

Relazione Progetto: XTetris

Alex Narder matricola:892300

June 23, 2022

1 Prefazione

Questo progetto è stato realizzato in autonomia secondo lo standard ANSI C. Il suo sviluppo è iniziato da subito e il codice è stato riscritto più di una volta nel corso del tempo.

2 Struttura del progetto

Il filesystem del progetto è semplice, nella directory principale sono presenti due directory; Documents/ e Relation/. La directory Documents/ contiene la documentazione generata attraverso Doxygen, mentre nella directory Relation/ è presente la relazione del progetto in formato pdf scritta tramite l'utilizzo di Latex. Inoltre nella directory principale troviamo i file sorgente, title.c contiene le funzioni grafiche dell'avvio quali la stampa del titolo e delle istruzioni, il file func.c contiene le istruzioni che poi verranno richiamate in ogni modalità. Le modalità vengono definite in mod.c, al suo interno troviamo le funzioni che regolano l'andamento del gioco per modalità singleplayer e multiplayer, in fine vi è il file main.c ovvero il punto di partenza di tutto il gioco.

Per l'avvio del gioco è sufficiente utilizzare il comando make, il Makefile compilerà i file in modo autonomo e poi avvierà il gioco.

3 Funzioni principali

Il progetto è strutturato in quattro file con estensione .c, il file che contiene le funzioni essenziali si trova in func.c. All'inizio del file si trovano le

librerie usate, e subito dopo troviamo le matrici dei tetramini. Per permettere il movimento del pezzo all'interno della tabella sono state utilizzate due variabili (`movx` e `movy`) che vengono decrementate e incrementate a seconda degli input ricevuti da tastiera, questa funzione è stata realizzata sia per il giocatore 1 che per il giocatore 2, il giocatore 1 (protagonista anche della modalità `singleplayer`) ha la funzione `movep1()` mentre il giocatore 2 `movep2()`, entrambe svolgono lo stesso compito. Per evitare che i pezzi si scontrino con le pareti è stata creata la funzione `otherblockcollision()`, questa funzione viene anche impiegata per evitare che i pezzi si sovrappongano tra di loro.

I tetramini che vengono stampati nella tabella fanno parte di essa, il funzionamento del gioco si basa sulla sostituzione dei caratteri all'interno della matrice, questa sostituzione avviene attraverso la funzione `copyblock()`, che copia uno dei tetramini randomici all'interno di un tetramino generale (`tetr1`), e attraverso `changetab()`. `Changetab()` sostituisce i caratteri vuoti all'interno della tabella con i caratteri del tetramino, la tabella viene successivamente ristampata con la funzione `Tab()` e i pezzi rimasti sulla tabella dalla scorsa stampa vengono cancellati utilizzando la funzione `cleartab()`.

Nella modalità `singleplayer` viene stampata una sola tabella, mentre nella modalità `multiplayer` viene stampata sia la tabella del `p1` che del `p2` e i turni sono alternati attraverso uno scambio che avviene ogni volta che un tetramino tocca il suolo. La funzione `stoptetr()` è quella che permette la collisione con il suolo e con tutto ciò che è sotto il tetramino.

A questo punto ci si può muovere, ci sono le collisioni e i pezzi rimangono attaccati al suolo/altri tetramini. Per la rimozione delle righe utilizzo più di una funzione; `removelines()` è la funzione principale che permette di rimuovere la riga piena e sostituirla con dei caratteri vuoti, poi questa funzione viene richiamata all'interno di `checkscorep1()` e `checkscorep2()`, rispettivamente per primo e secondo giocatore queste istruzioni abbassano di 1 riga tutto il campo partendo dalla riga vuota creata da `removelines()`. Ogni volta che la funzione rimuove delle righe un counter aumenta per verificare se più di una riga è stata cancellata con un solo pezzo e in quel caso la funzione `dupanother()` conferisce i punti adeguati. In conclusione per vincere in `singleplayer` è necessaria la funzione `win()`, appena tutti i pezzi danno somma 0 il gioco si conclude e la partita è vinta dal giocatore. Nella modalità `multiplayer` i giocatori vincono in base al punteggio più alto, la vittoria viene determinata nel momento in cui i pezzi finiscono, oppure quando uno dei due giocatori sfiora verso l'alto con un tetramino. Quando uno dei tetramini, sia nella modalità

singleplayer che multiplayer, esce dal campo viene dichiarato gameover, attraverso la funzione `returngameover()` per il singleplayer, e `mod2gameover` per il multiplayer.

4 Difficoltà incontrate

Le maggiori difficoltà incontrate sono principalmente legate ad alcune funzioni; la funzione `randomblock()` mi ha dato molti problemi in quanto andava in contrasto con la decrementazione del numero di tetramini rimasti, ad ogni numero generato la funzione diminuiva il numero di tetramini rimasti fino ad andare in negativo, nonostante l'utilizzo di statement vari il problema non svaniva, quindi ho creato 3 funzioni che hanno reso più semplice il tutto, una funzione per il numero random, un check della quantità di tetramini e in fine una funzione che selezionava il tetramino. Un problema che non sono riuscito a risolvere è quello dell'ultimo tetramino, infatti una volta posizionato il penultimo tetramino il gioco finisce.

Slegato dai problemi delle funzioni è stato il "problema" dello standar ANSI C che non ha permesso l'utilizzo di alcune librerie grafiche molto interessanti.