

TIC TAC TOE

CONSOLE APPLICATION

RICHIESTE.....3

LOGICA DI POSIZIONAMENTO.....3

 GIOCATORE VS CPU.....3

 INIZIA IL PC.....3

 INIZIA IL GIOCATORE.....6

 GIOCATORE 1 VS GIOCATORE 2.....7

RICHIESTE

- Possibilità di giocare in contemporanea con un altro giocatore (**P1** vs **P2**)
- Possibilità di giocare in singolo contro il computer (**P1** vs **CPU**)
- Avere la possibilità di scelta con quale forma iniziare (**X** oppure **O**)
- Inserimento nel caso del giocatore deve avvenire tramite un input da tastiera scrivendo il numero di cella che si vuole riempire

LOGICA DI POSIZIONAMENTO

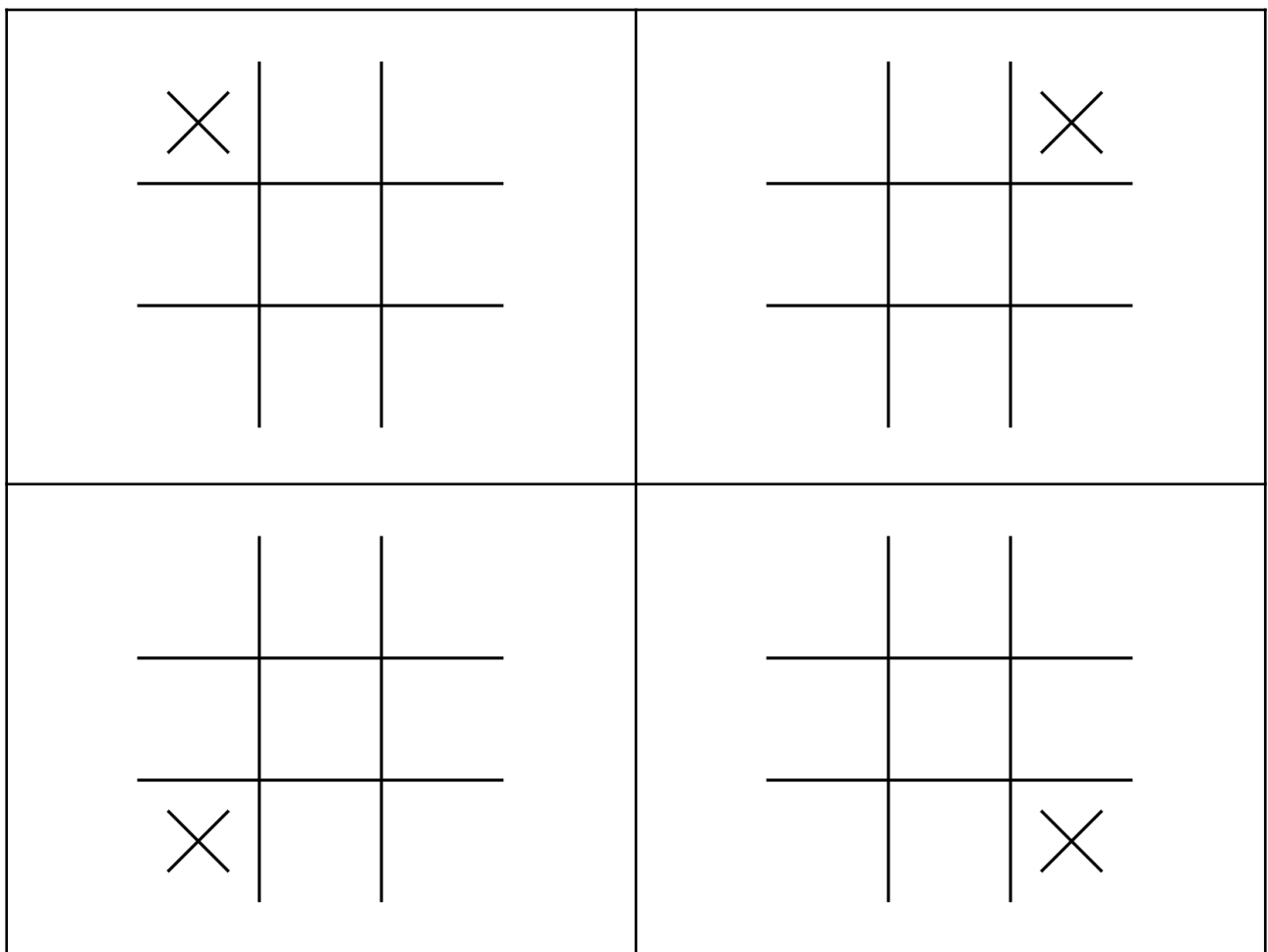
In questa sezione viene illustrato il funzionamento e la logica che viene attuata dal pc per capire dove posizionare la propria forma (che essa sia la **X** o il **O**)

