ТЕМА ПРАКТИЧНОЇ РОБОТИ:
РОЗРОБКА ГРИ "КРЕСТИКИ-НОЛІКИ" З ВИКОРИСТАННЯМ АЛГОРИТМУ "MINIMAX"

Опис реалізованих функцій в файлі TicTacToe.py

Назва методу	Детальне пояснення
__init__	Конструктор класу `TicTacToeGame`, ініціалізує параметри гри, такі як розмір матриці, розмір гри, координати та кольори. Обирає гравця, який починає гру випадковим чином у режимі 3 або залишає його невизначеним. Встановлює початковий стан гри в залежності від обраного режиму.
set_pen	Конфігурує олівець для малювання у грі (встановлює колір, товщину, швидкість та приховує відображення олівця).
set_window	Конфігурує вікно для гри (встановлює розмір, заголовок та фон).
draw_net	Малює сітку для гри (вертикальні та горизонтальні лінії) за допомогою Turtle.
click	Обробляє клік гравця у режимі Human-Human, визначає рядок та стовпчик кліку та оновлює гру.
count_none_elements	Підраховує кількість пустих елементів у грі.
update_table	Оновлює гру після ходу гравця, встановлює символ поточного гравця на дошці, оновлює таблицю та змінює чергу гравця у режимі 3.
final_screen	Виводить екран переможця, якщо є переможець, та очищує екран.
check	Перевіряє, чи є переможець у грі, перевіряючи рядки, стовпці, головну та вторинну діагоналі. Повертає символ переможця або 'ніхто', якщо немає переможця.
(main part of the script)	Головна частина скрипта, яка ініціалізує об'єкт гри, налаштовує вікно та олівець, малює сітку та ініціалізує гравців в залежності від режиму гри. Виконує головний цикл гри.

Опис реалізованих функцій в файлі Players.py

Назва методу	Детальне пояснення
__init__	Конструктор класу `Player`. Ініціалізує об'єкт гравця з вказаним символом (`turn`), який визначає, чи це "хрестик" ('x') чи "нулик" ('o').
pass_move	Базовий метод, який дозволяє гравцеві зробити хід у грі. У класі `Player` цей метод не реалізований (просто `pass`), оскільки само тіло методу буде визначатися у підкласах (`HumanPlayer`, `Computer`).
__init__	Конструктор класу `HumanPlayer`. Викликає конструктор батьківського класу `Player` і передає йому символ гравця ('x' або 'o').
pass_move	Метод, який визначає хід гравця у режимі Human. Встановлює символ поточного гравця та очікує кліку на вікні гри.
__init__	Конструктор класу `Computer`. Викликає конструктор батьківського класу `Player` і передає йому символ гравця ('x' або 'o').

Назва методу	Детальне пояснення
pass_move	Метод, який визначає хід комп'ютера. Якщо це перший хід, комп'ютер обирає випадкову позицію. В іншому випадку, використовує алгоритм minimax для визначення оптимального ходу.
minimax	Алгоритм minimax для визначення оптимального ходу комп'ютера. Рекурсивно оцінює всі можливі ходи та вибирає найкращий хід. Враховує максимізацію чи мінімізацію гравця.

Опис алгоритму

Ініціалізація гравців та стану гри:

- Клас Computer успадковує від класу Player і має метод pass_move, який визначає хід комп'ютера в грі.
- У методі minimax, спочатку визначається, хто максимізує виграш (max_one) і хто є іншим гравцем (other_player).

Оцінка стану гри:

- Метод minimax перевіряє, чи вже є переможець чи гра завершилася в нічию. Якщо так, повертає відповідні очки для цього стану гри.

Рекурсивне визначення оптимального ходу:

- Залежно від поточного гравця (max_one або other_player), ініціалізуються початкові значення для оцінки (final).
- За допомогою вкладеного циклу перебираються всі можливі ходи.
- Кожен хід симулюється, і викликається рекурсивно метод minimax для оцінки результатів гри.

