

INTRODUZIONE AL PROGETTO

Ho realizzato questo progetto in linguaggio C. Il progetto consiste nel ricreare il gioco arcade Snake. Ho deciso di personalizzare questo gioco con funzionalità come i livelli, le mele di diversi tipi e li account personali.

In cosa consiste il gioco Snake?

L'obiettivo del gioco è riuscire a mangiare tutte le mele presenti nel tabellone fino a quando il serpente non diventa talmente grande da riempire tutta la tabella. Giocando bisogna stare attenti a non scontrarsi contro se stessi o contro i muri perché questo comporterà la perdita della partita. Per rendere il gioco più divertente e competitivo ho aggiunto anche un sistema di livelli, una mela d'orata che vale più punti esperienza, il tutto salvato in degli account personali muniti di password richiesta ad ogni accesso in modo tale da poter caricare i propri record e posizionamenti in classifica.

REQUISITI SOFTWARE E ISTRUZIONI DI COMPILAZIONE

REQUISITI:

Per eseguire il gioco è necessario utilizzare una qualsiasi distribuzione linux con un terminale che supporti i caratteri UNICODE, è anche indispensabile avete il compilatore GCC installato.

GUIDA ALL'AVVIO:

Prima di tutto bisogna scaricare la cartella con il gioco (<https://github.com/xMatte08/snake>), posizionarsi all'interno della cartella e eseguire il comando `"gcc -I./firme corpi/*.c main/main.c -o snake -lm"` per generare l'eseguibile, dopo aver generato l'eseguibile eseguirlo con il comando `"./snake"`.

STRUTTURA DEL CODICE

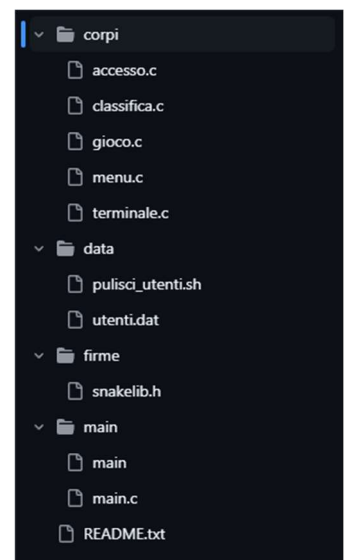
La cartella main contiene il file main.c, il punto centrale del programma, dove vengono richiamate tutte le funzioni implementate nei vari file .c. All'interno della cartella data sono presenti due file: utenti.dat, che è il file binario in cui vengono salvati i dati degli utenti, e pulisci_utenti.sh, uno script che permette di svuotare il file dati in modo semplice e veloce.

Infine, la cartella firme contiene il file snakelib.h, un unico header che raccoglie tutte le dichiarazioni delle funzioni presenti nei vari file .c, oltre alle struct necessarie alla gestione del gioco. Includendo questo header nel main.c, ho potuto rendere tutte le funzioni disponibili nel file principale.

Questa struttura modulare mi ha permesso di mantenere il progetto ordinato, facilitando la lettura del codice e la gestione dei singoli componenti del gioco.

Entrando un po' più nello specifico:

- nel file accesso.c sono presenti tutte le funzioni legate alla gestione degli account e al salvataggio degli utenti nel file .dat;
- nel file classifica.c ci sono le funzioni che gestiscono la classifica e i livelli degli account;
- nel file gioco.c troviamo il vero e proprio cuore del gioco, con tutte le funzioni che gestiscono il loop e la logica per farlo funzionare, oltre alla stampa della matrice usata per rappresentare la tabella e lo snake;
- nel file menu.c ci sono tutti i menù del gioco, come ad esempio quello iniziale;



- infine, nel file terminale.c sono presenti le funzioni che permettono di mettere il terminale in modalità RAW, rendendo getchar() non bloccante, così da avere un gioco fluido senza dover premere invio ogni volta.

Tutti questi file .c sono collegati al file header, che centralizza e organizza tutto il codice rendendolo più gestibile.

FUNZIONALITA' PRINCIPALI DEL GIOCO

Di seguito si descrivono le funzionalità principali offerte dal programma Snake, così come implementate nel progetto:

Login

All'inizio del programma, puoi accedere con un account già esistente. Devi inserire il tuo nome utente e la password. Il programma controlla queste informazioni con quelle nel file degli utenti. Se sono corrette, puoi entrare nel menu principale del gioco. Se non vanno bene, vedrai un messaggio di errore e puoi riprovare. Una volta che ti sei loggato con successo, il programma associa il tuo account alla sessione attuale, così da salvare i tuoi progressi mentre giochi.

Registrazione

Se un utente non ha ancora un account, può registrarsi. Basta selezionare l'opzione di registrazione e inserire un nuovo nome utente e una password. Il sistema verifica se il nome utente scelto è già in uso. Se va bene, viene creato un nuovo account: il programma aggiunge una nuova voce alla lista degli utenti e imposta i punteggi iniziali di mele normali, mele d'oro, livello a zero. I dati vengono subito salvati su file, così la registrazione è sicura. Alla fine, l'utente riceve una conferma che la registrazione è andata a buon fine, e può effettuare il login per cominciare a giocare.

Classifica

La classifica mostra i migliori punteggi dei giocatori, divisi per tipo di punteggio. Dopo aver effettuato il login, l'utente può andare al menu principale e scegliere di vedere le classifiche. Il programma offre tre elenchi diversi:

- Classifica per mele normali: qui ci sono gli utenti ordinati in base al numero massimo di mele normali mangiate in una partita. Chi ne ha mangiate di più è in cima.
- Classifica per mele d'oro: in questa lista, gli utenti sono ordinati secondo il numero massimo di mele d'oro raccolte in una partita. Poiché le mele d'oro sono rare, i punteggi qui sono generalmente più bassi.
- Classifica per livello: qui gli utenti sono ordinati in base al livello massimo raggiunto. Un livello più alto indica che il giocatore è riuscito a giocare di più e a raccogliere più mele senza perdere.