GIOCATORE VS CPU

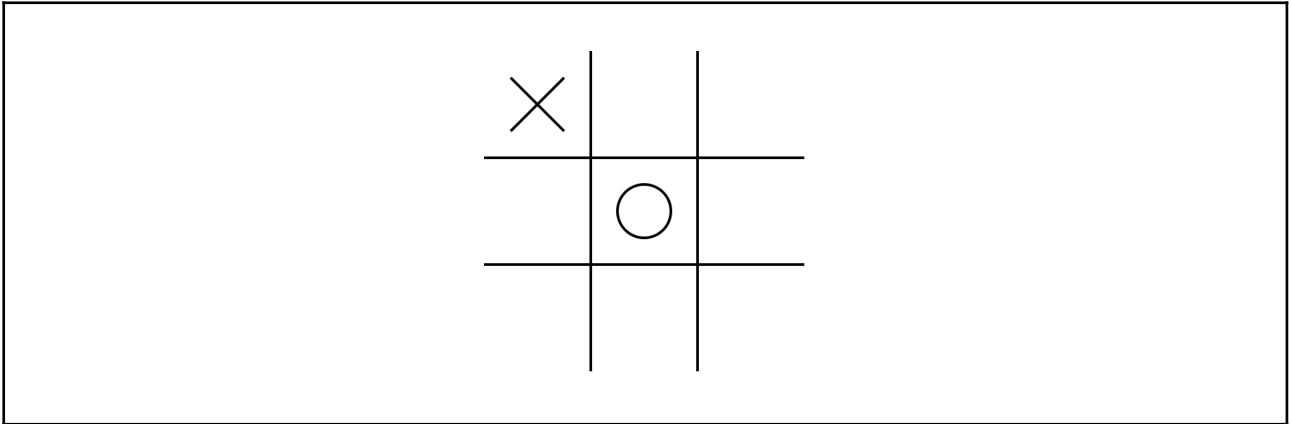
Ponendo il caso che il giocatore che il giocatore abbia scelto la **O** e il pc il **X**

INIZIA IL PC

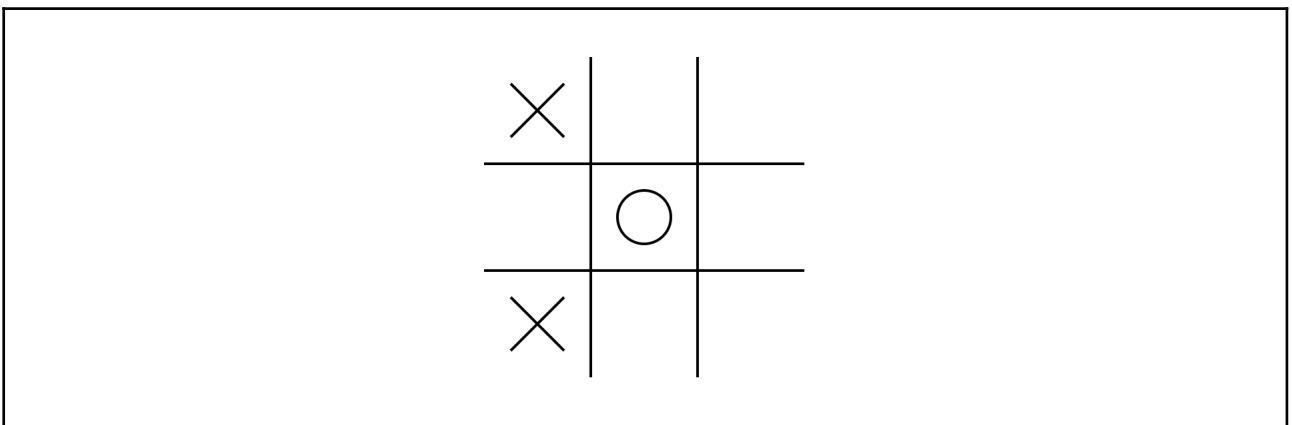
[**TURNO 1**] - Il PC lui cercherà di mettere la X negli spigoli.



