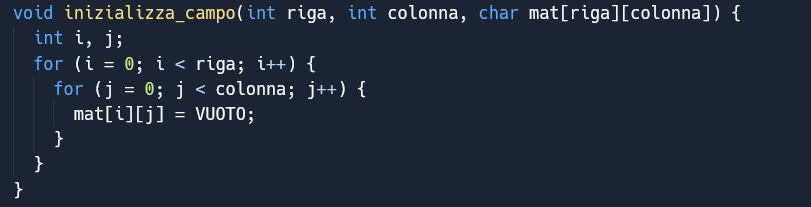
|  |
| --- |
| Forza 4 : Manuale di istruzione |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Le librerie che abbiamo incluso sono librerie standard in C che forniscono funzionalità di base per lo sviluppo di programmi. Ecco una breve descrizione di ciascuna:   1. stdio.h: Questa libreria fornisce le funzioni per l'input e l'output standard, come printf() e scanf(), che sono utilizzate per la stampa e la lettura da console. 2. stdlib.h: Questa libreria fornisce molte funzioni di utilità, come allocazione dinamica di memoria, conversione di stringhe in numeri, e altre funzioni di gestione della memoria e di conversione. 3. string.h: Questa libreria fornisce molte funzioni per la manipolazione di stringhe, come strlen(), strcpy(), strcat(), strcmp(), che sono utilizzate per la gestione delle stringhe. 4. unistd.h: Questa libreria fornisce una serie di funzioni relative al sistema operativo, come fork(), exec(), sleep(), pipe(), dup(), che consentono di interagire con il sistema operativo e di gestire processi, file descriptor, e altre risorse di sistema. |  | |
| 2 | Questi define sono utilizzati per definire costanti simboliche nel programma. Ecco una breve descrizione di ciascuno:   1. righe: Definisce il numero di righe del campo di gioco. Nel gioco del forza 4, il campo di gioco è composto da un certo numero di righe. 2. colonne: Definisce il numero di colonne del campo di gioco. Questo determina la larghezza del campo di gioco. 3. VUOTO: Rappresenta il carattere che indica una posizione vuota nel campo di gioco, cioè una posizione in cui nessun giocatore ha inserito una pallina. 4. ROSSO: Rappresenta il carattere che indica una pallina inserita dal giocatore rosso nel campo di gioco. 5. GIALLO: Rappresenta il carattere che indica una pallina inserita dal giocatore giallo nel campo di gioco. 6. vincita: Specifica il numero di palline dello stesso colore che devono essere allineate in modo consecutivo (orizzontalmente, verticalmente o diagonalmente) per ottenere la vittoria. In forza 4, il giocatore che riesce a formare una fila di quattro palline dello stesso colore vince la partita. |  | |
| 3 | Questi define definiscono sequenze di escape ANSI per cambiare il colore del testo o dello sfondo nella console. Ecco una breve descrizione di ciascuno:   1. ANSI\_COLOR\_RED: Cambia il colore del testo in rosso. 2. ANSI\_COLOR\_GREEN: Cambia il colore del testo in verde. 3. ANSI\_COLOR\_YELLOW: Cambia il colore del testo in giallo. 4. ANSI\_COLOR\_BLUE: Cambia il colore del testo in blu. 5. ANSI\_COLOR\_MAGENTA: Cambia il colore del testo in magenta. 6. ANSI\_COLOR\_CYAN: Cambia il colore del testo in ciano. 7. ANSI\_COLOR\_ORANGE: Cambia il colore del testo in arancione. 8. ANSI\_COLOR\_B\_RED: Cambia il colore dello sfondo in rosso. 9. ANSI\_COLOR\_B\_GREEN: Cambia il colore dello sfondo in verde. 10. ANSI\_COLOR\_B\_YELLOW: Cambia il colore dello sfondo in giallo. 11. ANSI\_COLOR\_GRAY: Cambia il colore del testo in grigio. 12. ANSI\_COLOR\_RESET: Ripristina il colore del testo e dello sfondo ai valori predefiniti. 13. ANSI\_RESET\_ALL: Ripristina tutti i colori e gli stili al valore predefinito.   Queste sequenze di escape ANSI sono utilizzate per migliorare l'estetica del testo o per evidenziare determinate informazioni nella console. Ad esempio, potresti utilizzarle per colorare il testo di output in base a determinate condizioni o per migliorare la leggibilità del tuo codice. |  | |
