

Бот

Більшість структури коду взята з бота-прикладу player1.py

- Функція main()

Отримує дані щодо номера гравця функцією.

parse_info_about_player(), запускає функцію play().

Якщо у програму не надходить даних, бот програв, виводить це на екран.

- Функція parse_info_about_player()

Отримує номер гравця.

- Функція play()

Цикл для виконання ходів:

З step() отримує координати, куди походити.

Виводить у віртуальну машину свій хід.

Кожний 8 хід(а також, якщо було важко знайти варіанти минулого разу) робить у зворотному напрямку, більше про це у функції step().

- Функція step()

Вирішує, куди походити

Для початку отримує дані щодо розмірів поля(parse_field_info), саме поле(parse_field), розмірів та значення фігури(parse_figure).

Потім проходиться по елементах(0 або X) гравця та для кожного дивиться, чи є можливість якось поставити туди фігуру. Тобто для кожного варіанту розміщення, щоб зірочка фігури була на цьому елементі перевіряє, чи при цьому фігура не перетинає інших елементів та не виходить за межі поля.

Таким чином якщо існує хоча б один хід, то функція його зробить. Водночас як такої тактики вибору немає. Єдине, що я змінюю параметром switch(передається з play(), робить зворотний порядок) є те, що спочатку функція буде дивитися варіанти розміщень знизу, а потім зверху чи навпаки.

Для того, щоб функція добре працювала і для першого, і для другого гравця, вона перевертає порядок перегляду рядків і стовпців для гравця 2.

У результаті для гравця 1 порядок перегляду буде з правого низу(тобто там частіше будуть знаходитися варіанти, малюнок буде рухатися до центру), а для гравця 2 з лівого верху(також буде йти до центру).

Через те, що кожен восьмий хід робиться у зворотному(вертикально) порядку, малюнок у результаті ніби “обходить” суперника і з іншого боку.

А через те, що коли довго було знаходити вільне місце минулого разу, також змінюється порядок, на малюнку бот буде намагатися уникати заходити в тупики.

- Функція `parse_field_info()`

Отримує дані щодо розмірів поля.

- Функція `parse_field()`

Отримує поле, без лишніх чисел та `ui`.

- Функція `parse_figure()`

Отримую розміри фігури та її саму.

Також я додав функцію `debug_info()`

Вона використовує модуль `logging`, аби логувати важливіші дані(зафарбовує їх у жовтий).