

Короткий опис гри:

Сеттінг:

Створений вченими штучний інтелект вийшов з-під контролю вчених. Усі системи вийшли з ладу, він отримав доступ до державних секретних даних та паролів, воєнних ресурсів та системи авіації, світ у паніці. Завдання гравця знешкодити штучний інтелект.

Сюжет:

Гравцю потрібно пройти 4 рівні, аби знешкодити його. Також у гравця на всю гру буде 3 життя, які зникають, якщо гравець не пройшов рівень.

1 рівень:

Гравцю потрібно розв'язати певні завдання, які стосуються вибору парного або непарного числа зі списку. В кінці рівня користувач отримує географічні координати— місцезнаходження штучного інтелекту. Він вирушає на секретну базу, де знаходиться сам інтелект, щоб його знешкодити локально. Таким чином гравець переходить на другий рівень.

2 рівень:

Гравець знаходиться біля входних дверей секретної бази. На цей раз завдання пов'язане із щасливими числами— потрібно вибрати позицію щасливого числа у списку із двох чисел (користувач вводить 0, якщо це число стоїть перше у списку, 1 — якщо друге). Із одержаної послідовності 0 та 1 — двійкового коду — гравець отримує пароль до входних дверей. Довжина двійкового коду — 10 чисел, тобто 10 разів виводиться список із двох чисел, одне з яких — щасливе.

3 рівень:

Після проходження головних дверей, гравець повинен спуститися у цоколь, де знаходиться штучний інтелект. Тут його чекають ще одні зачинені двері, до яких потрібно підібрати пароль. Гравцю виводиться список з певної кількості чисел, із якого він повинен визначити, які числа є числами Улама. Гравець, відповідно до порядку їх розташування у списку, вводить в консоль, та програма перетворює їх на букви латинського алфавіту.

4 рівень:

Гравець потрапив у приміщення та знаходиться біля самого штучного інтелекту і він повинен локально відключити його. Гравець повинен буде використати засвоєнні на минулих рівнях знання — перед ним п'ять разів буде виводитись список із п'яти чисел, із якого йому потрібно буде ввести одне число, яке має комбіновані ознаки, тобто число, яке:

1. Не має жодної ознаки — не є ні парним, ні щасливим, ні Улама;
2. Є парним та є щасливим (проте не є Улама);
3. Є парним та є числом Улама (проте не є щасливим);
4. Є щасливим та є Улама (проте не є парним);
5. Поєднує усі три ознаки - є як парним, так і щасливим, та числом Улама.

Геймплей:

Гра відбувається під час взаємодії з командним рядком (під час гри на кожному етапі виводиться те, що потрібно зробити).

Особливості гри:

Під час гри виводяться коментарі штучного інтелекту

Процес роботи:

Комунікація: команда на платформі Microsoft Teams; змішана форма комунікації (офлайн та онлайн)

Розподіл завдань:

1. Спільне створення модулів numbers та main_game, блок-схеми, алгоритму, звіту, опису гри;
2. Перший рівень (модуль level1) - Ольга Люба;
3. Другий рівень (модуль level2) - Анастасія Ткачишин;
4. Третій рівень (модуль level3) - Ярослав Романус;
5. Четвертий рівень (модуль level4) - Анна-Аліна Бондарець і Олександра Стасюк;

Етапи розробки проекту:

1. Перша зустріч команди у Zoom, знайомство, обговорення планів;
2. Друга зустріч (офлайн + онлайн), створення концепту гри, створення блок-схеми, використання підходу top-down та розподіл завдань, встановлення дедлайнів для індивідуальних завдань;
3. Виконання індивідуальних завдань;
4. Третя зустріч (офлайн + онлайн), обговорення та перевірка результатів, тестування та виправлення модулів, початок розробки основного модуля;
5. Четверта зустріч (онлайн в Microsoft Teams), закінчення розробки гри, проведення тестування, виправлення всіх багів та формування фінального продукту.

Особливості технічної реалізації:

Ми використали підхід top-down для розробки нашої гри та розділили нашу гру на декілька модулів та функцій у кожному модулі. Для побудови алгоритму у вигляді блок-схеми ми скористалися сервісом draw.io. У процесі розробки наших модулів ми використовували модуль random.

