(player) для проверки наличия победителя после каждого хода.
    - Вызывает метод is_draw() для проверки ничьи.
    - В случае, если победителя или ничьи нет, переключает текущего игрока (с “X” на “O” и наоборот) и вызывает метод AI() для хода компьютера (если текущий игрок “O”).
- Метод AI() (ходит ИИ):
    - Вызывает метод check_line, чтобы определить наилучший ход для ИИ, блокировать игрока или занять центр.

- После выбора хода, вызывает make_move(x, y) для его выполнения.

- Метод check_winner(player):
  - Проверяет, есть ли выигрышная комбинация для указанного игрока (“X” или “O”).
- Метод is_draw():
  - Проверяет, заполнены ли все клетки игрового поля, что означает ничью.

3. Отображение результатов:

- **Ходы игроков:**
  - Результат ходов отображается непосредственно на игровом поле.
  - Каждая нажатая клетка отображает “X” или “O” соответствующего игрока (с разным цветом).
- **Сообщение о результате:**
  - Когда игра заканчивается (есть победитель или ничья) отображается сообщение в виде диалогового окна:
  - Если есть победитель: диалоговое окно с текстом “Игрок X победил!” или “Игрок O победил!”
  - Если ничья: диалоговое окно с текстом “Игра закончилась вничью!”.
  - После отображения сообщения вызывается метод reset_game(), который сбрасывает состояние игрового поля для начала новой игры.
- **Возврат в главное меню:**
  - После завершения игры или в любой момент игры, пользователь может нажать кнопку “Назад”, чтобы вернуться в главное меню.
  - Метод back_to_menu() скрывает игровое поле и вызывает метод show_menu(), чтобы отобразить главное меню.

4. Описание кода

```
from tkinter import *
from tkinter.messagebox import showinfo

class TicTacToe:

    def __init__(self, master):
        # ... (код класса TicTacToe)
    def start_game(self):
        # ... (код метода start_game)
```

```
    def hide_menu(self):
        # ... (код метода hide_menu)
    def create_game_board(self):
        # ... (код метода create_game_board)
    def back_to_menu(self):
       # ... (код метода back_to_menu)
    def show_menu(self):
        # ... (код метода show_menu)
    def make_move(self, x, y):
        # ... (код метода make_move)
    def check_winner(self, player):
         # ... (код метода check_winner)
    def is_draw(self):
         # ... (код метода is_draw)
    def reset_game(self):
         # ... (код метода reset_game)
    def check_line(self, sum_O, sum_X):
         # ... (код метода check_line)
    def AI(self):
        # ... (код метода AI)


if __name__ == "__main__":
    root = Tk()
    app = TicTacToe(root)
    root.mainloop()
```

1. Класс TicTacToe:

- Метод __init__(self, master):
    - Инициализирует главное окно игры master.
    - Устанавливает заголовок окна “Крестики-нолики”.
    - Устанавливает размеры и фон окна.
    - Создает заголовок игры “TicTacToe”.
    - Создает кнопки “Против Компьютера” и “Выйти из игры” для главного меню.
    - Инициализирует текущего игрока (current_player) как “X” и список (maps) для хранения состояния игрового поля.

- Метод start_game(self):
    - Скрывает главное меню и вызывает метод для создания игрового поля.
- Метод hide_menu(self):
    - Скрывает заголовок игры и кнопки главного меню.
- Метод create_game_board(self):
    - Создает фрейм для размещения игрового поля.
    - Создает двумерный список кнопок (buttons) для представления клеток поля.
    - Добавляет обработчик событий для каждой кнопки (метод make_move).
    - Создает кнопку “Назад” для возврата в меню.
- Метод back_to_menu(self):
    - Скрывает игровое поле и кнопку "Назад".
    - Вызывает метод для отображения главного меню.
- Метод show_menu(self):
    - Отображает заголовок игры и кнопки главного меню.
- Метод make_move(self, x, y):
    - Обрабатывает ход игрока на клетке с координатами (x, y).
    - Обновляет текст кнопки текущим игроком (“X” или “O”) и меняет цвет текста
    - Обновляет состояние игрового поля в списке maps.
    - Проверяет победителя или ничью с помощью методов check_winner и is_draw.
    - Если игра не закончена, переключает текущего игрока.
    - Если текущий игрок “O” вызывает ход ИИ.
- Метод check_winner(self, player):
    - Проверяет, есть ли победитель для заданного игрока (player), проверяя все выигрышные комбинации (строки, столбцы, диагонали).
    - Возвращает True, если победитель есть, False в противном случае.
- Метод is_draw(self):
    - Проверяет, есть ли ничья, проверяя все клетки, заполнены ли они.
    - Возвращает True, если ничья, False в противном случае.
- Метод reset_game(self):
    - Сбрасывает состояние игрового поля, очищая текст всех кнопок и обнуляя список maps.
    - Устанавливает текущего игрока как “X”.

