

Звіт до 3 міні-проєкту

Алгоритм роботи бота:

1. Початкове опрацювання поля. Із самого початку я отримую поле за допомогою інпутів перетворюю його у список списків. Пусті ячейки я заповнюю нижніми підкресленнями, а букви залишаю на своїх місцях. Опісля цього я роблю із полем операцію потрібну для оцінки ходів. Кожну пусту клітинку навколо противника я оцінюю одиницею. Опісля, кожну клітинку навколо одиницю я оцінюю двійками. І так далі. У результаті у мене залишається поле із оціненими клітинками, які будуть відігравати роль 'ціни' ходів для бота. Також потрібно зазначити, що можуть залишатися неоцінені клітинки, адже деякі клітинки бувають відрізнаними від ворога, тому ми їх залишаємо пустими.

	14	O	9	9	9	9	9	9	9	9	
	13	O	8	8	8	8	8	8	8	8	
	12	O	O	7	7	7	7	7	7	7	
	11	O	O	6	6	6	6	6	6	6	
	10	9	O	5	5	5	5	5	5	5	
	9	8	O	O	O	4	4	4	4	4	
	8	7	O	2	O	O	3	3	3	4	
	7	6	O	1	1	1	2	2	3	3	
	6	5	O	1	X	1	1	2	2	3	
	5	4	O	1	X	X	1	1	2	2	
	4	3	2	1	X	X	X	1	1	2	
	4	3	2	1	1	1	X	X	1	2	
	4	3	2	2	2	1	X	X	1	2	
	4	3	3	3	2	1	1	1	1	2	
	4	4	4	3	2	2	2	2	2	2	
	5	5	4	3	3	3	3	3	3	3	

Візуалізація оцінки поля

2. Тепер потрібно перебрати усі можливі кроки. Для цього у мене є функція checker, яка дивиться чи можливий хід і оцінює його 'ціну'. Вона просто добавляє цінник кожної оціненої клітинки до ціни ходу. При умові, що хід ставиться на пусту, неоцінену клітинку, то до 'ціни' добавляється 100, адже ми не хочемо, щоб бот заповнював вже відрізану площу, адже противник однаково не має доступу до цієї території. Потім я сортую усі

ходи за їхньою 'ціною' і вибираю той, у якому 'ціна' найдешевша, адже тоді бот буде завжди ставити найближчий можливий хід до противника.

3. Окрім всього нижче описаного, я додав можливість циклічно переходити по полю, тобто при умові, що моя фігура біля якоїсь стіни, то можна поставити фігуру так, щоб вона 'вилізла' з протилежної сторони. Це сильно збільшує можливість бота, і дає йому набагато більший потенціал.