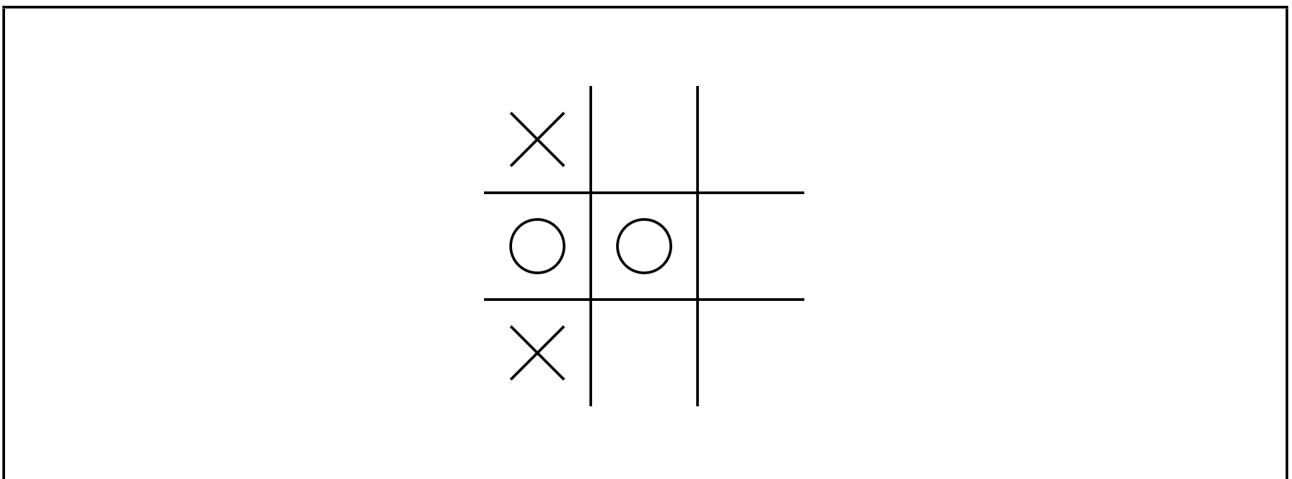
[TURNO 2] - Durante il primo turno del giocatore non dovrà essere eseguito nessun controllo perché non c'è nessun rischio di vittoria da parte del giocatore, in questo caso poniamo che il giocatore posizioni il **○** al centro.



[TURNO 3] - Controlla se negli spigoli (orizzontalmente e verticalmente alla prima X) c'è spazio libero. (Essendo che siamo al terzo turno obbligatoriamente uno dei due deve essere libero)

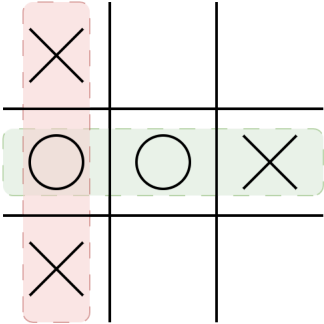
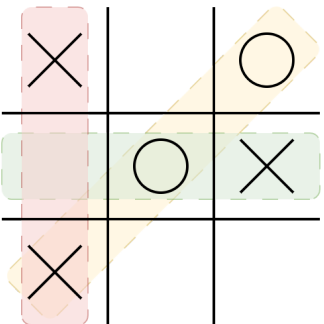
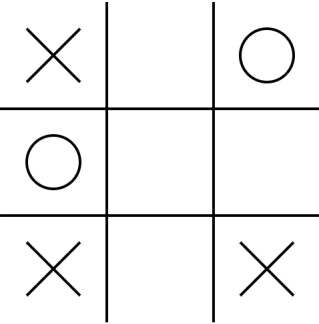