| 4 | Queste sono dichiarazioni di funzioni in linguaggio C. Ecco una breve descrizione di ciascuna:   1. inizializza\_campo: Questa funzione inizializza il campo di gioco. Prende come argomenti le dimensioni del campo (numero di righe e colonne) e una matrice bidimensionale che rappresenta il campo di gioco. Serve per preparare il campo prima che inizi la partita. 2. stampaCampo: Questa funzione stampa il campo di gioco attuale. Prende come argomenti le dimensioni del campo (numero di righe e colonne) e la matrice bidimensionale che rappresenta il campo di gioco. Questa funzione visualizza il campo di gioco corrente sulla console. 3. controlloVincita: Questa funzione controlla se un giocatore ha vinto la partita. Prende come argomenti il carattere che rappresenta il giocatore, le dimensioni del campo (numero di righe e colonne) e la matrice bidimensionale che rappresenta il campo di gioco. Restituisce un intero che indica se il giocatore ha vinto o meno. 4. inserisciPallina: Questa funzione inserisce una pallina nel campo di gioco in una determinata colonna. Prende come argomenti la colonna in cui inserire la pallina, il carattere che rappresenta il giocatore che sta inserendo la pallina, le dimensioni del campo (numero di righe e colonne) e la matrice bidimensionale che rappresenta il campo di gioco. Restituisce un intero che indica se l'inserimento è avvenuto con successo. 5. inserisciNomiGiocatori: Questa funzione chiede all'utente di inserire i nomi dei giocatori. Prende come argomenti due puntatori a stringa che rappresentano i nomi dei giocatori. Serve per acquisire i nomi dei giocatori prima che inizi la partita. 6. visualizzaRegole: Questa funzione visualizza le regole del gioco. Non richiede argomenti e serve per informare i giocatori sulle regole del gioco prima di iniziare a giocare. 7. visualizzaCrediti: Questa funzione visualizza i crediti o le informazioni sui creatori del gioco. Non richiede argomenti e può essere utilizzata per dare credito agli sviluppatori del gioco.   Queste funzioni sono progettate per gestire diversi aspetti del gioco del forza 4, come l'inizializzazione del campo, la gestione delle mosse dei giocatori, il controllo della vittoria e la visualizzazione delle informazioni aggiuntive come regole e crediti.  IL MAIN |  |  |
| 5 | Questo codice implementa un gioco Forza 4 in linguaggio C. Ecco una descrizione dettagliata:   1. **Definizione delle variabili e delle costanti**:    * Viene dichiarata una matrice bidimensionale campo per rappresentare la griglia di gioco.    * Due array di caratteri giocatore1 e giocatore2 per memorizzare i nomi dei giocatori.    * turno per tener traccia del turno attuale (0 per il giocatore 1, 1 per il giocatore 2).    * scelta per memorizzare l'input del menu.    * colonna per memorizzare la colonna scelta dal giocatore durante il gioco. 2. **Menu principale**:    * Un ciclo while viene utilizzato per visualizzare un menu finché il giocatore non sceglie di iniziare la partita.    * Il menu offre opzioni per inserire i nomi dei giocatori, visualizzare le regole, i crediti o uscire dal gioco. 3. **Funzioni per gestire il menu**:    * inserisciNomiGiocatori(giocatore1, giocatore2): Permette ai giocatori di inserire i loro nomi.    * visualizzaRegole(): Mostra le regole del gioco.    * visualizzaCrediti(): Mostra i crediti del gioco. 4. **Inizializzazione del campo di gioco**:    * Dopo che i nomi dei giocatori sono stati inseriti, la funzione inizializza\_campo prepara la griglia di gioco vuota. 5. **Loop del gioco**:    * Un altro ciclo while gestisce il flusso del gioco.    * I giocatori alternano i loro turni inserendo le loro palline nella colonna desiderata.    * Dopo ogni mossa, viene controllato se il giocatore corrente ha vinto.    * Se un giocatore vince, il gioco termina e viene visualizzato un messaggio di vittoria. 6. **Utilizzo di colori ANSI**:    * Il codice utilizza sequenze di escape ANSI per aggiungere colori al testo stampato sulla console, rendendo più accattivante l'interfaccia utente. 7. **Chiusura del programma**:    * Alla fine del gioco o se il giocatore sceglie di uscire dal menu, il programma termina con return 0. |  | |