Структура нашого проекту:

Модулі:

1. numbers.py - забезпечує всі функції для роботи з типами чисел(парні, щасливі, Улама), які використовуються у рівнях.

Функції:

- ulam_sequence(num: int) -> list - повертає список всіх чисел Улама, які \leq за num
- is_ulam(num: int) -> bool - повертає True якщо num число Улама, інакше False
- happy_number(num: int) -> int - Повертає рандомне щасливе число, яке \leq за задане
- is_happy(num: int) -> bool - Перевіряє чи щасливе задане число
- is_even(num: int) -> bool - Перевіряє чи парне задане число

2. level1.py - відповідає за запуск та роботу 1-ого рівня гри.

Функції:

- `first_level()` -> bool - запускає перший рівень, повертає значення True, якщо користувач правильно його пройшов і розв'язав усі завдання та False, якщо ні;
- `even_and_odd(max_num: int, positive: bool)` -> tuple - повертає кортеж з двох елементів: списку з 5 рандомних чисел та типу числа, яке користувачу потрібно вибрати (парне чи непарне). У списку лише одне число відповідає необхідній парності, а інші 4 - ні. Функція приймає 2 аргументи: максимальне число, яке може бути згенероване у списку, і булеве значення, яке відповідає за можливість знаходження від'ємних чисел у списку (якщо True - лише невід'ємні, якщо False - від'ємні також).
- `right_answer(task: tuple)` -> int - приймає кортеж з типом числа та списком з 5 чисел. Повертає число з цього списку, яке відповідає вказаному типу (парне чи непарне).
- `check_user(task: tuple)` -> bool - приймає кортеж з типом числа та списком з 5 чисел. Повертає True, якщо відповідь користувача співпадає з необхідною відповіддю, і False в іншому випадку.

3. `level2.py` - відповідає за запуск та роботу другого рівня гри.

Функції:

- `second_level()` -> bool - власне запускає другий рівень, повертає значення True, якщо рівень пройдено і False у іншому випадку

4. `level3.py` - відповідає за запуск та роботу третього рівня гри.

Функції:

- `generate_grid()` -> list - Повертає список з 20 чисел, що ≤ 26 (бо ми маємо перетворювати ці числа в літери алфавіту)
- `num_into_letter(num: int)` -> str - Повертає відповідну для заданого числа букву з алфавіту
- `third_level()` -> bool - Власне запускає третій рівень, повертає значення True, якщо рівень пройдено і False у іншому випадку

5. `level4.py` - відповідає за запуск та роботу четвертого (фінального) рівня гри.

Функції:

- `none_prop()` -> list - генерує список з усіма числами, що задовольняють умову: усі числа не є ні парними, ні щасливими, ні числами Улама;
 - `even_and_happy()` -> list - генерує список з усіма числами, що задовольняють умову: всі числа є парними та щасливими;
 - `even_and_ulam()` -> list - генерує список з усіма числами, що задовольняють умову: всі числа є парними та числами Улама;
 - `happy_and_ulam()` -> list - генерує список з усіма числами, що задовольняють умову: всі числа є щасливими та числами Улама
 - `all_prop()` -> list - генерує список з усіма числами, що задовольняють умову: всі числа є і парними, і щасливими, і числами Улама
 - `create_output_list(lst_right_numbers: list)` -> list - створює список із 4-ма елементами, які є неправильними відповідями для заданого завдання (ці елементи вибираються, як числа, що не входять у відповідний створений вище список із правильних відповідей)
 - `right_answer(lst_right_numbers: list)` -> int - вибирає рандомно число із списку із правильними відповідями. Це число і буде єдиною правильною відповіддю у кінцевому результаті
 - `fourth_level()` -> bool - власне запускає четвертий рівень, повертає значення True, якщо усі відповіді гравця були введені правильно та False, якщо хоча б одна відповідь неправильна.
6. `main_game.py` - основна частина гри, запускає у собі всі рівні та підсумовує результати

Алгоритм:

