

Оглавление

1	Запуск игры	2
2	Создание игры	3
3	Игра	4
4	Сохранение и загрузка игры	6
5	AI	7
6	Структура классов	9
7	Что стоило бы добавить или исправить	11

Глава 1

Запуск игры

Чтобы запустить игру, нужно запустить `main.py`. Из сторонних библиотек потребуется только `numpy`.

Глава 2

Создание игры

В меню игрок может выбрать один из пунктов меню:

1. 1 - Простой режим.
2. 2 - Пользовательский режим.
3. 3 - Загрузить игру
4. 0 - Выйти

В простом режиме автоматически генерируется поле размером 5x5. Количество бомб генерируется случайным образом от 2 до 5. Бомбы расставляются ещё до первого хода, поэтому игрок может проиграть первым же ходом.

В пользовательском режиме пользователь дополнительно вводит размер поля (не меньше 5 по каждому из измерений), а также количество бомб (не менее 2). Проверка ввода присутствует, в случае чего программа будет запрашивать данные до тех пор, пока пользователь не введёт корректные значения.

Глава 3

Игра

Во время игры игрок после каждого хода видит карту, где клетка принимает одно из значений

- Цифра от 0 до 9 - количество бомб в соседних клетках, сама клетка уже открыта
- 'F' - в клетке поставлен флаг
- 'X' - клетка закрыта от пользователя.

Нумерация для пользователя начинается 1, и ввод клетки также происходит с 1

Игрок может сделать только одно действие из перечня:

1. Open - открыть клетку.
2. Flag - поставить флаг.
3. AI - запросить подсказку от АИ.
4. Save - сохранить текущее положение игры.
5. Exit - выйти из игры.

В случае выбора Open или Flag игрок должен указать координаты клетки, с которой нужно произвести действия. После каждого

хода происходит проверка, выиграл ли игрок. Если на карте открыты все ячейки, кроме бомб - то он выиграл, иначе игра продолжается.

По команде AI написанный мной алгоритм решения задачи подсказывает подходящее действие для текущего хода.

Глава 4

Сохранение и загрузка игры

Для работы с сохранениями используется библиотека `pickle`. Она выбрана потому, что:

- Имеется опыт работы с этой библиотекой.
- Пользователь не может узнать координаты, просто открыв файл.

Загрузку игры можно произвести только с главного меню, а сохранить игру - только во время игры. В файл `.pickle` попадает информация о карте, её размеры, а также текущий счёт пользователя. Пользователь при выполнении действий должен указать имя файла без расширения. Все игры автоматически сохраняются в папку `savings`.

Глава 5

AI

Алгоритм для выбора состоит из двух этапов:

1. Поиск точного решения задачи.
2. Если точного решения нет - то найти клетку с минимальной вероятностью нахождения бомбы в клетке.

Для первого пункта каждой открытой ячейке в соответствие ставится группа (Group): list из номеров ячеек, которые не открыты и находятся рядом. Создаётся list из всех групп, который хранится у AI_Manager. После этого группы преобразовываются: каждая сравнивается с каждой. Все повторяющиеся группы удаляются, из множеств вычитаются подмножества, а пересекающиеся множества дробятся на три части. Преобразования продолжаются до тех пор, пока изменять будет нечего. После этого остаются три группы:

- Группы, в которых количество мин == 0
- Группы, в которых количество мин == количеству бомб
- Остальные

Если есть хоть одна группа типа 1 или типа 2, то AI сразу выдаёт решение.

Если же таких групп нет, то для каждой ячейки подсчитывается вероятность нахождения бомбы. По формуле 5 можно посчитать вероятность нахождения бомбы, зная вероятность нахождения бомбы из каждой группы, в которую входит ячейка. $P = 1 - (1 - A_1) * (1 - A_2)$..

Затем значения в каждой группе корректируются, учитывая, что сумма вероятностей по группе должна быть $\equiv 1$. В качестве ответа выбирается ячейка с минимальным значением вероятности.

Следует отметить, что АИ выиграла несколько простых игр, когда для тестирования я каждый раз выполнял команду, которую посоветует АИ.

Подробнее алгоритм разобран на Хабре: <https://habr.com/ru/post/211188/>

Алгоритм отличается от исходного тем, что последние ходы не рассматриваются отдельно, а реализация сильно отличается от предложенного там варианта.

Глава 6

Структура классов

GameTable - карта. Не принимает никаких решений, хранит матрицу значений для каждой из ячеек.

GameKeeper - сущность, которая хранит карту. Может расставлять объекты на карту, хранит все основные показатели игры (количество бомб, текущее число очков). При этом решений практически не принимает, кроме случайного распределения бомб. Может просить Shower вывести матрицу.

GameManager - сущность, которая передаёт инструкции по поводу игры (открыть ячейку, поставить флаг, показать счёт) от админа. Связывает между собой админа и хранителя игры, который хранится как атрибут класса. Соответственно, на каждую игру создаётся свой менеджер

GameHelper - сущность, которая взаимодействует с игроком. Единственная сущность, которая что-то выводит на экран, а также принимает от пользователя параметры.

GameAdmin - сущность, которая принимает ключевые решения: какую создать игру, что вообще сделать. Создаётся при запуске программы, по сути - вершина всей структуры. Владеет хелпером, и менеджером, которого каждый раз пересоздаёт при создании новой игры

Solver - сущность, которая просто возвращает True/False, является ли клетка бомбой. Практически не используется, роль минимальна

Saver - сущность, которая сохраняет игру.

Loader - сущность, которая загружает игру

AI - фасад для взаимодействия с АИ. Имеет только один метод - `predict()`. Говорит менеджеру, что нужно сделать

AI_Manager - сущность, которая выполняет всю логику выбора правильного хода. Считает вероятности, выполняет действия с группами.

Cell - сущность ячейки. Практически не используется, как ни странно

Group - сущность группы, хранит все возможные операции, возможные с группой.

Глава 7

Что стоило бы добавить или исправить

Здесь перечислены вещи, которые я не успел сделать к дедлайну, но хотел:

- Сделать генерацию карты после первого хода.
- Если вокруг ячейки нет бомб, то открыть все соседние (но тогда игра 5x5 может значительно упроститься).
- Добавить комментарии в код
- Изменить реализацию Cell, потому что там хранится атрибут вероятности, но фактически не используется
- Декомпонировать менеджера AI.
- На последнем ходу AI иногда советует не открыть, а поставить флаг.
- Сделать нормальную архитектуру/написать нормальный код.
- Сделать нормальную документацию (даже это я писал на скорую руку)