

Saper (Minesweeper) - Projekt w C

Opis projektu

Saper to klasyczna gra, której celem jest odkrywanie pól na planszy bez natrafiania na miny. Każde odkryte pole wyświetla liczbę min sąsiadujących z tym polem. Gra umożliwia ustawianie flag na polach które podejrzewamy o zawieranie min, co zapobiega ich przypadkowemu odkryciu.

Projekt został zaimplementowany w języku C na potrzebę kursu Informatyka Stosowana realizowanego przeze mnie na Politechnice Warszawskiej.

Funkcjonalności

1. Trzy poziomy trudności oraz tryb niestandardowy:

- **Łatwy:** 9x9 pól, 10 min.
- **Średni:** 16x16 pól, 40 min.
- **Trudny:** 16x30 pól, 99 min.
- **Niestandardowy:** możliwość definiowania własnych wymiarów planszy i liczby min.

2. Polecenia gracza:

- **f x y** – ustawienie lub usunięcie flagi na polu (x, y).
- **r x y** – odkrycie pola (x, y).

3. System punktacji:

- Wynik gracza obliczany jako iloczyn liczby odkrytych pól i mnożnika zależnego od poziomu trudności:
 - **Łatwy:** mnożnik 1.
 - **Średni:** mnożnik 2.
 - **Trudny:** mnożnik 3.
 - **Niestandardowy:** mnożnik 1.

4. Tablica wyników:

- Przechowuje maksymalnie 100 wyników graczy.
- Wyniki są sortowane według liczby punktów i zapisywane w pliku.
- Program wyświetla top 5 najlepszych rezultatów.

5. Obsługa plików:

- Możliwość wczytywania planszy z gotowego pliku za pomocą flagi **-f**.
- W pliku można również określić sekwencję ruchów, które zostaną automatycznie wykonane.

6. Obsługa flag:

- **-f <plik>** – uruchomienie gry na podstawie planszy zapisanej w pliku.
- **-s** – wyświetlenie tablicy wyników.

Struktura projektu

```
├── include/
│   ├── board.h      # Deklaracje funkcji i struktur związanych z planszą
│   ├── game.h       # Deklaracje funkcji związanych z logiką gry
│   ├── score.h      # Deklaracje funkcji związanych z wynikami
│   └── file.h        # Deklaracje funkcji do obsługi plików
├── src/
│   ├── board.c       # Implementacja funkcji związanych z planszą
│   ├── game.c        # Implementacja logiki gry
│   ├── score.c       # Implementacja obsługi wyników
│   └── file.c        # Implementacja obsługi plików
├── main.c            # Główny plik programu
├── Makefile          # Plik do kompilacji programu
├── scoreboard.txt    # Plik przechowujący tablicę wyników
├── input.txt         # Przykładowy plik wejściowy
└── README.md        # Dokumentacja projektu
```

Sposób uruchomienia

1. Kompilacja programu

Uruchom polecenie:

```
make
```

2. Uruchomienie gry

- **Standardowy tryb gry:**

```
make run
```

lub

```
./minesweeper
```

Następnie program poprosi o wybór poziomu trudności.

- **Uruchomienie z plikiem (w przypadku użycia `make` program domyślnie otworzy plik `input.txt`:**

```
make run_file
```

lub

```
./minesweeper -f input.txt
```

Następnie program uruchomi grę na podstawie planszy zawartej w pliku input.txt i wykona ruchu zadeklarowane pod planszą

- **Wyświetlenie tablicy wyników:**

```
make run_scoreboard
```

lub

```
./minesweeper -s
```

Format plików

Plik wejściowy planszy (input.txt):

```
5 5 [ Wymiary planszy x y ]
. . * . . [ . - puste pole, * - mina ]
. . . . .
. . . * .
. . . . .
* . . . .
1 1 [ Polecenie: r 1 1 ]
5 5
1 5
3 5
```

Plik wyników (scoreboard.txt):

```
Jan 1000
Maja 451
Kacper 133
```

Przykład działania

Uruchomienie gry:

```
make run
```

Wyjście:

```
Welcome to Minesweeper game!
Select game difficulty (1 - easy, 2 - medium, 3 - hard, 4 - custom): 1
Level: 1, Rows: 9, Cols: 9, Mines: 10
  1  2  3  4  5  6  7  8  9
1 ■ ■ ■ ■ ■ ■ ■ ■ ■
2 ■ ■ ■ ■ ■ ■ ■ ■ ■
3 ■ ■ ■ ■ ■ ■ ■ ■ ■
4 ■ ■ ■ ■ ■ ■ ■ ■ ■
5 ■ ■ ■ ■ ■ ■ ■ ■ ■
6 ■ ■ ■ ■ ■ ■ ■ ■ ■
7 ■ ■ ■ ■ ■ ■ ■ ■ ■
8 ■ ■ ■ ■ ■ ■ ■ ■ ■
9 ■ ■ ■ ■ ■ ■ ■ ■ ■

SCORE: 0
Enter your command (f x y to flag Cell, r x y to reveal Cell): r 1 5
  1  2  3  4  5  6  7  8  9
1      1  1  1      2 ■
2      1  1  1      2 ■
3      1 ■ 2  1  1  1 ■
4      1 ■ ■ ■ ■ ■ ■
5      1  1 ■ ■ ■ ■ ■
6      1 ■ ■ ■ ■ ■ ■
7      1 ■ ■ ■ ■ ■ ■
8  1  1 ■ ■ ■ ■ ■ ■
9 ■ ■ ■ ■ ■ ■ ■ ■ ■

SCORE: 35
Enter your command (f x y to flag Cell, r x y to reveal Cell):
```

Autor

- Olaf Naruszko