[TURNO 4] - Il giocatore a questo punto metterà il **○** dove preferisce

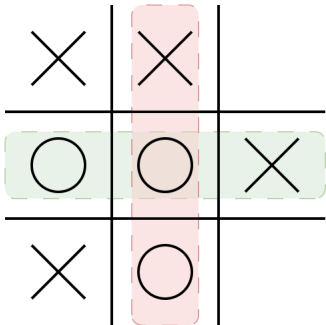
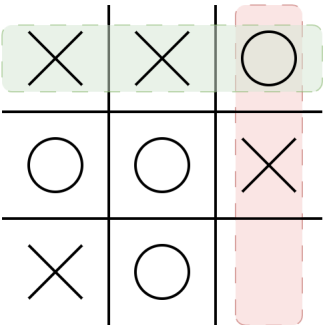
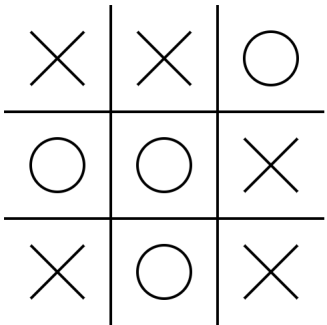


[TURNO 5] - A partire da questo punto la CPU andrà a verificare se (orizzontalmente, verticalmente e nel caso si trova al centro anche la diagonale CASO 1.1) ci sono due **O** consecutivi:

- Se ci sono andrà a mettere una **X** per bloccare la possibile vittoria
- Se non ce ne sono proverà a mettere la **X** in uno degli spigoli

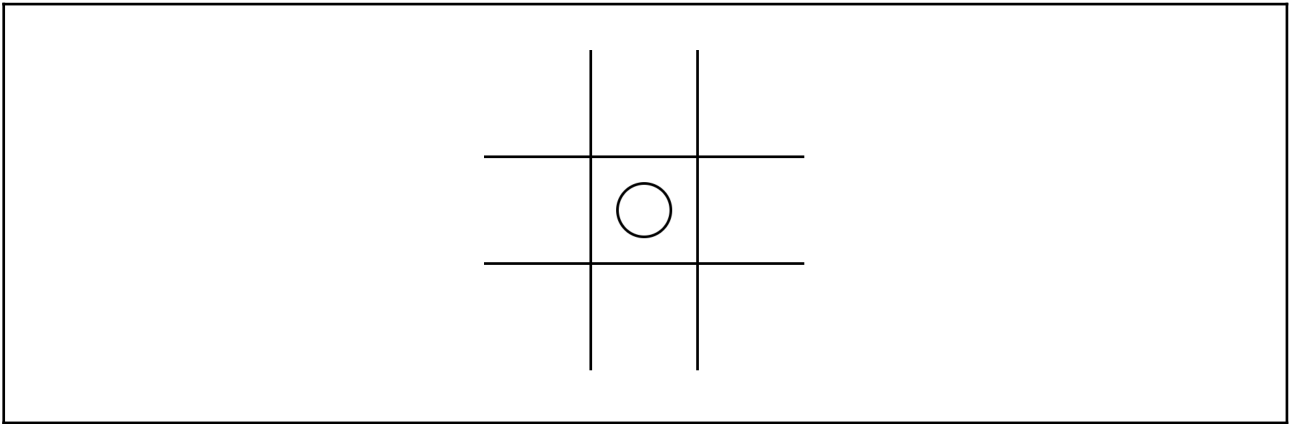
CASO 1	CASO 1.1
	
CASO 2	
	

Infine da qui in poi un ciclo controlla se sono presenti tre **X** o **O** consecutivi per verificare se è avvenuta una vittoria. Di seguito possibile continuo dal **CASO 1**

