Verifica di laboratorio: sviluppo di un gioco in C# con WPF

Obiettivo

Realizzare un'applicazione Windows basata su WPF che implementi un gioco chiamato *Trova il Tesoro*. Il programma dovrà utilizzare vettori e matrici per gestire la griglia del gioco e seguire i principi di programmazione ad oggetti per una corretta suddivisione delle responsabilità.

Consegne

1. **Funzionalità richieste:**
    - Creare una griglia 5x5 (matrice) che rappresenti il campo di gioco.
    - Posizionare casualmente un "tesoro" in una cella della griglia.
    - Consentire al giocatore di cliccare sulle celle della griglia per cercare il tesoro.
    - Mostrare feedback visivo (es. simboli o colori) per i tentativi falliti e riusciti.
    - Conteggiare e mostrare il numero di mosse effettuate dal giocatore (facoltativo).
2. **Struttura del progetto:**
    - Il programma deve essere suddiviso in classi distinte:
        - `GameLogic`: Gestisce la logica del posizionamento del tesoro e verifica i tentativi.
        - `GridManager`: Gestisce la matrice e le operazioni correlate.
        - `Player`: Tiene traccia del numero di mosse effettuate.
        - `Window`: Gestisce l'interfaccia utente come classe di tipo Window.
    - Al termine, commentare con i <summery> per spiegare chiaramente i blocchi di operazione.
3. **Interfaccia utente:**
    - Utilizzare WPF per creare una griglia dinamica di pulsanti che rappresentano le celle.
    - Mostrare messaggi grafici e/o message box per il feedback del giocatore (es. *"Tesoro trovato in X mosse!"*).
4. **Gestione degli errori:**
    - Prevedere la gestione di situazioni anomale (es. input non validi).
    - Soluzioni che non si eseguono, ovvero con errori di compilazione non potranno raggiungere la sufficienza.

Durata e ambiente

- **Tempo di svolgimento:** 100 minuti.
- **Ambiente di sviluppo:** Visual Studio con il linguaggio C#.

Criteri di valutazione

Criterio	Descrizione	Punteggio massimo
Correttezza funzionale	Il programma funziona correttamente: il tesoro viene generato casualmente e la ricerca produce risultati coerenti.	25 punti
Struttura del codice	Il codice è suddiviso in classi con responsabilità ben definite (es. `GameLogic`, `GridManager`, `Player`).	20 punti
Commenti nel codice	Ogni riga/parte del codice è accompagnata da commenti chiari che spiegano le funzionalità e le scelte progettuali.	15 punti
Implementazione della UI (WPF)	La griglia è visualizzata correttamente e risponde agli input dell'utente (es. clic sui pulsanti).	15 punti
Gestione degli errori	Il programma gestisce correttamente i possibili errori (es. input fuori dai limiti) o casi eccezionali.	10 punti
Ottimizzazione e leggibilità	Il codice è ben strutturato e facilmente leggibile grazie all'indentazione, ai nomi significativi delle variabili e ai metodi.	10 punti
Creatività ed esperienza utente	Eventuali elementi extra (es. simboli, colori, notifiche) che migliorano l'interattività e la presentazione visiva.	5 punti

Totale punteggio massimo: 100 punti

Indicazioni per lo sviluppo

1. Progettare le classi separatamente, seguendo i principi SRP (Single Responsibility Principle).
2. Implementare la matrice nel codice, garantendo che venga aggiornata correttamente.
3. Configurare l'interfaccia grafica (griglia di pulsanti) con eventi di click collegati alla logica del gioco.
4. Testare il programma per verificare il corretto funzionamento delle funzionalità richieste [No unit test].
5. Documentare ogni fase dello sviluppo con commenti summery esplicativi e organizzare il codice per facilitare la comprensione.

Obiettivi extra (facoltativi)

- Implementare una funzionalità per resettare il gioco senza chiudere l'applicazione.
- Conteggiare e mostrare il numero di mosse effettuate dal giocatore

Ecco come dovrebbe funzionare il gioco Trova il Tesoro:

1. **Avvio del gioco:**
    - L'applicazione apre una finestra che mostra una griglia di dimensione 5x5 (o altra dimensione specificata), composta da pulsanti che rappresentano le celle.
    - Il tesoro viene posizionato in modo casuale in una cella della griglia, senza che l'utente possa conoscere la posizione.
2. **Interazione del giocatore:**
    - Il giocatore clicca sui pulsanti della griglia per cercare il tesoro.
    - Ogni clic corrisponde a un tentativo e viene registrato come una "mossa".
3. **Feedback visivo:**
    - Quando il giocatore clicca su una cella:
        - Se **non** contiene il tesoro, il pulsante cambia (es. mostrando un simbolo "❌" o un colore diverso) per indicare il tentativo fallito.
        - Se la cella contiene il tesoro, il pulsante mostra un'icona di successo (es. "🎉" o un colore diverso) e appare un messaggio di congratulazioni.
4. **Messaggio finale:**
    - Quando il giocatore trova il tesoro, l'applicazione mostra una message box con un messaggio del tipo: *"Tesoro trovato!"*
    - Il gioco termina e il giocatore può scegliere di ricominciare (facoltativo).
5. **Conteggio delle mosse:**
    - Il numero totale di mosse viene registrato e mostrato alla fine per indicare la performance del giocatore (facoltativo).
6. **Gestione degli errori:**
    - Se il giocatore clicca più volte sulla stessa cella, il gioco non conta ulteriori mosse (facoltativo).
    - Eventuali input non validi vengono gestiti in modo da non causare crash dell'applicazione.