Relazione del Progetto di Programmazione
A.A. 2023/2024
TETRIS

Joel Baccello, Samuele Castiglioni, Mathieu Linty e Sofia Zanon

February 2024

0 Introduzione

Abbiamo suddiviso il lavoro in tre parti: il menù a Joel, la partita del gioco assegnata a Sofia e Mathieu e la classifica a Samuele.

1 Menù

Quando si avvia il programma la prima cosa che si può notare è la schermata di gioco. Essa è stata costruita partendo da un ciclo do-while (contenuto nella funzione doBody) che si ripete fino a quando il giocatore non sceglie di chiudere il gioco con la scelta "Esci" della schermata. Per ogni scelta che il giocatore può prendere nel menù viene richiamata una funzione in grado di eseguire ciò che è stato richiesto:

- Nuova Partita: permette di iniziare una nuova partita di Tetris, con tanto di salvataggio del punteggio e inserimento del nome del giocatore. Viene richiamata la funzione Giuoco e successivamente la funzione per l'inserimento del nome del giocatore, ossia ScritteGameover che a sua volta richiama doInsertName, per la realizzazione della finestra su cui stampare il punteggio ottenuto e gli underscore per rappresentare i caratteri vuoti del nome del giocatore che deve essere inserito, e InsertGamerTag per aggiornare a ogni lettera inserita la stampa della finestra;

- Classifica: permette di visualizzare la cronologia delle partite giocate, infatti vengono stampati i nomi dei giocatori con i corrispettivi punteggi. Viene richiamata la funzione Classifica;

- Esci: permette di uscire dalla schermata e di chiudere il gioco. In questo caso non viene richiamata una funzione, semplicemente viene assegnato un valore alla variabile booleana che funge da condizione di terminazione del ciclo in modo che la ripetizione venga interrotta.

Per realizzare la stampa della schermata sono state sfruttate le window di ncurses, che sono state definite all'interno della classe Wd, i cui metodi più interessanti e rilevanti sono:

- il costruttore: che permette di creare a tutti gli effetti una window ricevendo come parametri l'altezza, la larghezza della finestra e le coordinate x,y di dove deve essere collocata sullo schermo;
- COPY: riceve una stringa come parametro, in modo che questa stringa possa essere poi stampata su schermo grazie alla prossima funzione descritta;
- PRINT_PHRASE: permette di stampare su schermo la frase che è stata registrata all'interno della classe.
- BOX: sfrutta la funzione di ncurses "box" per stampare il riquadro della finestra;
- REFRESH: sfrutta la funzione "wrefresh" di ncurses per aggiornare la finestra che viene passata come parametro, operazione fondamentale per ottenere la corretta stampa delle finestre.

In totale sono state utilizzate tre window per la schermata: la finestra "di base" ossia stdscr, la finestra per il titolo (strettamente legata a una libreria scritta appositamente per permettere la stampa del titolo) e, infine, una finestra per il menù, contenente le scelte tipiche che un menù di gioco contiene. Fondamentali per la riuscita della stampa sono le seguenti funzioni:

- Center: riceve come parametro la larghezza della finestra su cui è stampata una seconda finestra che funge da secondo parametro della funzione. Essa permette di stampare la finestra più piccola al centro, rispetto all'asse orizzontale, della finestra più grande (quella di cui è stata passata la larghezza);
- N: riceve come parametro il numero delle colonne della finestra che si sta considerando e il numero di sezioni verticali in cui si vuole che la finestra venga suddivisa. In questo modo la funzione restituisce il numero di colonne necessarie per suddividere la finestra in un certo numero di sezioni.

Infine, la libreria Title è stata progettata per realizzare un titolo che scalasse in grandezza in base alla grandezza del monitor su cui si sta giocando, attraverso dei calcoli aritmetici. Per farlo, è stato necessario definire una funzione per ciascuna lettera che compone il nome del gioco.

2 Gioco

Classe Griglia:
Ad ogni ciclo di animazione conserva e aggiorna lo stato della partita tramite un array bidimensionale di interi. I valori attribuibili a ogni elemento sono 0, che corrisponde ad una cella vuota (‘’ ‘) , e 1, che corrisponde al blocchetto base di un tetramino (‘{}’).

I campi della classe sono:

- due interi che contengono il numero di righe e di colonne;

- l'array bidimensionale di dimensione 22x32 con ogni valore inizializzato a 0 nel costruttore. Visto che in ncurses i caratteri verticali hanno ampiezza doppia rispetto ai caratteri orizzontali e che il blocchetto base è dato da due caratteri, ogni ascissa può solo assumere valori pari;
- un puntatore alla finestra di gioco;
- un intero per il punteggio.