6 Le Funzioni



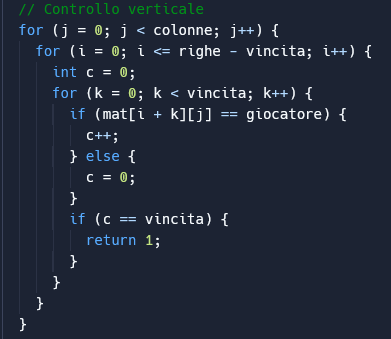
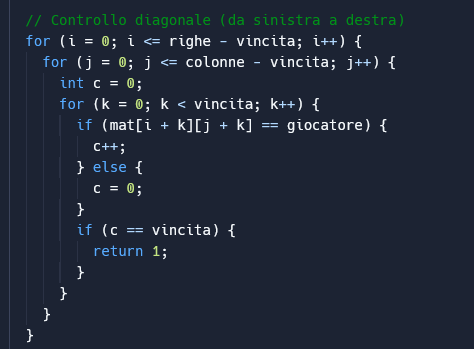
La funzione inizializza\_campo serve per preparare la griglia di gioco vuota all'inizio di una partita. Ecco una spiegazione dettagliata della funzione:

* **Argomenti**:
  + int riga: rappresenta il numero di righe della griglia di gioco.
  + int colonna: rappresenta il numero di colonne della griglia di gioco.
  + char mat[riga][colonna]: è la matrice bidimensionale che rappresenta il campo di gioco, dove verranno memorizzati i valori dei singoli punti della griglia.
* **Operazioni**:
  + La funzione utilizza un doppio ciclo for per scorrere attraverso ogni elemento della griglia di gioco.
  + Ogni elemento della matrice viene inizializzato con il valore VUOTO. Tuttavia, il valore VUOTO non è definito direttamente nella funzione. Solitamente, nelle implementazioni di giochi come Connect Four, il valore VUOTO è una costante definita altrove nel programma per rappresentare che una casella della griglia non contiene alcuna pallina di nessun giocatore.

La funzione stampaCampo è responsabile della stampa della griglia di gioco sulla console. Ecco una spiegazione dettagliata di come funziona:

* **Argomenti**:
  + int riga: rappresenta il numero di righe della griglia di gioco.
  + int colonna: rappresenta il numero di colonne della griglia di gioco.
  + char mat[riga][colonna]: è la matrice bidimensionale che rappresenta il campo di gioco, contenente le informazioni sulle palline posizionate dai giocatori.
* **Operazioni**:
  + La funzione inizia stampando una nuova linea vuota, per separare la griglia di gioco da eventuali altre informazioni presenti sulla console.
  + Utilizza due cicli for annidati per scorrere attraverso ogni elemento della griglia di gioco.
  + Per ogni elemento della griglia, stampa il carattere corrispondente. Se il carattere rappresenta una pallina rossa (ROSSO) o una pallina gialla (GIALLO), viene stampato con il colore corrispondente utilizzando le sequenze di escape ANSI per colorare il testo. Se la casella è vuota, il carattere viene stampato senza alcun colore.
  + Dopo aver stampato tutti gli elementi di una riga, viene stampato un carattere di linea verticale (|) per separare le colonne.
  + Una volta completata la stampa di una riga, viene stampata una linea orizzontale (---------------) per separare le righe della griglia.
  + Il ciclo continua fino a quando non vengono stampati tutti gli elementi della griglia.





Questa funzione controlloVincita verifica se un giocatore ha vinto il gioco Connect Four, esaminando la griglia di gioco per trovare quattro palline consecutive dello stesso colore in qualsiasi direzione: orizzontale, verticale o diagonale.

Ecco una spiegazione dettagliata della funzione:

* **Argomenti**:
  + char giocatore: il giocatore di cui si vuole verificare la vittoria.
  + int riga: il numero di righe della griglia di gioco.
  + int colonna: il numero di colonne della griglia di gioco.
  + char mat[riga][colonna]: la matrice bidimensionale che rappresenta il campo di gioco.
  + int vincita: il numero di palline consecutive necessarie per vincere il gioco (generalmente 4 in Forza 4).