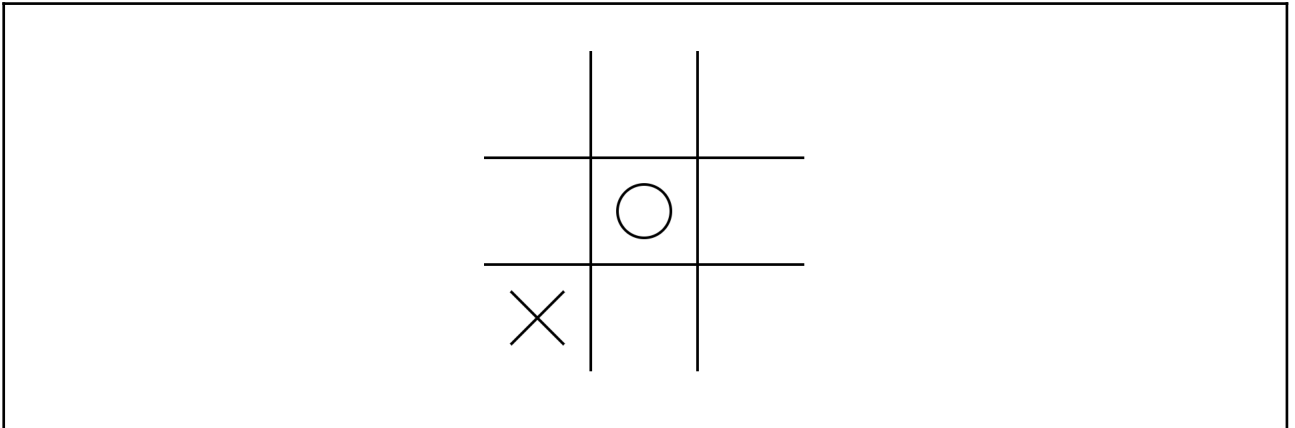
		
---	---	---

INIZIA IL GIOCATORE

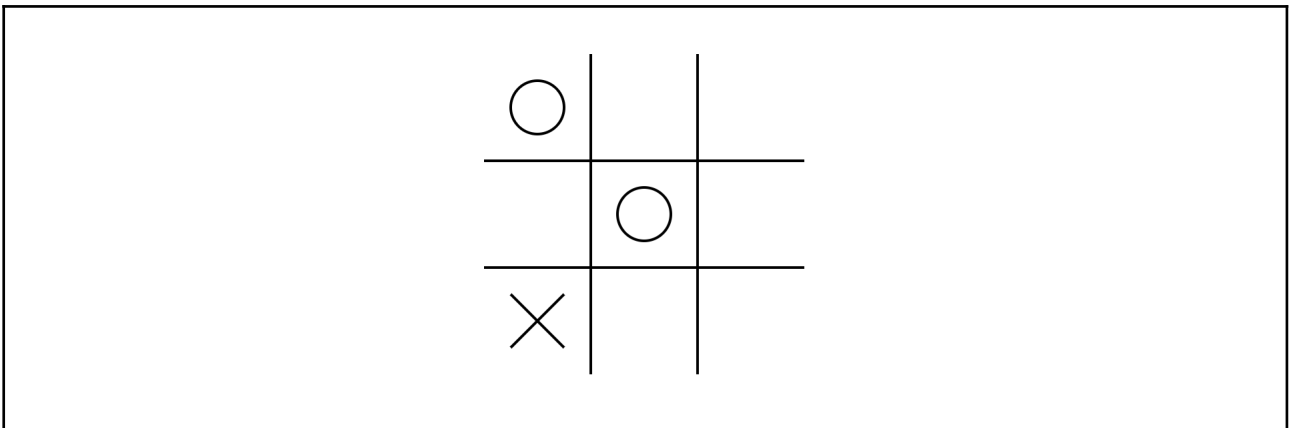
[TURNO 1] - Il giocatore sceglie dove mettere il **O**



