



**27 GENNAIO 2022**

**BY JACOPO COSTANTINI, ALVISE SILVESTRI & MATTEO ZAMBON**

# **THE XTETRIS PROJECT**

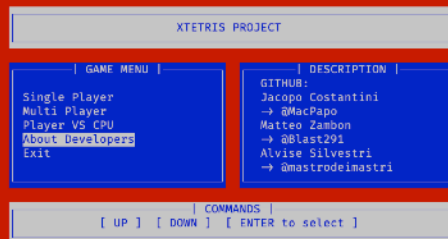


**Una alternativa al classico Tetris** 

# XTETRIS

**X**tetris è stato il nostro primo progetto all'interno del mondo universitario; durante tutto il corso dello sviluppo abbiamo impiegato tutte le nostre energie mentali e creative per renderlo il più originale possibile.

Infatti, a nostro modo di vedere, la musica e la libreria grafica utilizzata, gli danno quel tocco in più...



## STRUTTURA:

il progetto è suddiviso complessivamente in 14 files tra .c e .h ed è basato sulla libreria grafica “ncurses” che permette di suddividere il terminale in finestre personalizzabili; infatti quest’ultime ci hanno permesso di creare delle interfacce di gioco differenti tra le varie modalità ed un menu interattivo .

Le finestre di gioco si aggiornano dinamicamente in base a delle matrici di interi contenenti le varie posizioni dei tetramini; le loro celle possono assumere valori da 0 a 6, così da permetterci di colorare diversamente i diversi pezzi.

Il cuore del gioco parte dal file main.c che poi andrà a richiamare, a seconda della scelta fatta nel menu, la funzione corrispondente alla modalità di gioco. Entrando all’interno di essa viene chiamata una funzione che inizializza il terminale a seconda di determinati parametri dichiarati in un file .h (commonConfig.h) contenente tutti i vari parametri necessari al programma per essere eseguito, inoltre, all’interno di questo file, vengono definite funzioni trasversali a tutte le modalità di gioco che poi verranno scritte all’interno del corrispondente file .C .

All’interno delle varie modalità di gioco la mossa viene fatta tramite la funzione wgetch(), presente all’interno della Libreria di “ncurses”, che restituisce un valore

intero inserito poi all'interno di uno switch che ci permette di svolgere azioni diverse a seconda della scelta fatta.

Alla fine del sopracitato switch, vengono chiamate delle funzioni in grado di controllare se ci sono delle condizioni di gameover oppure se un giocatore ha fatto punto, con la conseguente eliminazione delle righe piene.

Se un giocatore ha perso o ha finito i pezzi, il programma stamperà a video il suo punteggio e, tramite la pressione di un qualunque tasto, verrà riportato al menu di scelta.

Per quanto riguarda i giocatori, è stata creata una struct "player" contenente il punteggio, la finestra di gioco e la sua relativa matrice. all'inizio di ogni partita vengono dichiarati e poi inizializzati, tramite una funzione "addplayer()", tutti i giocatori necessari.

I vari tetramini presenti nel gioco vengono dichiarati tramite la struct tet che ne definisce il tipo e l'orientazione. la loro stampa a schermo è resa possibile da un'array tridimensionale di tipo tet\_location (struct che prende come dati la riga e la colonna).

La funzione che implementa la CPU è stata suddivisa in varie funzioni che prima controllano la miglior riga da completare, poi lo spazio più piccolo in essa e alla fine, tramite una scelta random, viene fatto scendere uno dei pezzi che può completare quel vuoto.

## **ORGANIZZAZIONE:**

Lo svolgimento del progetto è stato portato avanti da tutti i componenti del gruppo in maniera trasversale, per far in grado che ogni persona sapesse che cosa fa ciascuna parte del programma, e avesse contribuito a scriverla e renderla funzionante.

## MAGGIORI PROBLEMATICHE:

Inizialmente è stato un po' difficoltoso adattarsi alla sintassi e alle funzioni che permettevano un uso adeguato della libreria ncurses, in quanto non ci eravamo mai interfacciati con essa.

Nella fase di sviluppo delle modalità singleplayer e player vs player non abbiamo incontrato grandi difficoltà per quanto riguarda l'utilizzo di ncurses, mentre, è stato leggermente più complesso rendere le finestre di gioco dinamiche a seconda dei valori presenti all'interno della matrice, infatti abbiamo dovuto ripensare le spaziature, in quanto i tetramini non sembravano rappresentati da celle quadrate, ma bensì rettangolari.

La parte che ha richiesto lo sforzo maggiore, però, è stata la progettazione della scelta del computer, in quanto non avevamo mai progettato un'intelligenza artificiale, e

creare un algoritmo che facesse delle mosse in base al campo di cui predisponessa non è stato affatto semplice ma, alla fine, siamo riusciti a venirne a capo.

