

C++ #13 Realizzazione del gioco dell'oca in C++

Obiettivo: Realizzare un programma che implementi il gioco dell'oca utilizzando il linguaggio di programmazione C++.

Il codice

Viene in seguito illustrato il codice utilizzato per implementare una semplice versione del gioco dell'oca. Nel dettaglio, il codice prevede una netta distinzione tra il motore grafico che permette di stampare la griglia date le posizioni dei due giocatori ed il codice di gestione del gioco (i.e., lancio dei dadi aggiornamento della posizione, etc...).

- **Motore Grafico:** La funzione *void grafica(int pos1, int pos2)* ha il ruolo di stampare la griglia di gioco date le posizioni *pos1* e *pos2* dei due differenti giocatori. Nel dettaglio viene stampata una griglia con 17 caselle su ognuna delle quali viene riportato un numero da 00 a 16. L'impiego delle variabili relative alle posizioni degli utenti sono utili per stampare sulla specifica posizione del giocatore gli indicatori "XX" e "YY". Ogni volta che il giocatore cambia posizione attraverso questa funzione viene effettuato un refresh della pagina.



Esempio componente grafica del gioco dell'oca

- **Codice di gestione del gioco:** La dinamica del gioco viene sviluppata all'interno della funzione main. Nel dettaglio l'intero gioco viene implementato attraverso un ciclo while indefinito che prevede l'impiego di due condizioni $(pos1 < NUMPOS) \ \&\& \ (pos2 < NUMPOS)$. In questo caso i giocatori continueranno a lanciare dadi fino a che un giocatore non supera la posizione definita dalla `DEFINE NUMPOS`. Ogni volta che viene lanciato un dado viene aggiornata la GUI attraverso la funzione, precedentemente illustrata, `grafica(pos1, pos2)`.

Modifiche:

Al fine di rendere il gioco più avvincente è possibile introdurre una serie di modifiche:

- Caselle speciali: per avanzare, indietreggiare, o saltare dei turni
- Ponti: per passare da una casella ad un'altra.
- Gestione dei dadi: si può prevedere un secondo lancio quando si ottiene un sei.

C++ #12 Realizzare un programma per creare ASCII Art con le matrici

Obiettivo: Realizzare un programma per creare immagini utilizzando le matrici in stile ASCII Art. Attraverso il codice può essere inoltre aggiunto il senso del movimento.

Il codice (statico)

Viene in seguito illustrato il codice utilizzato per stampare su console una forma specifica creata con caratteri. L'esempio

specifico rappresenta una barchetta stilizzata.

Inizialmente, questo programma dichiara una matrice 11×11 e la inizializza con spazi vuoti. Successivamente, imposta alcuni caratteri specifici in posizioni specifiche della matrice per creare una forma particolare, e infine, stampa la matrice.

Ecco una descrizione dettagliata del funzionamento del codice:

1. **Inclusione delle librerie:** Questo programma include le librerie `<iostream>` e `<unistd.h>` per la gestione delle operazioni di input/output.
2. **Dichiarazione della funzione foo:** Questa funzione prende come argomento una matrice bidimensionale di caratteri di dimensione 11×11 e la stampa su console.
3. **Main:** Il main prevede una inizializzazione della matrice ed una serie di comandi che permettono di impostare alcuni caratteri specifici (come asterischi e trattini) in posizioni specifiche della matrice per creare una forma particolare. Infine la funzione `foo(matrix)` viene utilizzata per stampare il contenuto della matrice sulla console.

Il codice (dinamico):

In questa versione del codice, viene aggiunto il movimento dell'immagine grazie all'utilizzo delle funzioni `usleep` e `System("clear")` che rispettivamente bloccano il codice per un numero dato di microsecondi e cancellano il testo sulla console.

C++ #8 Esercizi con le Matrici

Obiettivo: Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo delle matrici.

Gli esercizi, a complessità crescente, possono essere risolti utilizzando anche differenti linguaggi di programmazione come C, Python, Pascal, Java, PHP.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

https://www.onlinegdb.com/online_c++_compiler

ESERCIZI CON MATRICI (DIFFICOLTA EASY)

1) Realizzare un programma che permetta di inserire in tutte le posizioni di una matrice di dimensione 5×5 un numero a scelta introdotto dall'utente. (N.B. In tutte le celle deve essere presente lo stesso numero)

2) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di dimensione 4×4 . (N.B. In tutte le celle deve essere presente un numero diverso scelto dall'utente)

2 bis) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di dimensione 4×4 e stamparne in seguito il contenuto.

3) Realizzare un programma che permetta di riempire, attraverso numeri casuali compresi tra 0 e 9, una matrice di dimensione 4×4 e stamparne in seguito il contenuto.

4) Realizzare un programma che permetta di riempire, attraverso numeri casuali compresi tra 0 e 9, una matrice di dimensione 4×4 . Determinare in seguito se un numero a scelta introdotto dall'utente è presente in una qualsiasi cella di una matrice 4×4 .

4 bis) Realizzare un programma che permetta di determinare quante volte un numero a scelta, introdotto dall'utente, è presente in una qualsiasi cella della matrice 4×4 .

Esempio

Inserire un numero: 3

Il numero 3 è presente 2 volte nella matrice

3	2	1	6
4	3	1	5
6	4	2	6
4	1	4	7

5) Realizzare un programma che permetta di riempire, con tutti zeri tranne nelle celle della diagonale principale (dove deve memorizzare uno), una matrice di dimensione 4×4 e stamparne in seguito il contenuto.

Esempio

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

ESERCIZI CON MATRICI (DIFFICOLTA MEDIUM)

6) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di dimensione 4×4 e ne calcoli la somma di tutti gli elementi presenti.

7) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di

dimensione 4×4 e dopo avere determinato la somma di tutti gli elementi presenti per ogni riga, determini la riga con la somma maggiore.

8) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di dimensione 4×4 . In seguito determini se tutti gli elementi presenti nella matrice risultino maggiori di n e minori di m (con n ed m forniti dall'utente).

9) Realizzare un programma che permetta di memorizzare in una matrice 10×10 la tavola pitagorica