# Sviluppo di un giocatore software per Connect X

Liam Busnelli - 0001019817
Fabio Chiarini - 0001028936
Alessandro Testa - 0001043390

Alma Mater Studiorum - Università di Bologna

A.A. 2022/2023

## Contesto

- Il Connect X è una versione generalizzata del gioco Connect 4, in cui la partita viene giocata su una matrice di dimensione $M \times N$. Questa variante del gioco offre una maggiore complessità strategica, consentendo ai giocatori di personalizzare la dimensione della griglia e il numero di gettoni da allineare per vincere. Ad esempio, nel Connect 4 tradizionale, il gioco si svolge su una griglia 6x7 e l'obiettivo è allineare 4 gettoni, mentre nel Connect X, possiamo avere dimensioni della griglia e un numero di gettoni allineati variabili.

# Introduzione



## Obiettivo

- L'obiettivo del gioco Connect X è allineare X gettoni verticalmente, diagonalmente o orizzontalmente. Si richiede che il giocatore sia in grado di prendere decisioni strategiche efficaci nel minor tempo possibile, considerando le molteplici possibilità di gioco offerte ad esempio bloccare le mosse dell'avversario, creare allineamenti e sfruttare spazi vuoti per future mosse vantaggiose.
- Nel progetto ci si propone di sviluppare un giocatore software in grado di giocare nel modo migliore possibile su tutte le istanze ragionevoli di Connect X, rispettando un limite rigido di tempo per ogni mossa. Ciò richiede l'implementazione di algoritmi e strategie intelligenti che consentano al giocatore di prendere decisioni rapide ed efficienti.

# Metodologia



Per affrontare il problema, abbiamo adottato diverse strategie e tecniche:

## Tecnica Minimax

- Abbiamo implementato l'algoritmo Minimax, una tecnica di ricerca decisionale, per valutare le mosse possibili. Ci consente di esplorare l'albero di gioco fino a una certa profondità, valutando i possibili risultati di ogni mossa. L'obiettivo è massimizzare il vantaggio del giocatore e minimizzare il vantaggio dell'avversario.

# Metodologia



## Tecnica Alpha-Beta Pruning

- Abbiamo utilizzato la potatura dell'albero di gioco, in particolare l'Alpha-Beta Pruning, per ridurre l'effort computazionale. Ci permette di eliminare i rami dell'albero di gioco che non influenzeranno la decisione finale, accelerando così il processo decisionale.

# Metodologia



## Tecnica Evaluation Function

- Abbiamo progettato una funzione di valutazione che tiene conto di diversi fattori. Assegna un punteggio a ciascuna configurazione di gioco, considerando il numero di gettoni allineati, la presenza di spazi vuoti per future mosse e la formazione di blocchi o minacce dell'avversario. La valutazione accurata delle configurazioni di gioco è fondamentale per prendere decisioni strategiche intelligenti.

# Metodologia



## Tecnica Variable Deepening

- Abbiamo gestito il limite rigido di tempo per ogni mossa attraverso una ricerca in tempo limitato. Utilizzando una ricerca iterativa con una profondità crescente, siamo in grado di esplorare un numero maggiore di mosse finché il limite di tempo non viene raggiunto. Questo ci consente di effettuare una mossa ottimale nel tempo disponibile.

# Implementazione



## Classe Combo

- La classe Combo rappresenta una sequenza di celle marcate sul tabellone di gioco. Ogni combo tiene traccia della sua lunghezza, del numero di celle libere adiacenti che potrebbero estendere la combo, della direzione della combo orizzontale, verticale o diagonale, e di altre informazioni utili.

# Implementazione



## refreshCombos

- Il metodo refreshCombos è uno dei metodi più critici. Questo metodo aggiorna l'elenco delle combo di un giocatore dopo ogni mossa, aggiungendo nuove combo create dalla mossa e aggiornando le combo esistenti che sono state estese dalla mossa.

## CalculateComboValue

- Il metodo calculateComboValue è usato per determinare il valore di una combo. Il valore è calcolato in base alla lunghezza della combo, al numero di celle libere che potrebbero estenderla ed altri fattori.

# Risultati



## Discussione

- Con l'attuale implementazione, il codice gestisce in modo efficiente le combo nel gioco di Connect X, aggiornandole dinamicamente insieme al loro valore in risposta alle mosse effettuate durante il gioco. Questo approccio strategico permette di guidare il giocatore verso la vittoria. Tuttavia, è possibile migliorare ulteriormente il codice aggiungendo funzionalità avanzate come l'utilizzo di tabelle hash per memorizzare le mosse già valutate, aumentando così l'efficienza dell'algoritmo. Inoltre, sarebbe possibile implementare un meccanismo di apprendimento automatico per migliorare le prestazioni del giocatore, consentendo al programma di imparare da partite precedenti e di adattarsi alle strategie degli avversari.