I metodi principali sono:

- Update: disegna sulla finestra di gioco i caratteri corrispondenti al valore di ogni cella dell'array;
- SetState: prende come parametri le coordinate (y, x) e lo stato da attribuire ad esse e modifica la cella corrispondente nell'array controllando che la x sia pari;
- isRowFull: determina se ogni cella della riga passata come argomento è impostata ad 1, ovvero se la riga è piena;
- MoveRowsDown: sposta tutte le righe al di sopra della riga di input in basso di uno;
- ClearRow: imposta tutte le celle di una riga a 0;
- DeleteRow: chiama ClearRow e MoveRowsDown sulla riga presa in input;
- ControlRows: controlla ogni riga della griglia: se è piena la cancella e aumenta il punteggio (in base al numero di righe cancellate contemporaneamente).

Classe Tetratmino:
È la classe padre da cui ereditano le diverse forme dei tetramini.

Campi:

- due interi per le coordinate y e x riferite alle celle della griglia. La y è inizializzata a 0 in modo che il tetramino cominci la sua discesa dalla riga più in alto. La x è calcolata affinché la sua posizione sia centrale nella finestra di gioco;
- un booleano HasReachedEnd inizializzato a false, utile a controllare se il tetramino si sia fermato;
- un puntatore alla finestra di gioco;
- un puntatore alla griglia per accedervi direttamente senza doverla copiare in un campo aggiuntivo.

Metodi:

- Costruttore: inizializza i puntatori alla finestra di gioco e alla griglia con quelli passati come argomento;

- MoveDown: imposta la velocità di stampa di ncurses al valore della costante arbitraria SPEED. Poi se il tetramino non è arrivato all'ultima riga viene fatto scendere di uno, altrimenti viene impostato a true il valore di HasReachEnd. Il parametro 'n' serve a compensare la diversa altezza dei tipi di tetramino;

- isGameOver: ritorna true se il tetramino, fermandosi, va oltre il margine superiore (la partita è finita).

Classe Quadrato:
È classe figlia di Tetramino. Non presenta ulteriori campi rispetto a Tetramino.
Metodi:

- Costruttore: eredita i campi dalla classe padre;

- Display: sfruttando il metodo SetState della griglia, disegna sulla schermata di gioco il tetramino con forma quadrata;

- Clear: rende non visibile il tetramino impostando a 0 le celle in cui è contenuto;

- CheckCollision: verifica se ci sono tetramini immediatamente sotto il tetramino in caduta. In questo caso imposta a true il valore di HasReachedEnd;

- CheckLeftEdge: controlla se il tetramino è a contatto col bordo sinistro della schermata di gioco o se ci sono altri tetramini alla sua sinistra;

- CheckRightEdge: controlla se il tetramino è a contatto col bordo destro o se ci sono altri tetramini alla sua destra;

- Update: gestisce il movimento del tetramino sulla schermata di gioco. Chiama Clear e controlla l'input preso da tastiera: se è la freccia a sinistra o a destra verifica che il tetramino possa muoversi in quella direzione e ne aggiorna la posizione. Successivamente chiama CheckCollision, MoveDown della classe padre e infine aggiorna nuovamente la posizione del tetramino.

Classe Linea:
Identica alla classe Quadrato, eccetto alcune accortezze.

Campi: un booleano (isRotated) per verificare se il tetramino è ruotato, inizializzato a false.

Metodi:

- Display: disegna la Linea in orizzontale se isRotated è false, in verticale altrimenti;

- Clear: differenzia i casi in cui il tetramino è ruotato oppure no;

- CheckCollision: differenzia i casi in cui il tetramino è ruotato oppure no;

- CheckLeftEdge: differenzia i casi in cui il tetramino è ruotato oppure no;

- ChehckRigthEdge: differenzia i casi in cui il tetramino è ruotato oppure no;
- cantRotate: a seconda dell'orientamento in cui si trova il tetramino, determina se esso possa ruotare oppure no, controllando le celle adiacenti;
- Rotate: chiama Clear e nega il valore di isRotated, ruotando di fatto il tetramino;
- Update: vi è il controllo aggiuntivo dell'input freccia in su con il quale, se possibile, si ruota il tetramino.