 **Controllo orizzontale**:

* I due cicli for esterni scorrono attraverso ogni riga della griglia di gioco.
* Il secondo ciclo for scorre attraverso ogni colonna, ma si ferma quando mancano meno di vincita colonne alla fine della riga. Questo è importante perché non è possibile trovare una sequenza vincente se non ci sono abbastanza colonne rimanenti per formare una sequenza completa.
* All'interno del secondo ciclo, c'è un altro ciclo for che controlla le successive vincita colonne.
* In questo ciclo, viene controllato se ci sono vincita palline consecutive dello stesso giocatore. Se sì, la funzione restituisce immediatamente 1 indicando che il giocatore ha vinto.

 **Controllo verticale**:

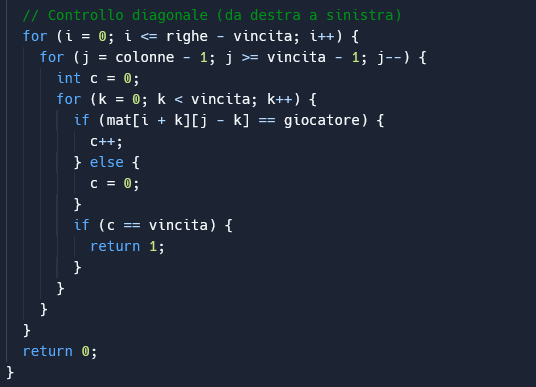
* Questo blocco di codice esegue un controllo simile al precedente, ma scorre attraverso le colonne anziché le righe per cercare sequenze vincenti verticali.

 **Controllo diagonale (da sinistra a destra)**:

* I due cicli for esterni esaminano ogni possibile inizio di una diagonale da sinistra a destra nella griglia.
* All'interno di questi cicli, c'è un ciclo for interno che controlla una sequenza di vincita celle in ciascuna diagonale.
* Questo ciclo verifica se ci sono vincita palline consecutive dello stesso giocatore in ogni diagonale. Se sì, restituisce immediatamente 1.

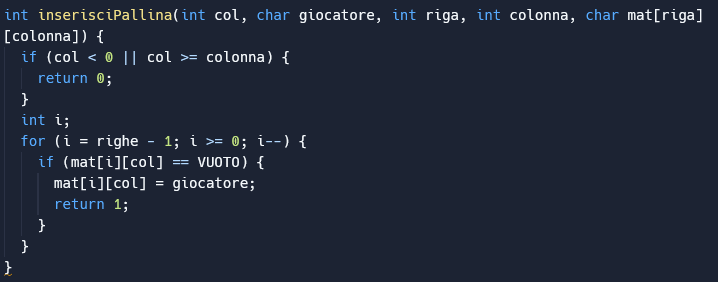
 **Controllo diagonale (da destra a sinistra)**:

* Questo blocco di codice esegue un controllo simile al precedente, ma esamina le diagonali della griglia da destra a sinistra
* **Ritorno**:
  + Se la funzione non trova nessuna sequenza vincente, restituisce 0 (false).



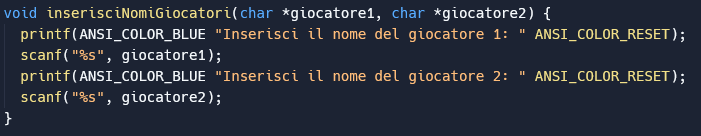
Questa funzione, inserisciPallina, gestisce l'inserimento di una pallina nella griglia di gioco. Vediamo come funziona:

* **Argomenti**:
  + int col: la colonna in cui il giocatore desidera inserire la pallina.
  + char giocatore: il giocatore che sta inserendo la pallina (ROSSO o GIALLO).
  + int riga: il numero di righe della griglia di gioco.
  + int colonna: il numero di colonne della griglia di gioco.
  + char mat[riga][colonna]: la matrice bidimensionale che rappresenta il campo di gioco.
* **Controllo dei limiti della colonna**:
  + La prima istruzione if verifica se la colonna specificata dal giocatore è valida. Se la colonna è inferiore a 0 o maggiore o uguale al numero di colonne della griglia, la funzione restituisce 0, indicando un inserimento non valido.
* **Inserimento della pallina**:
  + Il ciclo for scorre le righe della colonna, partendo dall'ultima riga e procedendo verso l'alto.
  + Per ogni riga, la funzione verifica se la cella è vuota (VUOTO). Se lo è, imposta il valore della cella con il giocatore corrente (giocatore) e restituisce 1, indicando che l'inserimento è avvenuto con successo.
  + Se non viene trovata nessuna cella vuota nella colonna, la funzione ritorna senza effettuare alcuna modifica alla griglia, indicando che la colonna è piena e l'inserimento non è possibile.