[TURNO 2] - Il PC cercherà di posizionare la X negli spigoli

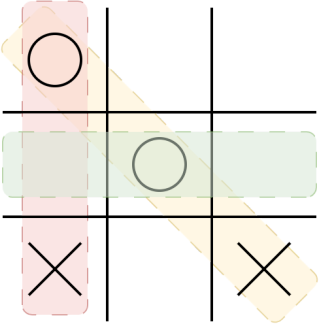
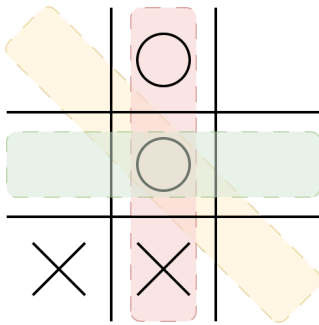


[TURNO 3] - Il giocatore sceglie dove mettere il **O**



[TURNO 4] - A partire da questo punto la CPU andrà a verificare se (orizzontalmente, verticalmente e nel caso si trova al centro anche la diagonale) ci sono due **O** consecutivi:

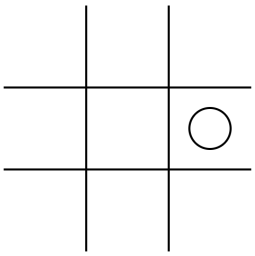
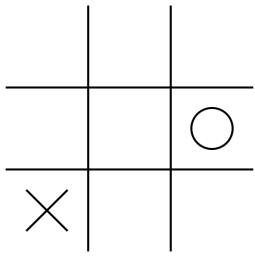
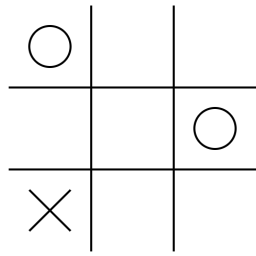
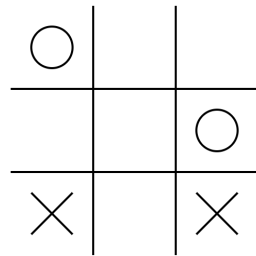
- Se ci sono andrà a mettere una **X** per bloccare la possibile vittoria
- Se non ce ne sono proverà a mettere la **X** in uno degli spigoli

CASO 1	CASO 2
	
<i>In questo caso per esempio abbiamo nella diagonale principale due O quindi si avvererà il primo caso</i>	<i>In questo caso per esempio abbiamo nella verticale due O quindi si avvererà il secondo caso</i>

Infine da qui in poi un ciclo controlla se sono presenti tre **X** o **O** consecutivi per verificare se è avvenuta una vittoria.

GIOCATORE 1 VS GIOCATORE 2

[TURNO 1-3] - I primi turni possono procedere senza controlli (tranne verificare se la casella è già piena oppure no)

			
---	---	--	---

[TURNO 4] - A partire da questo punto un ciclo controlla per ogni casella se sono presenti tre simboli uguali, in questo caso viene determinata la vittoria.

