

SPRAWOZDANIE – SAPER

Wykonała Edyta Kozłowska

Spis treści

Podręcznik używania programu

Zasady gry.....	1
Sposób uruchomienia programu.....	1
Opis poszczególnych opcji.....	2

Szczegóły implementacji programu

Podział programu na moduły.....	2
Kluczowe funkcje.....	2
Testy.....	4

Podsumowanie.....	4
-------------------	---

Podręcznik używania programu

Zasady gry

Grę zaczyna się z zakrytą planszą. Jej celem jest, aby pola pod którymi kryją się bomby oznaczyć flagą, a wszystkie inne odkryć. Przy odkrywaniu można trafić na pole „[*]” – oznacza ono bombę i kończy grę. Pierwszy ruch ma zapewnienie, że nie trafi się na bombę, a więc można trafić na pole z cyfrą, np. „[1]” lub puste pole „[]”. Jeśli jest to pole z cyfrą, to oznacza, że dokładnie tyle bomb jest dookoła tego pola, a więc maksymalnie może być 8. Jeśli trafimy na puste pole, to odkrywają się pola dookoła niego. Jeśli wydają nam się, że dane pole to bomba to należy je oflagować „[f]”, aczkolwiek można to cofnąć, jeśli uznamy inaczej.

Sposób uruchomienia programu

Mając pobrane pliki, należy w terminalu przejść do ścieżki plików (np. `cd C:\Users\user\folder`). Teraz należy skompilować program za pomocą komendy `make`, a

następnie trzeba go wywołać wpisując `./saper`. Wywołując program można dopisać `-f <ścieżka do pliku>` aby uruchomić program w trybie wczytywania planszy i ruchów z pliku.

Opis poszczególnych opcji

Na początku po wywołaniu programu należy wybrać poziom, a jeśli zostanie wybrany poziom 4, należy następnie podać długość i szerokość planszy oraz ilość min z podanego zakresu. Następnie należy wpisać swój ruch. Wpisujemy go w formacie „opcja x y”, gdzie opcja to r – jeśli chcemy odkryć pole, lub f – jeśli chcemy oflagować pole, a x i y to współrzędne punktu, z którym chcemy wykonać akcję (x to długość, y to szerokość). Po zakończeniu gry należy wpisać swój nick, a program zapisze go wraz z wynikiem do pliku. Można również przy wywoływaniu programu wpisać `-f <ścieżka do pliku>` aby program wczytywał plansze i ruchy z pliku, do którego podano ścieżkę.

Szczegóły implementacji programu

Podział programu na moduły

Program został podzielony na 4 moduły: plik główny – *main.c*, oraz 3 pliki, w których znajdują się funkcje programu. W pliku *plansza1.c* są funkcje dotyczące generowania planszy, odkrywania jej i wypisywania. W *plansza2.c* są dwie funkcje, jedna z nich wylicza ile pozostało bomb do oflagowania i wypisuje tę ilość, a druga liczy punkty, które gracz uzyskał. Plik *plansza3.c* zawiera funkcję, które mają sprawdzać czy gracz ukończył grę, czyli czy wszystkie bomby zostały oflagowane, a pozostała plansza została odkryta, po przegraniu gry przypisują planszy wyświetlanej lokalizację bomb, aby użytkownik mógł zobaczyć gdzie się znajdowały oraz jest funkcja która sortuje wyniki z pliku i wypisuje 5 najlepszych graczy.

Kluczowe funkcje

W pliku *main.c* jest tylko jedna funkcja – główna. W niej są zadeklarowane zmienne. Najważniejsze z nich to *plansza_wys* i *plansza_wla*. Pierwsza to plansza, która jest wyświetlana do terminala, a druga to ta, która kryje w sobie, gdzie są bomby i ilość bomb dookoła danego pola. *Plansza_wys* na początku jest pusta i z każdym ruchem odkrywa odpowiednie pole bądź pola przypisując sobie wartość z *plansza_wla*. Na początku tej funkcji sprawdzane jest, przy pomocy *getopt* czy użytkownik podał flagę i ścieżkę do pliku i jeśli tak to program wykonuje się w tym trybie. Jeśli użytkownik uruchomił program bez flagi, to program pomija to i prosi użytkownika o podanie poziomu gry. Następnie przy

