Sprawozdanie z realizacji projektu

1. Podręcznik użytkownika

Opis zasad gry

Projekt implementuje grę typu "saper", w której użytkownik odkrywa pola na mapie, starając się uniknąć bomb. Celem gry jest odkrycie wszystkich pól niebędących bombami. Użytkownik może oznaczać podejrzane pola flagami.

Sposób uruchomienia programu

Program jest uruchamiany z poziomu konsoli za pomocą komendy:

./a.out [poziom_trudności]

Dostępne poziomy trudności:

- 0 - łatwy (9x9, 10 bomb)
- 1 - średni (16x16, 40 bomb)
- 2 - trudny (16x30, 99 bomb)
- 3 - niestandardowy (wczytywanie mapy z pliku)

W przypadku wyboru poziomu 3 użytkownik musi podać nazwę pliku z mapą.

Opis poszczególnych opcji

Podczas gry użytkownik wykonuje ruchy poprzez podanie komendy w formacie:

[komenda] [wiersz] [kolumna]

Dostępne komendy:

- r - odkrycie pola
- f - oznaczenie pola flagą

Gra kończy się zwycięstwem, gdy wszystkie pola niebędące bombami zostaną odkryte, lub porażką, gdy użytkownik odkryje pole z bombą.

2. Szczegóły implementacji

Podział programu na moduły

Program składa się z kilku modułów:

- main.c - główny plik programu, obsługuje interakcję z użytkownikiem i steruje przebiegiem gry.
- generuj_mape.c / generuj_mape.h - odpowiada za generowanie planszy gry i losowanie rozmieszczenia bomb.

- tablica_graf.c / tablica_graf.h - obsługuje operacje na tablicy graficznej gry oraz interakcję użytkownika.

Interfejs kluczowych funkcji

- void generuj_mape(char mapa[][MAX_SIZE], int sizex, int sizey, int trudnosc); - generuje planszę gry w zależności od poziomu trudności.
- void wybierz_pole(int sizex, int sizey, int *x, int *y, int *opcja); - pobiera ruch użytkownika.
- void sprawdz_pola(char mapa[][MAX_SIZE], char tablica_graf[][MAX_SIZE], int sizex, int sizey, int x, int y, int opcja, int trudnosc, int *wynik); - obsługuje odkrywanie pól i oznaczanie flagami.
- void zapisz_wynik(int wynik); - zapisuje najlepsze wyniki do pliku wyniki.txt.

Opis ważniejszych struktur

Program nie korzysta ze struktur danych, zamiast tego wykorzystuje dwuwymiarowe tablice znakowe do przechowywania stanu gry.

Opis i wynik poszczególnych testów

Testy polegały na sprawdzeniu poprawności:

- Generowania mapy dla różnych poziomów trudności
- Prawidłowego oznaczania pól i reagowania na ruchy użytkownika
- Obsługi warunków końcowych gry (wygrana, przegrana) Wszystkie testy zostały zaliczone pomyślnie.

3. Podział pracy w zespole

Projekt został zrealizowany indywidualnie.

4. Podsumowanie

Czy wszystkie funkcjonalności udało się zaimplementować?

Tak, wszystkie zaplanowane funkcjonalności zostały zaimplementowane.

Trudności i problemy napotkane w trakcie realizacji

Główną trudnością było zarządzanie pamięcią i obsługa interakcji z użytkownikiem w konsoli. Problemem było również zapewnienie, aby pierwsze odkryte pole nigdy nie zawierało bomby.

Sposoby rozwiązania problemów

- Zastosowanie funkcji memset() do inicjalizacji tablicy visited

- Ponowne generowanie mapy, jeśli pierwsze odkryte pole zawiera bombę

Wnioski

Projekt pozwolił na zdobycie doświadczenia w pracy z językiem C, obsłudze plików oraz implementacji mechanik gry logicznej.