

# Sprawozdanie z projektu „gra saper”

Krystian Kamiński

## 1. Wymagania

- Możliwość ustawienia poziomu trudności, na który składa się wymiar planszy, oraz liczba min.
- Możliwość zaznaczania flagami potencjalnych min
- Pokazywanie informacji o liczbie chorągiewek
- Pokazywanie obliczonego współczynnika 3BV.
- Plansza na której toczy się rozgrywka może być:
  - Wczytywana z pliku (tj. informacja o pozycji bomb na planszy). Plik nie zawiera informacji o liczbie min, z którą styka się puste pole.
  - Wygenerowana 'w locie' - przy nowej grze.
  - Zapisana po uprzednim wygenerowaniu przez program.
- Pokazanie podsumowania gry
  - Czas rozgrywki (całkowity, nie trzeba na bieżąco pokazywać czasu rozgrywki podczas gry)
  - Wynik rozgrywki

## 2. Wykorzystane biblioteki:

- random
- time

oraz do testów jednostkowych:

- Io
- pytest
- \_pytest.monkeypatch

## 3. Działanie programu:

Aby użytkownik mógł zagrać w grę, musi otworzyć w terminalu plik saper\_program.py wtedy będzie miał do wyboru kilka możliwości, Spośród tych poniżej przedstawionych. Aby wybrać którąś opcję należy Wpisać odpowiednią liczbę i ją zatwierdzić.

```

Select number
1. Beginner
2. Advanced
3. Expert
4. Set own board, random mines
5. Set board from file
6. Exit the game
Select difficult level: █

```

W przypadku wyboru opcji od 1 do 3 wyświetlona zostanie tablica do grania, należy wybrać odpowiednie pole i opcję na jak je wykorzystać

```

There are 10 flags to use
      1  2  3  4  5  6  7  8

1   .  .  .  .  .  .  .  .
2   .  .  .  .  .  .  .  .
3   .  .  .  .  .  .  .  .
4   .  .  .  .  .  .  .  .
5   .  .  .  .  .  .  .  .
6   .  .  .  .  .  .  .  .
7   .  .  .  .  .  .  .  .
8   .  .  .  .  .  .  .  .

Select x value in range(1, 8): 2
Select y value in range(1, 8): 2
Select 'f' to flag option or 'd' to discover field: d

```

Gra trwa do momentu, aż gracz wybierze minę lub odkryje wszystkie puste pola, a następnie pojawi się możliwość zapisu planszy do pliku.

Gra trwa do momentu wybrania w menu opcji nr 6 czyli, końca gry.

Analogicznie jak dla opcji 1-3 wygląda sytuacja przy wyborze 4-5, z tą różnicą, że w przypadku wyboru 4 mamy możliwość ustawienia ilości min, i rozmiaru planszy z podanego zakresu, a w przypadku opcji 5 możemy wczytać planszę z pliku.

Poniżej przedstawię przykładową rozgrywkę:

```

Select number
1. Beginner
2. Advanced
3. Expert
4. Set own board, random mines
5. Set board from file
6. Exit the game
Select difficult level: 4
You choosed 4 option

Set width of board from 8 to 30: 30
Set high of board from 8 to 24: 8
Set amount mines from 10 to 203: 50
There are 50 flags to use
  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
2  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
3  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
4  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
5  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
6  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
7  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
8  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

Select x value in range(1, 30): 1
Select y value in range(1, 8): 1
Select 'f' to flag option or 'd' to discover field: d
Game is over, you lost

The game lasted 0 seconds.

  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

1  *  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  *  *  .  .  .  .  .  .  .  .  .  *  *
2  .  .  .  .  *  .  .  .  .  .  .  .  *  .  .  .  *  *  *  *  .  .  .  .  .  .  .  .  .
3  .  *  *  .  *  .  .  .  .  .  .  .  *  .  .  .  .  .  *  .  .  .  .  .  .  .  .  .  *
4  *  .  .  .  .  .  .  .  *  .  .  .  .  .  .  .  .  .  *  *  *  *  .  .  .  *  *  .  .
5  .  .  .  .  .  .  .  .  .  .  *  .  *  .  .  *  .  .  .  .  .  .  .  .  .  *  .  *
6  .  .  .  .  *  .  .  .  *  .  *  .  .  .  .  .  .  .  .  *  *  *  *  .  *  *  .  .
7  .  *  .  *  .  .  .  .  .  .  .  .  .  .  *  .  .  .  *  .  .  .  .  .  .  .  .  .  .
8  *  .  .  .  .  *  .  .  .  .  .  .  .  .  .  .  .  *  .  .  *  .  *  .  .  *  .  .  *  *

Type 1 if you want to save it to file:
Select number
1. Beginner
2. Advanced
3. Expert
4. Set own board, random mines
5. Set board from file
6. Exit the game
Select difficult level: █