Оновлення оптимального ходу:

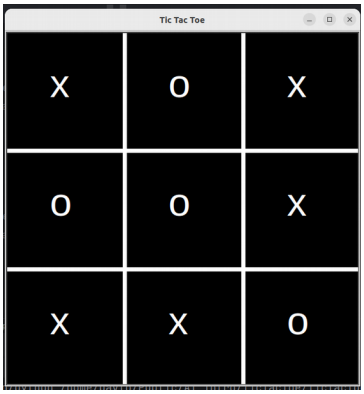
- Порівнюється отриманий результат (score) з поточним найкращим результатом (final), і якщо новий результат кращий, він оновлюється.

Повернення оптимального ходу:

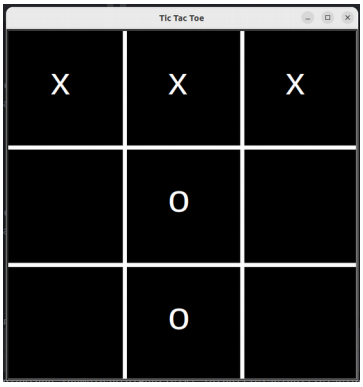
- Після перебору всіх можливих ходів метод повертає структуру з координатами (x, y) оптимального ходу та його оцінкою (score).
- У вас метод pass_move класу Computer викликає метод minimax для визначення оптимального ходу комп'ютера з урахуванням поточного стану гри. Цей алгоритм дозволяє комп'ютеру вибирати такі ходи, які максимізують його виграш або мінімізують виграш опонента в залежності від ситуації у грі.

Тестування

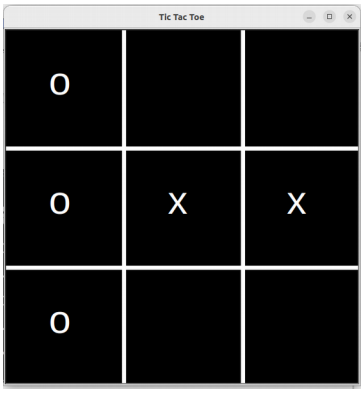
```
/home/pavlo/Public/AI_intro/TicTacToe/venv/bin/python /home/pavlo/Public/AI_intro/TicTacToe/TicTacToe.py
Choose the mode: 1 - Human-Computer, 2 - Computer-Human, 3 - Human-Human
Enter number 1-3: 1
```

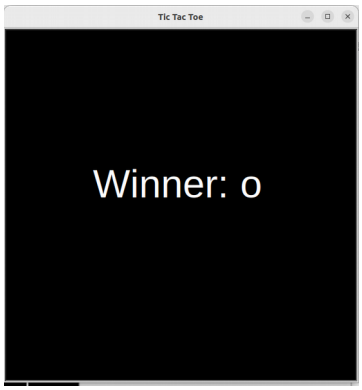



```
/home/pavlo/Public/AI_intro/TicTacToe/venv/bin/python /home/pavlo/Public/AI_intro/TicTacToe/TicTacToe.py
Choose the mode: 1 - Human-Computer, 2 - Computer-Human, 3 - Human-Human
Enter number 1-3: 2
```




```
/home/pavlo/Public/AI_intro/TicTacToe/venv/bin/python /home/pavlo/Public/AI_intro/TicTacToe/TicTacToe.py
Choose the mode: 1 - Human-Computer, 2 - Computer-Human, 3 - Human-Human
Enter number 1-3: 3
```





Висновки

Робота вдало вирішила завдання створення гри в хрестики-нулики з використанням алгоритму Мінімакс та графічного відображення засобами бібліотеки Turtle.

Проект став хорошою платформою для розуміння концепцій теорії ігор та реалізації алгоритмів штучного інтелекту в контексті ігор.

Є потенціал для розвитку проекту, додавання нових функцій та оптимізації алгоритму Мінімакс.

Рекомендується подальше розширення можливостей взаємодії гравців та можливість вибору складності гри.