Questa funzione, inserisciNomiGiocatori, consente ai giocatori di inserire i loro nomi prima di iniziare una partita. Vediamo come funziona:

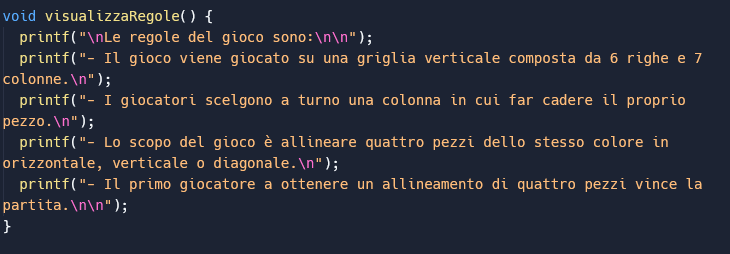
* **Argomenti**:
  + char \*giocatore1: puntatore a una stringa che memorizzerà il nome del giocatore 1.
  + char \*giocatore2: puntatore a una stringa che memorizzerà il nome del giocatore 2.
* **Input dei nomi dei giocatori**:
  + La funzione utilizza printf per stampare messaggi sulla console, chiedendo ai giocatori di inserire i loro nomi.
  + Viene utilizzata la sequenza di escape ANSI ANSI\_COLOR\_BLUE per colorare il testo in blu, rendendo più visivamente distinguibili le prompt dei nomi dei giocatori.
  + scanf viene utilizzato per acquisire le stringhe di input dei nomi dei giocatori dalla console. I nomi inseriti vengono memorizzati nelle variabili puntate da giocatore1 e giocatore2.



La funzione visualizzaRegole stampa le regole del gioco Connect Four sulla console. Ecco una spiegazione dettagliata:

* **Stampa delle regole**:
  + La funzione utilizza printf per stampare le regole del gioco in modo chiaro e leggibile sulla console.
  + Vengono stampate diverse frasi che spiegano le regole chiave del gioco:
    1. La griglia di gioco è una griglia verticale con 6 righe e 7 colonne.
    2. I giocatori prendono a turno per inserire un pezzo nella griglia, scegliendo una colonna dove farlo cadere.
    3. Lo scopo del gioco è allineare quattro pezzi dello stesso colore in orizzontale, verticale o diagonale sulla griglia.
    4. Il primo giocatore che riesce a ottenere un allineamento di quattro pezzi dello stesso colore vince la partita.

Queste regole forniscono una panoramica concisa e chiara del gioco Forza 4, aiutando i giocatori a comprendere le fondamenta del gioco prima di iniziare a giocare. La stampa delle regole avviene in un formato facilmente leggibile, consentendo ai giocatori di consultare rapidamente le informazioni necessarie prima di iniziare una partita.

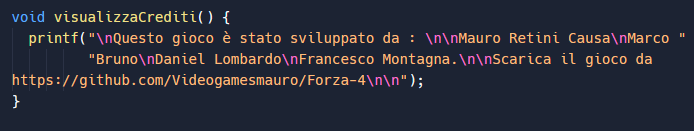


Questa è una funzione in linguaggio C chiamata visualizzaCrediti, che stampa a schermo i crediti di un gioco. Qui è presente una stringa di testo formattata che contiene i nomi dei membri del team di sviluppo del gioco e un link per scaricare il gioco da GitHub.

La funzione printf viene utilizzata per stampare il testo a schermo. Il testo è stato formattato con la sequenza di escape \n per inserire dei caratteri di nuova linea, rendendo la stampa più ordinata.

La funzione stampa i nomi dei membri del team di sviluppo: Mauro Retini Causa, Marco Bruno, Daniel Lombardo e Francesco Montagna. Inoltre, fornisce un link per scaricare il gioco da GitHub.

Infine, la funzione non ha alcun valore di ritorno (void), il che significa che non restituisce alcun dato quando viene chiamata. È utilizzata solo per scopi di output visivo.



* Mauro Retini Causa
* Marco Bruno
* Francesco Montagna
* Daniel Lombardo

3 E Informatica 2023/2024 ITIVE3PA

Francesco Montagna