- Метод check_line(self, sum_O, sum_X):
    - Проверяет есть ли на игровом поле линия где определенное кол-во “O” или “X” и возвращает индекс нужного хода
- Метод AI(self):
    - Реализует логику хода ИИ, используя метод check_line для поиска выигрышного хода или блокирования хода игрока.
    - Использует центр клетки для приоритетного хода.
    - Если нет приоритетных ходов, выбирает случайную клетку.
    - Вызывает метод make_move для хода ИИ.

2.**Основной блок (if __name__ == "__main__":)**

- Создает экземпляр окна Tkinter (root).
- Создает экземпляр класса TicTacToe (app).
- Запускает основной цикл обработки событий Tkinter (root.mainloop()), что делает окно интерактивным.

5. Рекомендации по использованию

- Запуск игры:
    - Запустите исполняемый файл программы “Крестики-Нолики”.
- Главное меню:
    - В главном меню вам доступны две кнопки: “Против Компьютера” и “Выйти из игры”.
    - Нажмите кнопку “Против Компьютера”, чтобы начать игру против компьютерного противника.
    - Нажмите кнопку “Выйти из игры”, чтобы закрыть программу.
- Игровой процесс:
    - После нажатия кнопки “Против Компьютера” отобразится игровое поле 3x3.
    - Вы, как игрок “X”, делаете первый ход, нажимая на любую свободную клетку на поле.
    - Компьютерный противник (игрок “O”) делает ответный ход.
    - Продолжайте поочередно делать ходы, нажимая на свободные клетки, пока один из игроков не выиграет, или не наступит ничья.
    - Победитель определяется, если один из игроков собрал три своих символа (“X” или “O”) в ряд (горизонтально, вертикально или по диагонали).
    - Ничья наступает, если все клетки на поле заполнены, и никто не выиграл.
- Результат игры:

- После окончания игры (победа или ничья) отобразится всплывающее окно с сообщением о результате.
- Нажмите “OK” в сообщении, чтобы закрыть окно.
- Возврат в главное меню:
- В любое время, вы можете вернуться в главное меню нажав кнопку “Назад”

- Начало новой игры:
  - Чтобы начать новую игру, вернитесь в главное меню и снова нажмите кнопку “Против Компьютера”.

6. Дополнительные возможности

- Реализовать выбор уровня сложности для ИИ.
- Реализовать возможность игры двух игроков (локально).
- Добавить возможность выбора стороны (X или O) для игрока.
- Добавить сохранение статистики.

7. Тестирование

- Тестирование программы должно включать:
  - Проверку корректности отображения элементов интерфейса.
  - Проверку в возможности совершения ходов игроком и ИИ.
  - Проверку перехода между главным меню и игровым полем.
  - Проверку логики работы AI.
  - Проверку правильности определения победы или ничьей.
  - Проверку отображения сообщений о результатах игры.

8. Документация

- Данное руководство программиста является основным документом по программе.
- Дополнительную документацию можно добавить в виде комментариев к коду.
- Важно использовать стандартные соглашения по наименованию переменных и функций.
- Документация должна быть четкой, понятной и доступной для пользователя.

9. Замечания

- Программа использует библиотеку tkinter для создания GUI.

- Программа использует библиотеку messagebox – для необходимых сообщений пользователю.

10. Дополнительные замечания

- Программа может быть расширена и улучшена.
- Программа может быть использована как основа для создания более сложных приложений.
- Важно следовать принципам модульности, повторного использования кода и ясности написания кода.