“Gypsy” – гра у загадки з елементами RPG.

Сеттінг – дім головного героя-гравця, який знаходить в Інтернеті номер циганки, що обіцяє недорого передбачити його\її майбутнє.

За сюжетом, у головного героя недостатньо коштів, щоб дозволити собі заплатити циганці за її передбачення, але він\вона все рівно почує своє майбутнє, якщо зможе відгадати 10 загадок. Якщо гравцю вдається відповісти на всі загадки безпомилково, то він\вона почує своє майбутнє зразу. Якщо під час розгадування він\вона допустить хоч одну помилку, то буде змушений підібрати код до скриньки, яка чарівним чином з’являється у його кімнаті, шляхом розв’язання трьох прикладів на додавання бінарних чисел, і тільки тоді почує своє майбутнє. Якщо ж гравець не зможе назбирати 10 правильних відповідей до того, як усі загадки будуть використаними, він\вона буде змушеним пройти гру спочатку (вона автоматично перезапуститься).

Для того, щоб запустити гру, потрібно скористатись будь-якою програмою, що може читати код на мові програмування Python (до прикладу, Visual Studio Code, PyCharm)

Скріншоти з гри:

1. Початок гри, знайомство з гравцем і ознайомлення гравця із концепцією гри.

2. Вибір однієї з двох дій, ознака RPG

3. Приклад завдань із загадками

4. Приклад завдань на додавання бінарних чисел.

5. Приклад передбачення.

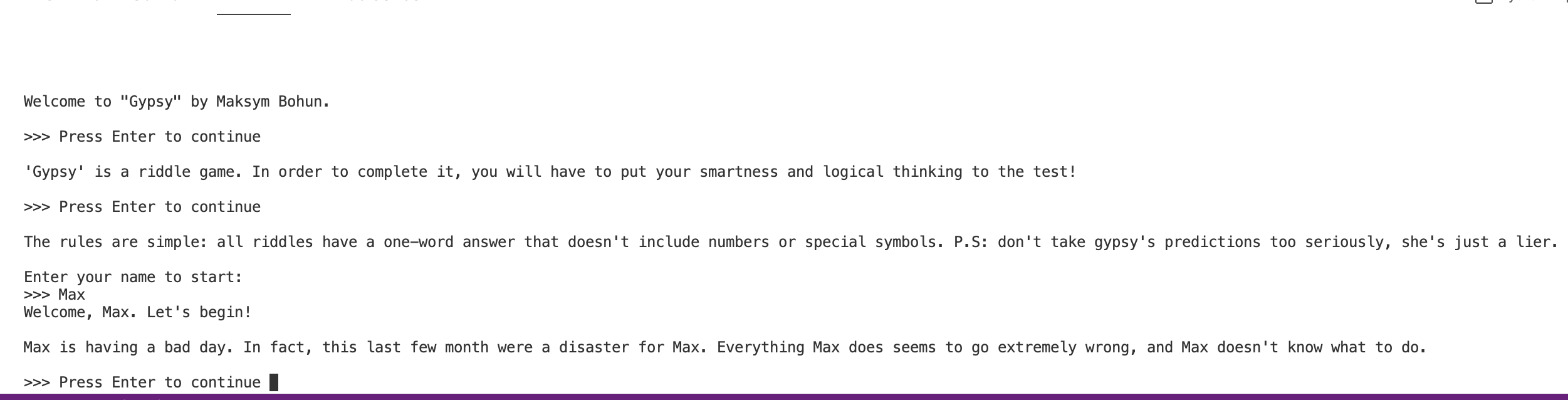
Розробка гри складалась із трьох етапів:

1. Розробка концепції гри. На цьому етапі я продумував різні варіанти ігор, які мені дозволяє створити моє знання програмування (розглядав квіз із відгадуванням столиць, РПГ з розгалуженнями сюжету та виселицю, але зупинився на грі у загадки з елементами РПГ. Тоді ж я продумав і сеттінг гри. Етап зайняв не більше години часу.

2. Продумування сюжету та функцій гри. Спочатку думав після загадок просто давати передбачення, але вирішив, що гра буде досить сухою, і вирішив додати вступ, що знайомив би гравця із грою та міні-гру на підбирання коду (подібну у своїй грі реалізував мій сусід по кімнати у Колегіумі Максим Сухарський, але в нього потрібно було перетворити бінарне число у десяткове, а я вирішив заставити гравця рахувати суму бінарних чисел). Етап зайняв не більше години часу.

3. Технічний етап. На цьому етапі потрібно вже було почати реалізовувати задумки, але код я писав поступово: спочатку, оскільки на той момент ми ще не вивчили функції, я написав програму, що випадковим чином виводила користувачеві загадки і перевіряла, чи відповідь на них правильна, а також рахувала помилки і кількість правильних відповідей. Згодом, коли ми вивчили функції, я додав першу частину коду у функцію main, а потім написав функції opening, minigame та randomprediction. У самому кінці я почав шукати більше загадок і додав нові передбачення, а потім описав роботу кожної функції у коді. Етап зайняв більше 10 годин.

Гра складається і чотирьох функцій та списку загадок. Першою завжди викликається функція opening, що знайомить гравця із грою і розказує передісторію, дає мотивацію відгадувати загадки. Другою завжди запускається функція main, у якій гравець повинен розгадувати загадки. Якщо кількість правильних відповідей досягає 10, і гравець не зробив жодної помилки, то запускається функція pandomprediction, що генерує передбачення та завершує гру. Якщо ж під час розгадування гравець допускає хоч одну помилку, то після main запуститься функція minigame, у якій гравець повинен буде розв’язати три випадково згенерованих приклади на додавання бінарних чисел. Програти на цьому етапі неможливо, отже четвертою точно запуститься функція randomprediction. Якщо ж загадок більше не залишиться, а кількість правильних відповідей буде меншою 10, то гравець програє і функція main знову викличе функцію opening.



2. Вибір однієї з двох дій, ознака RPG

3. Приклад завдань із загадками

4. Приклад завдань на додавання бінарних чисел.

5. Приклад передбачення.