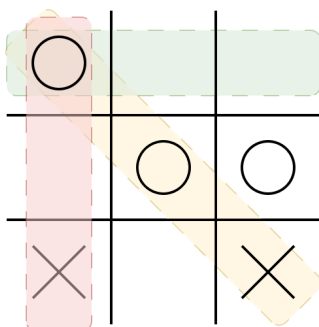


DIAGRAMMA DI FLUSSO

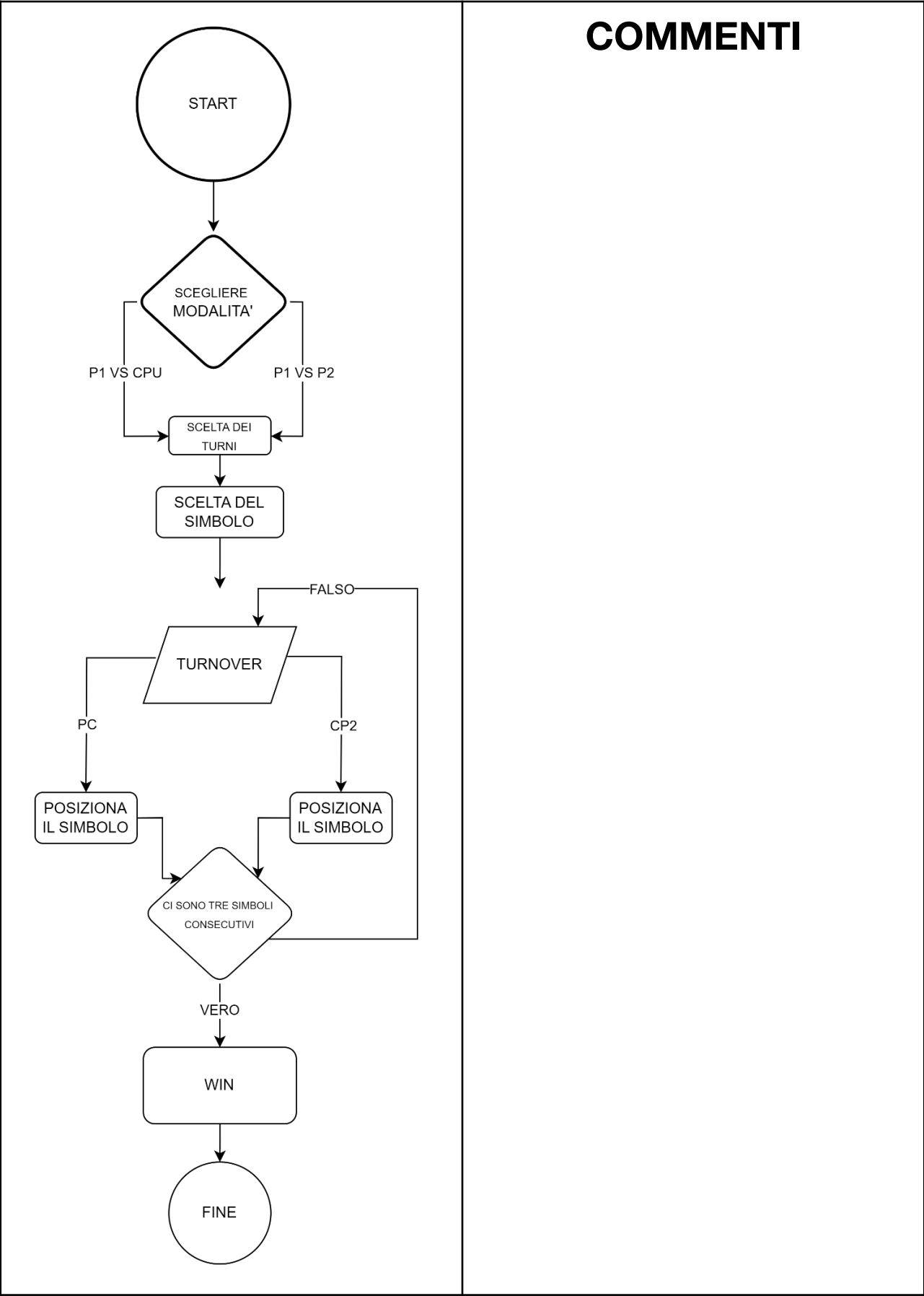
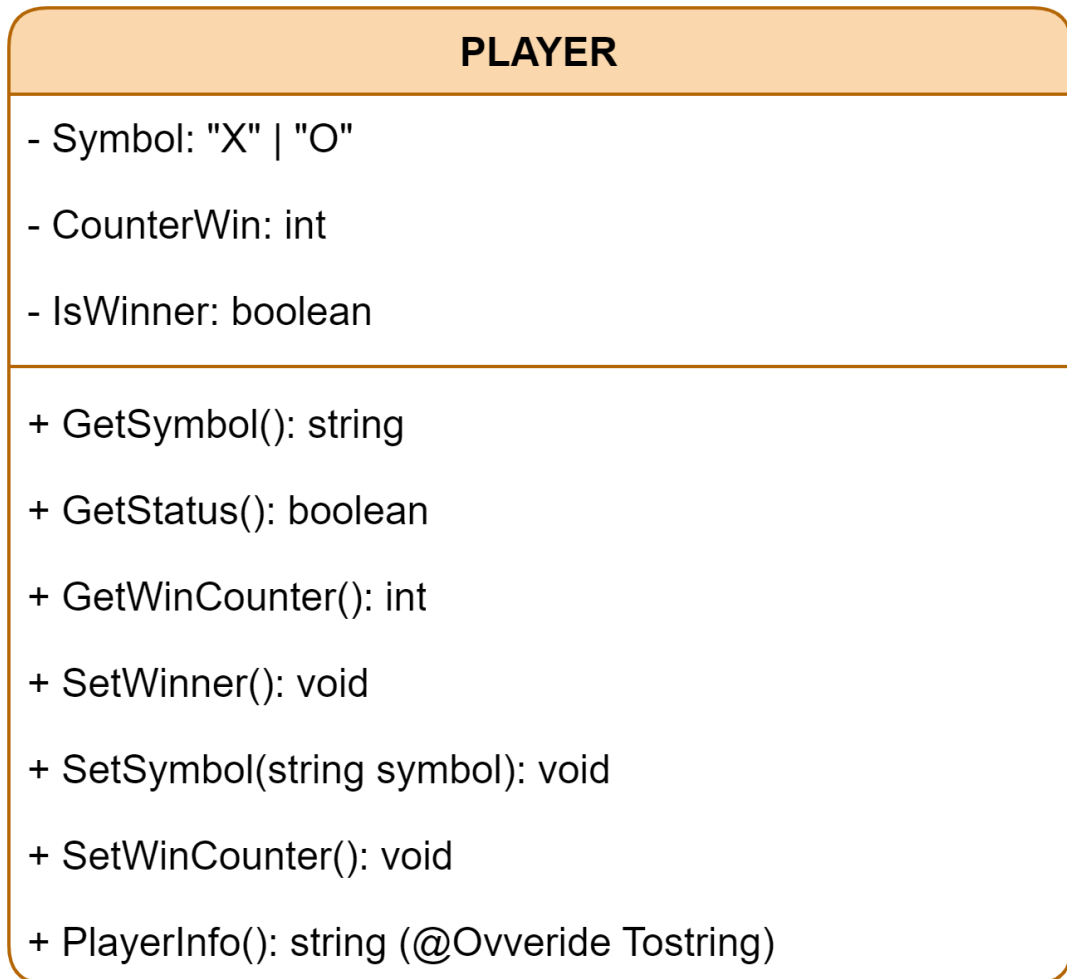