Giuoco:
Questa funzione gestisce il flusso del gioco tramite un ciclo while e in particolare:

- Crea la finestra di gioco e la griglia;
- Crea le finestre per la visualizzazione del tetramino successivo e del punteggio;
- Inizializza un Quadrato e una Linea;
- Inizializza due interi con un valore casuale (0 o 1) che determina il tipo del tetramino corrente e di quello successivo: a 0 corrisponde un Quadrato; a 1 una Linea;
- Dichiara un intero per conservare l'input;
- Ciclo while:
  - Finché non si perde la partita, il tetramino corrente fa il suo corso. Quando arriva in fondo se ne crea uno nuovo di tipo predeterminato. Avvenuto ciò, si chiama ControlRows per controllare se il giocatore ha fatto dei punti e si determina un nuovo valore per il tetramino successivo;
  - Chiama ClearRow sulle prime tre righe della griglia, non visibili sulla schermata di gioco, che hanno lo scopo di facilitare il controllo della fine della partita;
  - Chiama griglia.Update per aggiornare lo stato della partita;
  - Disegna nella finestra apposita la forma del tetramino successivo con le funzioni drawNextQuadrato e drawNextLinea;
  - Aggiorna e scrive il punteggio nella finestra designata.

3 Classifica

La realizzazione della classifica ha richiesto due fasi distinte: la realizzazione della sua meccanica e la realizzazione della sua resa grafica.
Per quanto concerne la prima, i punteggi vengono salvati su file attraverso la libreria fstream; dal file vengono salvati su una struct (nella funzione TxtToArr) e ordinati - in modo decrescente - utilizzando un bubble sort (nella funzione OrdinaClassifica). Per la visualizzazione, abbiamo utilizzato ncurses (nella funzione MostraClassifica).

Abbiamo utilizzato anche la libreria cstring per gestire gli array di char e la libreria cmath, le cui funzioni floor e log10 ci hanno consentito, nella funzione MostraClassifica, di determinare il giusto numero di spazi da stampare dopo gli score (ciò è indispensabile perché, altrimenti, gli score delle pagine già visualizzate, che non vengono sovrascritti, verrebbero di nuovo stampati!).
La struct User ha due elementi: un char name[60] e un int score, nei quali vengono salvati rispettivamente il nome del giocatore e il punteggio che ha ottenuto nella partita.
Le funzioni utilizzate sono le seguenti:

- int CountLines(ifstream &tetris_scores): usa fstream per contare le linee del file "tetris_scores.txt" - il file che salva tutti i punteggi. L'intero che ritorna coincide col numero di punteggi salvati, perciò viene passato come parametro alle altre funzioni;

- void TxtToArr(char NomeFile[], User a[], int totLines): usa fstream per passare nomi e punteggi su un array di struct User;

- void OrdinaClassifica(struct User a[], int size): usa un bubble sort per ordinare la classifica, in base al punteggio, in senso decrescente;

- void InserisciScore(char username[], int score): usa fstream per scrivere il nome e il punteggio sul file "tetris_scores.txt" (viene chiamata a fine partita, così da salvare il punteggio ottenuto);

- void MostraClassifica(User a[], int totLines): usa ncurses per stampare a schermo la classifica. Definisce le dimensioni della window e il numero di righe che stampa per ciascuna pagina; stampa la classifica attraverso un ciclo while e gestisce lo scorrimento della classifica attraverso uno switch:
  - ciclo while: le linee vengono stampate attraverso la funzione mvwprintw. Attraverso l'opzione A_REVERSE della funzione wattron, una delle linee della classifica risulta evidenziata, cosicché l'utente possa scrollare la classifica. Prevede anche un input dall'utente: premere il tasto esc causa l'interruzione del ciclo while e fa tornare al menu iniziale, gli altri casi sono gestiti dallo switch.
  - switch: gestisce i casi in cui l'utente preme la freccia in su o quella in giù, scorrendo la classifica nella rispettiva direzione. Esso include anche i casi in cui la linea selezionata sia quella più in alto (facendo visualizzare la pagina precedente o, nel caso in cui ci si trovi alla prima pagina, impedendo di salire ulteriormente) e il caso in cui la linea selezionata sia quella più in basso (analogo all'altro caso);
- void Classifica(): questa funzione chiama le altre in successione: apre il file "tetris_score.txt"; passa il valore di ritorno di CountLines alle altre funzioni; chiude il file e, infine, visualizza la classifica attraverso la funzione MostraClassifica.