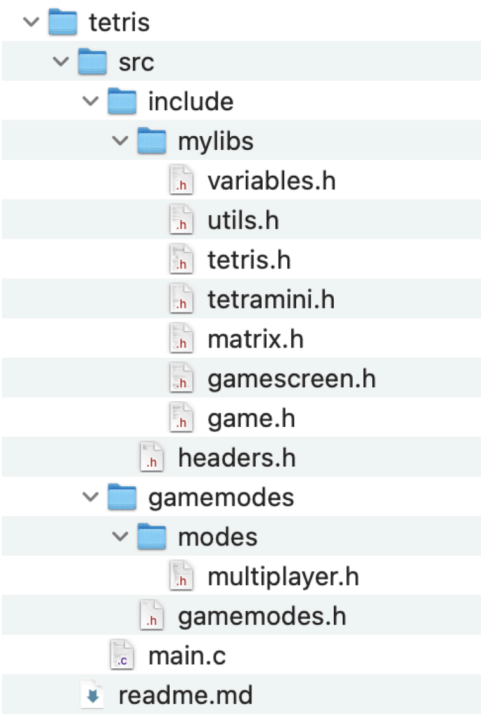
X-Tetris

Versione avanzata del gioco originale del Tetris. Il progetto è stato sviluppato in C con il supporto di poche librerie esterne, di seguito una rappresentazione grafica della struttura del progetto con annesse delle piccole descrizioni per i file/cartelle più importanti



main.c - viene semplicemente gestito il menù per scegliere la modalità di gioco

include/mylibs - in questa cartella ci sono le librerie utilizzate per sviluppare il progetto

tetris.h: contiene le funzioni per gestire la logica del gioco

tetramini.h - contiene la rappresentazione dei vari pezzi di gioco con le loro relative impostazioni

mutiplayer.h - file in cui è effettivamente implementato, il gioco (game loop, eventi, ecc…)

headers.h - wrapper generale per tutte le librerie, unico file incluso in main.c

Idea di sviluppo

Il concetto principale sul quale mi sono basato per sviluppare questo gioco è stato il creare delle strutture facili e intuitive che mi permettessero di organizzare le varie caratteristiche di ogni elemento in modo ordinato e facilmente interpretabile. Per questo nel sorgente sono presenti principalmente quattro strutture:

- **Matrix**: che contiene al suo interno informazioni quali numeri di colonne/righe e un array di interi che rappresenta la matrice stessa
- **Tetramino**: rappresenta il tetramino con tutte le sue caratteristiche (vedi sotto per maggiori info)
- **Tetris**: che rappresenta quella che io definisco “sessione di gioco”, al cui interno sono contenute tutte le informazioni relative alla sessione, come per esempio, la matrice di gioco, i punti, l’ultimo pezzo inserito, ecc…
- **Game**: contiene tutte le varie sessioni di gioco e informazioni come la modalità selezionata, l’ultimo input da tastiera, ecc…

Tetramino

Ogni tetramino è composto dalle seguenti informazioni:

- **code**: un codice identificativo
- **Matrix**: la matrice che lo rappresenta (non tutti i tetramini sono rappresentati in una matrice 3x3 ad esempio)
- **width/heigth**: rappresenta la lunghezza/altezza del tetramino
- **offsetX/offsetY**: rappresenta lo spazio vuoto tra le coordinate (0, 0) e l'inizio del tetramino

```
int I[] = {
    0, 1, 0, 0,
    0, 1, 0, 0,
    0, 1, 0, 0,
    0, 1, 0, 0};
```

Width: 1 height: 4
offsetX: 1 offsetY: 0

Quando il tetramino viene **ruotato** questi dati al suo interno **cambiano**. E sono stati utilizzati al fine di gestire meglio l'inserimento e la gestione delle collisioni.

Precisazioni:

- Tutti i tetramini di **default** sono salvati in un array statico e immutabile, questo perché quando un giocatore decide di cambiare pezzo, viene copiata nella sua sessione di gioco, la corrispondente struttura di default.
- Quando viene ruotato un tetramino, offsetX e offsetY vengono calcolati

Struttura della gestione del gioco (mutiplayer.h)

Dopo che il giocatore ha selezionato la modalità di gioco dal menù principale, potrà subito giocare, il programma resterà in attesa dell'input dell'utente per poi una volta interpretato a modo quest'ultimo, effettuare le relative elaborazioni stampando sul terminale il risultato. In sostanza il loop di gioco principale si preoccupa di:

1. Restare in attesa dell'input dell'utente
2. Elaborare l'input (muovere il pezzo, ruotarlo, cambiarlo, inserirlo, ecc)
3. Mostrare all'utente lo stato corrente della propria sessione di gioco

Precisazione

In questo file viene gestito il "Task" di cambio del tetramino tramite la funzione **switchTetraminoTaskM**(Game* game, int **incrementType**) nel loop principale di gioco, inoltre la stessa funzione viene utilizzata nella gestione di un altro task (l'inserimento di un tetramino) per gestire lo switch automatico del pezzo nella sessione di un altro giocatore nel caso il giocatore corrente inserisca nella propria mappa l'ultimo pezzo disponibile di un determinato tipo di tetramino, dato che il "pull" di tetramini è condiviso. Infatti il flag **incrementType** serve proprio a gestire questo comportamento, nel caso fosse settato a 1 (come avviene nel loop di gioco) allora verrà effettivamente modificato il tipo di tetramino selezionato dal giocatore che deve inserire un nuovo pezzo nella sua mappa.

Inserimento tetramino e gestione delle collisioni

Quando devo inserire un nuovo tetramino nella mappa ho bisogno di sapere le coordinate dove devo inserirlo, quindi quello che è stato fatto è controllare se nella posizione desiderata dall'utente(asse x) sia effettivamente possibile inserire il pezzo desiderato, quello che viene fatto nell'ordine è:

- Controllare innanzi tutto che il pezzo non sfori la mappa (rispetto alla lunghezza)
- Controllare rispetto alla posizione sull'asse delle x del tetramino, la **possibile** posizione dalla quale iniziare a inserire il tetramino rispetto all'asse delle Y
- Controllare se la matrice del tetramino e la sotto-matrice della mappa (considerando la x in input dell'utente e la y calcolata precedentemente) hanno punti in comune e quindi ci sarebbe una collisione che non permetterebbe l'inserimento del pezzo.

In tutte queste fasi vengono utilizzati per i calcoli i vari dati contenuti nella struttura del tetramino (gli offset, le lunghezze, ecc…).

Ovviamente quando devo controllare se la matrice del tetramino "interseca" la sotto-matrice della mappa di gioco, prima controllo fino a dove il pezzo potrebbe effettivamente "scendere".

"Accorgimenti" utili

- Per stampare più mappe contemporaneamente nel terminale ho utilizzato ncurses che ha semplificato di molto il lavoro, permette infatti di spostare il cursore del mouse nel terminale, nelle coordinate desiderate e cominciare a stampare da lì.

Principali difficoltà incontrate

La principale difficoltà incontrata è stata gestire le collisioni, ha richiesto un po' di più tempo delle altre funzionalità ma niente di impossibile, altre difficoltà incontrate importanti non ce ne sono state, se non l'installazione di ncurses (vedi sotto).

Problemi noti/riscontrati

- ncurses, questa libreria grafica su windows potrebbe dare qualche problema (nonostante sia installata), su unix tutto bene.
- La gestione dei colori e delle varie combinazioni di colori di ncurses non funziona benissimo.
- Su windows la schermata della selezione della modalità di gioco "balla" mentre su unix non ci sono problemi.

Filiberto Abbatangelo 891843