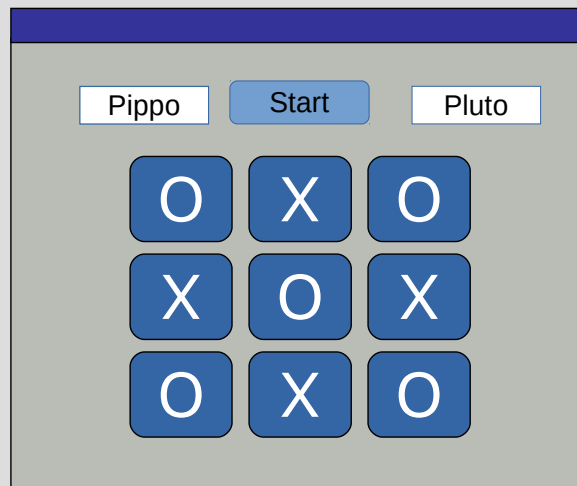


TRIS

Implementare il gioco del tris.

Layout della UI



Requisiti

L'interfaccia del programma deve implementare le seguenti funzionalità:

- Due *textbox* per l'inserimento dei nomi giocatori. (Si presuppone che al primo giocatore corrisponda il simbolo O mentre al secondo il simbolo X)
- Un bottone che determini l'inizio del gioco.
- Nove bottoni che identificano le "caselle" del gioco. Cliccando su un bottone, un giocatore segnerà la casella con il proprio simbolo, ma soltanto se questa non è già stata occupata. Inizialmente, i bottoni non presenteranno alcun simbolo.
- Il programma dovrà visualizzare mediante una *message box* l'esito del gioco, e cioè il nome del giocatore che ha vinto, oppure se c'è stato un pareggio.
- Per semplicità, si può supporre che il giocatore (O) muova sempre per primo.

Note sull'implementazione

Si può memorizzare lo stato in una matrice 3x3. (matrice `string`, `int` o `bool`, non ha particolare importanza). Un problema da risolvere è quello di associare i bottoni alle celle corrispondenti della matrice. Tale problema è connesso alla modalità di creazione dei nove bottoni. Si possono individuare due soluzioni; qui ne propongo una di base.

Soluzione "ingenua"

I nove bottoni creati a *design-time*; per ognuno di essi viene creato il corrispondente gestore di evento. Supponendo che i bottoni siano denominati: `btnUno`, `btnDue`, `btnTre`... e collocati da sinistra a destra e dall'alto in basso, si potrebbe scrivere il seguente codice:

```

...
string[,] tris = new string[3, 3];
string mossaCorrente = "O"; //cambia ogni volta ("O"->"X"->"O"...)

private void btnUno_Click(object sender, EventArgs e)
{
    tris[0, 0] = mossaCorrente;
    //... gestisce cambio mossa, verifica fine gioco, etc.
}

private void btnDue_Click(object sender, EventArgs e)
{
    tris[0, 1] = mossaCorrente;
    //... gestisce cambio mossa, verifica fine gioco, etc.
}

private void btnTre_Click(object sender, EventArgs e)
{
    tris[0, 2] = mossaCorrente;
    //... gestisce cambio mossa, verifica fine gioco, etc.
}
...

```

Nota bene: ogni gestore di evento utilizza coordinate diverse per memorizzare la mossa nella matrice.