```

Aby odczyt z pliku był poprawny musi on być w formacie .txt i zawierać wyłącznie znaki „.” oraz „\*”. Ilość znaków w rzędzie musi być z zakresu od 8 do 32, ilość linii od 8 do 24, a liczba min „\*” musi być z zakresu od 10 do  $(x-1)*(y-1)$ , gdzie x, y są wymiarami planszy

Przykładowa plansza poniżej:

```

. . . * . . .
. . . . . .
. . * . . . .
. . * . * . .
. . . . . **
* . . . . .
. * . * . . .
. . . . * . .

```

#### 4. Klasy, organizacja kodu:

Cały program składa się z 3 plików z zaimplementowanymi klasami, funkcjami na które składają się: `model.py`, `saper_program.py`, `model_io.py`

Plik `saper_program.py` zawiera funkcję `main`, korzysta z klas, metod, funkcji z pozostałych dwóch plików, stanowi element interakcji użytkownikiem.

`model.py` :

W tym pliku zaimplementowana jest klasa `Board`, której obiekty reprezentują tablice w grze `saper`. Parametry potrzebne do utworzenia obiektu tej klasy to: wartość współrzędnej `x`, `y`, oraz ilość min. Klasa zawiera metody niezbędne do ustawiania wartości pól, generowania losowych pozycji min itd.

Opis wszystkich funkcji/metod znajduje się w kodzie źródłowym.

Wszelkie błędne argumenty zostały obsłużone wyjątkami.

`model_io.py` :

Zaimplementowana została struktura potrzebna do wczytywania zapisywania losowo wygenerowanej tablicy do pliku tekstowego. funkcja `write_to_file`, zajmuje się zapisem danych do pliku, oraz sprawdzeniem czy dane zapisywane są poprawne, w przeciwieństwie do wczytywania z pliku, które zostało podzielone na dwie osobne funkcje `open_file` obsługującą plik, oraz `filehandle_board` która otrzymuje już dane pobrane z pliku.

Taka konfiguracja pozwoliła mi nieco uprościć testowanie.

Sytuacje wyjątkowe zostały obsłużone.

#### 5. Testy jednostkowe:

Testy zostały umieszczone w dwóch plikach `test_model.py`,  
`test_model_io.py`

`test_model.py` : Testowanie poprawności tworzenia obiektu klasy `Board`, sprawdzenie poszczególnych wartości atrybutów, przypadków granicznych,

poprawności obsługi wyjątków, dla niepoprawnych danych wejściowych.

test\_model\_io.py : sprawdzenie poprawności wczytywania/zapisywania danych do pliku, sytuacji granicznych, sprawdzenie rzucania wyjątków dla niepoprawnych danych w plikach.

## 6. Podsumowanie

Wszystkie testy jednostkowe ostatecznie przebiegają pomyślnie, nie są wykryte, żadne błędy spowodowane błędnymi danymi podanymi, przez użytkownika. Gra przebiega płynnie, jest stabilna, odporna na błędy ze strony użytkownika.