Le classifiche vengono aggiornate ogni volta che si seleziona l'opzione di visualizzazione della classifica, usando i dati attuali degli utenti. Ogni classifica è ordinata dal punteggio più alto al più basso e presenta una tabella con il nome dell'utente e il punteggio.

Gioco

Il gioco principale è, ovviamente, Snake. Dopo che l'utente si registra e accede, può iniziare una nuova partita. All'inizio, il serpente appare sullo schermo con una lunghezza di 3. Una mela normale viene posizionata casualmente nel tabellone di gioco. Il serpente inizia a muoversi automaticamente verso destra. L'utente può usare W A S D per cambiare direzione senza dover premere Invio, grazie a un sistema di input non bloccante. Ogni volta che il serpente mangia una mela normale, si allunga di un segmento e il contatore delle mele raccolte aumenta. Dopo, una nuova mela appare in una posizione casuale. Le mele d'oro, invece, compaiono più raramente, con una percentuale del 10%. Mangiare una mela d'oro darà una ricompensa di 25xp a differenza di una mela normale che ne dà solo 10. Man mano che il serpente mangia mele, la difficoltà aumenta. Il numero di mele mangiate viene mostrato sullo schermo, così il giocatore sa quante mele ha mangiato. La partita continua fino a quando il giocatore non sbaglia. Ci possono essere due modi per finire il gioco: il serpente può colpire un muro o il suo stesso corpo. Il programma controlla dopo ogni movimento se la testa del serpente tocca un ostacolo, se sì, il gioco termina. Alla fine, appare un messaggio di Game Over insieme al punteggio (mele normali, mele d'oro xp al prossimo livello). Dopo che il gioco finisce, il programma confronta i risultati con i record dell'utente nel suo profilo. Se, per esempio, l'utente ha raccolto 15 mele normali e il record precedente era 10, questo sarà aggiornato a 15. Lo stesso vale per le mele d'oro e per il livello. Così, ogni giocatore ha sempre i suoi migliori risultati. Infine, l'utente ritorna al menu principale, dove può decidere se giocare ancora, controllare le classifiche aggiornate o uscire dal programma.

Salvataggio dei dati su file

Una parte importante del progetto è la gestione dei dati degli utenti e dei punteggi tramite il salvataggio su file. Questo sistema permette di tenere traccia delle informazioni tra diverse esecuzioni del programma. Quando si avvia il programma, si apre un file binario che contiene la lista degli utenti registrati e i loro dati (username, password, mele normali, mele d'oro, xp). Ogni utente è registrato su una riga con i campi separati da spazi o virgole. Per esempio, una riga potrebbe essere:

Mario password123 15 2 300

qui Mario è l'username, password123 è la password, 15 sono le mele normali, 2 le mele d'oro e 300 è il numero di xp.

Quando il programma viene avviato, legge il file per riempire un'utenteTemporaneo, che poi verrà passato alla funzione main in modo tale da avere l'utente che in quel momento sta utilizzando il programma:

- Registrazione: quando si registra un nuovo utente, il suo profilo viene aggiunto al file.
- Fine partita: dopo una partita, se i record di un utente cambiano, il programma aggiorna i dati all'interno del file.

Per leggere e scrivere nel file, si usano le funzioni standard della libreria C (fopen, fclose per aprire e chiudere i file, o fread/fwrite per leggere o scrivere in file binari).

In sintesi, il salvataggio su file assicura che:

- Gli utenti registrati siano salvati anche dopo che il programma viene chiuso.
- I record accumulati dai giocatori siano conservati, per avere le classifiche nel tempo.
- Il programma possa ripartire in seguito e continuare a usare e aggiornare i dati esistenti, migliorando così l'esperienza utente.

CONCLUSIONI

Con questo progetto, ho potuto mettere in pratica molte cose che ho imparato sul linguaggio C, creando un programma che funziona e che è anche divertente da utilizzare. Ecco alcune delle cose che ho fatto:

- Ho scritto codice in modo modulare, suddividendolo in vari file per tenerlo in ordine e leggibile.
- Ho utilizzato strutture dati per gestire utenti, punteggi e classifiche.
- Ho lavorato con file binari per salvare i dati degli utenti in modo permanente.

Durante lo sviluppo, ho incontrato anche alcune difficoltà tecniche che mi hanno insegnato molto.

Una delle sfide più grandi è stata far sì che il terminale non bloccasse, così che `getchar()` potesse leggere i tasti senza aspettare l'Invio. All'inizio non sapevo da dove cominciare. Ho cercato soluzioni online, ma alla fine ho deciso di usare `termios` per mettere il terminale in modalità `raw`; è stato il modo più semplice per ottenere un input immediato.

Inizialmente, ho avuto anche qualche difficoltà nella gestione dei file. Ho cercato diverse guide online, ma spesso erano poco chiare. Alla fine, grazie al libro che ci ha dato il professore, ho capito meglio come funzionano la lettura e scrittura su file binari con `fread`, `fwrite`, `fopen` e `fclose`. Una volta chiariti questi concetti, non ho più avuto problemi con il salvataggio dei dati.

In generale, sono molto soddisfatto del risultato. Ho creato un gioco completo, con salvataggio, login, registrazione, classifiche e una logica di gioco che funziona discretamente bene. Questo progetto mi ha fatto capire quanto sia importante testare, organizzare il codice e fare ricerche quando ci sono dei dubbi. È stata un'esperienza utile.