

### TRIS – versione 2 (refactoring)

Migliorare l'implementazione del gioco, evitando di dover gestire separatamente gli eventi associati ai nove bottoni.

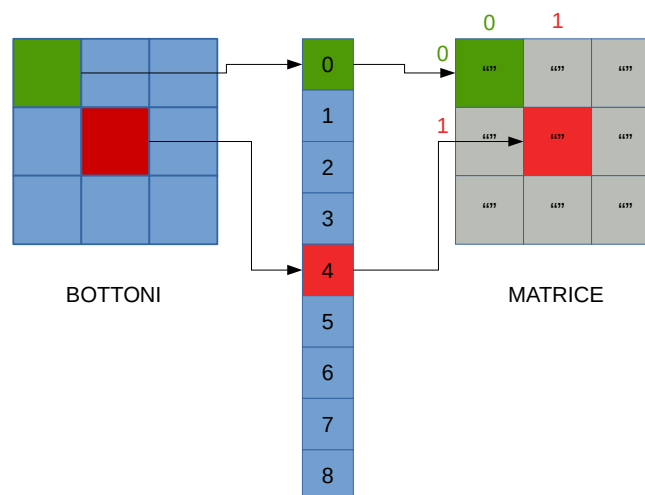
#### Note sull'implementazione

L'obiettivo è quello di far condividere ai nove bottoni lo stesso metodo che gestisce l'evento click. Per quanto riguarda l'associazione dell'evento click al metodo si può utilizzare il *property editor* a *design-time*.

Il problema è l'implementazione del codice che gestisce l'evento, poiché, in questo caso, il mittente può essere uno qualsiasi dei nove bottoni. L'obiettivo è:

1. Trovare il numero del bottone cliccato:  $0 \leftrightarrow 8$ . (Conviene sempre numerare da zero.)
2. Da questo, ottenere le coordinate di riga e colonna della matrice.

Ad esempio:



#### Individuare il bottone che ha sollevato l'evento

Premessa: occorre collocare i bottoni all'interno di un *panel*, e non direttamente nel *form*.

Il primo passo è quello di determinare la posizione del bottone all'interno del *panel*.

Nel gestore di evento, il parametro `sender` referencia il bottone cliccato. Occorre utilizzare questa variabile per cercare il bottone tra quelli contenuti nel *panel*, i quali sono memorizzati nella collezione `Controls`. (che è praticamente un `List<>` e dunque definisce il metodo `IndexOf()`):

```
private void Btn_Click(object sender, EventArgs e)
{
    Button b = (Button) sender; //-> "obbliga" C# a considerare sender di tipo Button
    int indice = pnlTris.Controls.IndexOf(b); //-> cerca posizione del bottone
    ...
}
```

Nota bene: non si può utilizzare direttamente `sender`, poiché non è del tipo giusto. Occorre "obbligare" il C# a considerare `sender` come se fosse di tipo `Button`; è a questo che serve la prima istruzione.

Altra nota: perché il processo di ricerca funzioni, è necessario che il *panel* contenga soltanto i nove bottoni. (Da qui l'importanza di collocare i bottoni nel *panel* invece che crearli nel *form*, il quale contiene anche altri controlli.)

Il secondo passo consiste nell'ottenere, a partire da `indice`, le coordinate di riga e di colonna della matrice. Si tratta di aritmetica di base e dunque si lascia la soluzione allo studente.