

Politechnika Śląska
Wydział Matematyk Stosowanej
Kierunek Informatyka

Gliwice, 25.01.2022

Programowanie I
projekt zaliczeniowy

"Saper"

Wojciech Olech gr.2 lab.4

1. Opis projektu.

Saper – gra logiczna polegająca na odkrywaniu pól. Liczby oznaczają, ile min znajduje się na sąsiednich polach. Jeżeli użytkownik odkryje pole z miną, przegrywa. Celem gry jest odkrycie wszystkich pól niezawierających min.

2. Wymagania

- Interaktywne menu gry
- Możliwość wyboru ustanowionego poziomu trudności (Łatwy, Średni, Trudny)
- Możliwość stworzenia własnego poziomu trudności (Niestandardowy)
- Możliwość powtórzenia gry na tym samym poziomie trudności bez konieczności powrotu do menu wyboru poziomu
- Losowe rozmieszczanie min
- Ruch po planszy przy użyciu wcześniej ustalonych klawiszy
- Możliwość oznaczania pól z minami (oflagowywanie min)
- Możliwość odkrywania pola nieoflagowanego
- Odkrywanie sąsiednich pól nieoflagowanych, jeżeli odkryte pole posiada zero sąsiadujących z nimi minami
- Kończenie rozgrywki w momencie trafienia na minę
- Kończenie rozgrywki w momencie odkrycia wszystkich pól niezawierających min
- Możliwość zmiany ustawień sterowania
- Wyjście z programu z zabezpieczeniem przed przypadkowym kliknięciem w przycisk (prośba o potwierdzenie zakończenia programu)

3. Przebieg realizacji

Projekt składa się jedynie z pliku main.cpp, w którym znajduje się cały kod programu. Przy włączaniu go po raz pierwszy, tworzy nowy plik, o nazwie „opcje.txt”, w którym są przechowywane klawisze sterowania programu.

Każde pole w tym programie jest opisane strukturą, w której mamy zapisane, co zawiera dane pole: gracza, minę, flagę, czy też liczbę, z iloma minami sąsiaduje pole.

Wykorzystałem dodatkowo do tego programu bibliotekę „windows.h”, którą użyłem w celu zabarwienia planszy kolorami (każda cyfra, która oznacza, ile min jest wokół danego pola, posiada swój kolor plus gracz jest oznaczony kolorem czerwonym)

Gdy zostanie odkryte pole, które nie sąsiaduje z innym polem, zostaje wywołana funkcja rekurencyjna, która odkrywa pola wokół, do momentu, kiedy wszystkie pola nie zostaną odkryte.

4. Instrukcja użytkownika

Instrukcja znajduje się wewnątrz programu.

Ruch domyślny: góra - w; dół - s; lewo - a; prawo - d; oflagowanie pola - f; odkrycie pola - p. Ruch można zmienić w opcjach.

Gra polega na odkrywaniu pól (X) niebędących minami. Czerwony punkt na planszy wskazuje obecną pozycję gracza.

```
Pozostało flag: 10
X  X  X  X  X  X  X  X  X
X  X  X  X  X  X  X  X  X
X  X  X  X  X  X  X  X  X
X  X  X  X  X  X  X  X  X
X  X  X  X  X  X  X  X  X
X  X  X  X  X  X  X  X  X
X  X  X  X  X  X  X  X  X
X  X  X  X  X  X  X  X  X
X  X  X  X  X  X  X  X  X
```

Jeżeli gracz wpadnie na minę, zostaje o tym poinformowany.

```
Pozostało flag: 10
0  2  *  2  0  0  0  0  0
0  2  *  2  1  1  1  1  1
0  2  2  3  2  *  1  1  *
0  1  *  2  *  3  2  1  1
0  1  1  2  2  *  1  0  0
0  0  0  0  1  1  2  1  1
0  0  0  0  0  0  1  *  1
0  0  0  0  0  0  2  3  3
0  0  0  0  0  0  1  *  *

Wpadłeś w minę :(
Legenda oznaczeń:
F - flaga postawiona na minie
x - flaga niepostawiona na minie
* - mina

Wciśnij dowolny przycisk, aby kontynuować
```

To samo tyczy się, gdy gracz wygra, czyli odkryje wszystkie pola niebędące minami.

```
Pozostało flag: 1
0  0  0  0  0
0  0  0  0  0
0  0  0  0  0
1  1  1  0  0
1  X  1  0  0
Wygrałeś !!!:)
Wciśnij dowolny przycisk, aby kontynuować
```

Można z gry wyjść wcześniej naciskając klawisz ESCAPE. W takim wypadku, gra się kończy automatycznie. Oczywiście, po każdym wyjściu, jest zapytanie, czy użytkownik chce zagrać jeszcze raz. W takim wypadku, znowu zostaje wygenerowana plansza o takich samych rozmiarach. Chyba, że użytkownik wybrał niestandardowy rozmiar, to wtedy program prosi go o ponowne wprowadzenie danych.

Gdy wybierzemy planszę niestandardową i wpisujemy jakąś liczbę po przecinku, program przyjmie jej cyfry, aż do przecinka.

5. Podsumowanie i wnioski.

Udało mi się zrealizować wszystko, co chciałem, by było w programie. Nie miałem z niczym problemów.