pomocy pętli *for* polom *planszy_wys* są przypisywane pola zakryte „[#]”, polom *planszy_wla* są przypisywane pola puste „[]”, a wartość pól tablicy *odwiedzone* są ustawiane na 0 – czyli nie zostały odwiedzone. Następuje wypisanie pierwszej w pętli zakrytej planszy, ilość bomb i liczba punktów do terminala. Tutaj użytkownik podaje swój pierwszy ruch i na jego podstawie dopiero tworzy się *plansza_wla*, aby zapewnić, że w pierwszym ruchu gracz nie trafi na bombę. Za pomocą pętli *while* jest kontynuowane podawanie przez gracza swojego ruchu i wypisywanie planszy. Pętla ta działa póki gracz nie trafi na bombę, wtedy *c = 1* (*c=odkrywanie_planszy*), lub dopóki gracz nie ukończy gry poprzez wygraną. Jak to się stanie to w *if* jest sprawdzane, który to był z przypadków i jeśli to był pierwszy to jest informacja o zakończeniu gry i wyświetlenie gdzie na planszy znajdowały się bomby. Gra w tym momencie jest zakończona i otwarty zostaje plik *wyniki.txt* aby zapisać tam wynik gracza i podany przez niego nick. Następnie otwarty jest jeszcze raz ten plik, tym razem w modzie „r” i przy pomocy funkcji *sortowanie* wypisanych zostają top 5 graczy i ich wyniki.

W pliku *plansza1.c* znajdują się 3 funkcje: *tworzenie_planszy*, *odkrywanie_planszy*, *wypisywanie_planszy*. Pierwsza z nich generuje plansze właściwą na podstawie pierwszego ruchu gracza, zapewniając że nie znajdzie się tam bomba. Przy pomocy *rand()* losowane są w pętli *while* współrzędne punktu na, których można potencjalnie dać tam bombę o ile nie są to współrzędne podane przez gracze lub już nie było tam bomby. Po ustawieniu bomb funkcja sprawdza ile bomb jest dookoła każdego pola w poczwórnej pętli *for* i przy pomocy *switch* przypisuje właściwą wartość. Następna funkcja w tym module to *odkrywanie_planszy*. Ta funkcja odkrywa pole podane przez użytkownika, jeśli będzie to bomba to funkcja zwraca 1, a jeśli będzie to puste pole, to rekurencyjnie odkrywa dalej pola. Jeśli gracz chciał oflagować pole to funkcja oflaguje pole lub je odflaguje, jeśli było oflagowane. Jest zabezpieczenie, żeby nie flagować pól odkrytych. Ostatnia funkcja w tym pliku to *wypisywanie_planszy*. Oczyszcza najpierw terminal, aby wypisać w terminalu za pomocą pętli *for* nową planszę.

Następny moduł to plik pod nazwą *plansza2.c*. Zawiera ona dwie funkcje. Pierwsza z nich to *pozostale_bomby*, która przy pomocy podwójnej pętli *for* przeszukuje ile jest oflagowanych pól na planszy wyświetlanej i odejmuje to od ilości min na tym poziomie, a na końcu wypisuje informację o ilości pozostałych bomb. Druga funkcja to *liczenie_pkt*. Sprawdza ona ile pól zostało odkrytych, czyli ile pól planszy wyświetlanej jest różna od pola zakrytego i oflagowanego. Zwraca liczbę tych pól pomnożonych przez numer poziomu.

Ostatni z modułów to plik *plansza3.c*. Zawiera ona trzy funkcję. Pierwsza z nich, czyli *przegrana* przypisuje planszy właściwej pola bomb. Wywoływana jest tylko kiedy gracz przegra. Następna funkcja to *czy_koniec*. Sprawdza za pomocą podwójnej pętli *for* czy wszystkie pola bomb są już oflagowane a pozostałe odkryte. Jeśli zwróci wartość 1 to oznacza, że gracz już wygrał. Ostatnia już funkcja to *sortowanie*, i ma ona za zadanie

posortować podane do niej wyniki i nicki graczy z pliku, w którym zapisane są wyniki i nazwy wszystkich graczy. Gdy są one już posortowane, wypisuje 5 najlepszych graczy do terminala.

Testy

Program był testowany na bieżąco, uruchamiając go i testując poszczególne funkcję. Jeśli nie wyskakiwał błąd i program działał poprawnie, kontynuowałam pisanie kodu, a jeśli przy kompilowaniu wyskakiwały jakieś błędy, poprawiałam je i testowałam jeszcze raz.

Podsumowanie

Z moich testów wynika, że program wykonuje wszystkie zadania prawidłowo. Jednakże funkcja główna jest dosyć długa, ale nie dałam rady jej bardziej podzielić na moduły. Problem sprawiło mi wykonanie pliku Makefile, a raczej uruchomienie go. Musiałam pobierać odpowiednie pliki, ale koniec końców udało mi się tego dokonać. Jednym z cięższych zadań myślę, że również było sortowanie wyników i wypisanie pięciu najlepszych, aczkolwiek dotarłam do tego jak zrobić to poprawie i działa to prawidłowo.