DIAGRAMMA UML

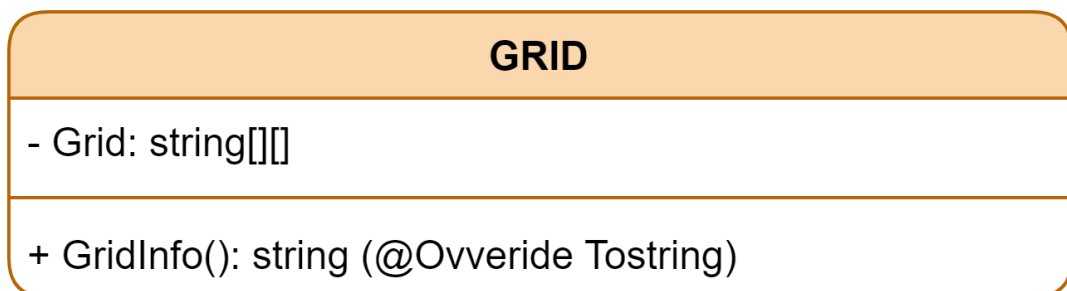
PLAYER

Oggetto che rappresenta il giocatore



GRID

Oggetto che si riferisce alla matrice dove sono contenuti i dati



GAME

Oggetto che rappresenta il vero e proprio gioco con la logica che controlla la vittoria

