Franceschetti Chiara - Matr. 830909 - Corso F1X

Laboratorio di Architettura degli Elaboratori II - gioco del "15"

Funzionamento generale e interazione con l'utente:

Il codice implementa in linguaggio Assembly il gioco del "15" visualizzando a terminale una griglia di numeri che rappresentano i tasselli del gioco. L'utente può interagire con essi per mezzo di comandi numerici specificati nelle istruzioni iniziali. Ad ogni input fornito dall'utente, la griglia viene aggiornata e visualizzata nuovamente a terminale fino al termine di una partita. Il gioco e' provvisto di un menù iniziale grazie al quale l'utente può scegliere diverse opzioni:

- iniziare una nuova partita
- visualizzare (a terminale) la classifica dei primi cinque giocatori che hanno già giocato una partita
- uscire dal gioco (e quindi terminare l'esecuzione del programma).

Funzionamento del gioco:

Quando l'utente sceglie di giocare una nuova partita vengono richiesti nome e data. Questi vengono utilizzati per generare numeri pseudo-casuale per stabilire la difficoltà della partita e il rimescolamento dei tasselli. La difficoltà è in realtà il numero di mosse che il programma esegue per mischiare i tasselli dello schema: lo schema fornito al giocatore è infatti ottenuto spostando la cella vuota in una posizione casuale per un numero di volte pari alla difficoltà della partita stessa.

La difficoltà della partita stabilisce inoltre il punteggio iniziale, che è 100 volte la difficoltà stessa: quando l'utente esegue una mossa (interagendo con lo schema per mezzo dei comandi 2, 4, 6, 8, 0, come specificato dalle istruzioni iniziali) il punteggio viene decrementato di un punto, che la mossa sia valida o meno, e il programma controlla se lo schema è ordinato. Nel caso in cui lo schema sia ordinato la partita è vinta: il punteggio è incrementato di 500 punti e nome, data e punteggio vengono salvati in classifica; se invece il giocatore termina la partita con il comando 0, il punteggio è decrementato di 500 punti. Infine nome, data e punteggio vengono salvati in classifica.

La classifica è l'elenco dei giocatori con 5 punteggi migliori in ordine decrescente di punteggio: nel caso in cui si giochi una partita con punteggio inferiore al quinto punteggio salvato, il giocatore corrente non viene memorizzato in classifica, in caso contrario è posizionato al quinto posto e la classifica viene riordinata.

Struttura del codice:

Il main costituisce lo scheletro del programma in quanto esegue le funzioni principali chiamando specifiche procedure. Esso:

- stampa le istruzioni iniziali con regole del gioco e comandi utilizzabili
- stampa il menù e salta all'indirizzo del codice che implementa la scelta ricevuta in input servendosi di una jump address table

- se l'utente sceglie di giocare una nuova partita:
    - legge il nome del giocatore e la data
    - calcola la lunghezza del nome servendosi di una procedura
    - si serve del nome e della lunghezza per passarli come argomento ad una procedura che genera un numero pseudo-casuale (ovvero la difficoltà della partita)
    - si serve della data per generare un ulteriore numero pseudo-casuale e chiama la procedura (costituita da procedure annidate) che se ne serve per creare lo schema disordinato iniziale
    - stampa lo schema di gioco chiamando una procedura ed esegue la mossa data in input dall'utente chiamando procedure annidate, ripetendo l'azione fino al termine della partita
    - aggiorna e riordina la classifica per mezzo di una procedura ricorsiva
- se l'utente sceglie di visualizzare la classifica il main chiama la procedura che la stampa a terminale
- se l'utente sceglie un'opzione non valida il programma stampa un messaggio di errore e visualizza nuovamente il menù.

Procedure essenziali:

Procedura 'random':

genera in output un numero pseudo-casuale ricevendo in input due numeri dei quali il primo funge da seed per la generazione del numero casuale e il secondo stabilisce il numero di operazioni xorshift da applicare al seed.

Procedura 'createschema':

riceve in input il base address della matrice che rappresenta lo schema di gioco, il numero di mosse casuali da effettuare (difficoltà della partita) e un numero casuale per determinare la prima mossa. Al suo interno chiama la procedura 'eseguimossa' tante volte quante sono le mosse da effettuare (numero ricevuto in input) aggiornando ogni volta la nuova posizione della cella vuota all'interno della matrice per infine restituirla come output.

Procedura 'eseguimossa':

riceve in input il base address della matrice, un numero casuale e l'attuale posizione della cella vuota all'interno della matrice. In base a quest'ultima verifica quali sono le possibili mosse all'interno dello schema e utilizza quattro registri temporanei come flag di ogni singola mossa. Viene calcolato il numero casuale ricevuto in input in modulo delle mosse disponibili: in questo modo è possibile determinare la mossa da effettuare in base alla seguente tabella.

Mosse disponibili	destra	sinistra	basso	alto
4	0	1	2	3
3	0	1	2	x
3	0	1	x	2
3	0	x	1	2
3	x	0	1	2
2	0	x	1	x
2	0	x	x	1
2	x	0	1	x
2	x	0	x	1

(le x indicano i flag uguali a 0)

In base alla mossa da eseguire viene calcolata la posizione della cella da scambiare con quella vuota. Quest'ultima è passata come argomento, insieme alla posizione della cella vuota e il base address della matrice, ad una ulteriore procedura 'scambia', che scambia le due celle date all'interno della matrice. La procedura 'eseguimossa' infine restituisce in output la nuova posizione della cella vuota all'interno della matrice.

Procedura 'ordinalista':

ordina la classifica utilizzando il metodo ricorsivo. La procedura riceve in input gli indici di riga del primo e dell'ultimo punteggio da ordinare in classifica, in questo modo nel caso in cui la classifica non sia ancora stata riempita con cinque nomi evita di ordinare le righe vuote (con punteggio nullo). La procedura trova la posizione in cui inserire l'ultimo punteggio inserito in classifica, lo inserisce mettendo in ultima posizione il punteggio precedentemente salvato in tale posizione e richiama se stessa sulla sottolista che ha come prima riga quella immediatamente successiva alla riga contenente il nuovo punteggio.

Possibili modifiche

Dato che il codice lavora utilizzando una struttura di memoria matriciale, una possibile modifica riguarda la gestione delle celle: è infatti possibile, anziché considerare la matrice come un vettore, come nel'attuale implementazione, gestire la matrice servendosi di indici di riga e di colonna, calcolando volta per volta la posizione della cella moltiplicando l'indice di colonna per il numero di byte di ogni riga e aggiungendo a questo l'indice di riga moltiplicato per la dimensione di ogni cella. Questo, pur risultando più semplice dal punto di vista della gestione del codice, implica l'impiego di un maggior numero di registri e una maggiore complicatezza di codice per il continuo calcolo della posizione delle celle.

Un miglioramento del programma potrebbe consistere nell'implementare una parte di codice che possa gestire l'inserimento di un input scorretto alla richiesta di nome e data ad ogni nuova partita: qualora infatti venisse inserito, per esempio, una data con formato errato essa verrebbe senza controllo inserita in classifica a fine partita. La soluzione sarebbe quella di implementare una procedura che verifichi che le cifre di giorno, mese e anno siano corrette (quindi che le date siano esistenti) e che anche i simboli '-' siano inseriti correttamente come richiesto. In caso la data inserita non fosse esistente verrebbe visualizzato un messaggio di errore e richiesto il nuovo inserimento della data. Nel caso invece di errato uso dei simboli questi verrebbero automaticamente corretti.