

Progetto ASD A.A. 2020-2021

Pietro Di Lena

Università di Bologna

(M,N,K)-game

- Il gioco a turni **(M,N,K)-game** è una versione generalizzata del classico *tris*
 - La partita viene giocata su una matrice di dimensione $M \times N$
 - Per poter vincere bisogna allineare (verticalmente, orizzontalmente o diagonalmente) K simboli consecutivi
 - Il tris è un $(3,3,3)$ -game
- Il goal del progetto è sviluppare un giocatore software in grado di giocare in modo *ottimale* a tutte le istanze possibili del (M,N,K) -game

Informazioni generali

- Può essere svolto in gruppo (massimo due persone)
- Si consegna una sola volta
 - se necessario, saranno richieste correzioni specifiche
- Il voto del progetto resta valido anche per gli anni accademici successivi
 - a meno che non cambino i docenti del corso
- E' possibile consegnarlo entro Febbraio 2022
 - oltre Febbraio 2022, attendere le nuove specifiche
- Il voto del progetto pesa per 1/3 sul voto finale
 - media pesata tra voto dello scritto + voto del progetto

In cosa consiste la prova di progetto

- **Sviluppo**

- Il linguaggio da adottare è Java
- Vi viene fornita l'interfaccia Java da implementare (oltre ad ulteriore software per poter effettuare test)

- **Relazione**

- Le scelte progettuali adottate devono essere commentate in una breve relazione
- (suggerimenti in fondo alle slide)

- **Discussione orale**

- Discussione orale sul lavoro svolto
- Non è un orale sugli argomenti del corso

Modalità di consegna e discussione

- **Codici sorgenti e relazione: tramite EOL**
 - E' sufficiente che lo faccia un solo componente del gruppo (se si lavora in gruppo)
 - Non è necessario formare subito i gruppi, i componenti dovranno essere specificati al momento della consegna
- **Discussione del progetto: appelli su AlmaEsami+TEAMS**
 - Devono partecipare tutti i componenti del gruppo
 - E' preferibile (*decideremo se farlo diventare obbligatorio*) che la discussione avvenga a ridosso della consegna
- Avremo diversi slot temporali per la consegna e la discussione (a partire da fine Maggio)

Interfaccia MNKPlayer

```
public interface MNKPlayer {  
    // Inizializza il giocatore software  
    // M = numero di righe nella matrice  
    // N = numero di colonne nella matrice  
    // K = numero di simboli da allineare  
    // first = true se è il primo a giocare  
    public void initPlayer(int M, int N, int K, boolean first);  
  
    // Seleziona e ritorna una mossa tra quelle in FC  
    // FC = array di celle libere (giocabili) nella matrice  
    // MC = array di celle già occupate (history del gioco)  
    public MNKCell selectCell(MNKCell[] FC, MNKCell[] MC);  
  
    // Ritorna il nome del giocatore  
    public String playerName();  
}
```

Attenzione: il costruttore dell'implementazione non deve prendere argomenti

Classe MNKCell

```
public class MNKCell {  
    // indice riga  
    public final int i;  
    // indice colonna  
    public final int j;  
    // stato della cella i,j: FREE, P1 (Player1), P2 (Player2)  
    public final MNKCellState state;  
  
    public MNKCell(int i, int j, MNKCellState state) {  
        this.i      = i;  
        this.j      = j;  
        this.state = state;  
    }  
  
    // Altro codice poco rilevante ...  
    ...  
}
```

Package `mnkgame`

- Pacchetto Java contente l'interfaccia grafica per giocare e testare la propria implementazione
- Nel pacchetto sono anche disponibili due implementazioni (banali) di `MNKPlayer` a scopo illustrativo
- Da riga di comando il gioco può essere avviato nel seguente modo

```
java MNKGame <M> <N> <K> [MNKPlayer class]
```

- L'ultimo argomento è opzionale. Se omissso, il gioco parte in modalità Human vs Human

Qual è il punto "caldo" del progetto

- Il metodo `selectCell(...)` deve effettuare una scelta *intelligente* della mossa tra tutte quelle possibili
- Lo spazio di ricerca delle mosse *intelligenti* cresce esponenzialmente rispetto alla dimensione della matrice di gioco ed il numero di simboli da allineare (in particolare ad inizio partita)
- L'implementazione deve cercare di trovare sempre una soluzione accettabile/ottima/quasi-ottima in poco tempo (non vogliamo aspettare secoli per una mossa, anche se ottima)
- L'interfaccia di gioco (`MNKGame`) fa uso di un timeout di 10 secondi di attesa per una mossa e poi termina il programma
- Per quanto 10 secondi non siano un vincolo rigido (non sarò una macchina nella valutazione), servono a ricordare che bisogna puntare su soluzioni efficienti in termini di risorse di calcolo

(M,N,K)-game: proprietà note

- Tramite una *dimostrazione per furto di strategia* si può dimostrare che il secondo giocatore non può avere una strategia che gli assicuri la vittoria
 - Il secondo giocatore non può vincere se il primo ha una strategia ottima, al massimo può pareggiare (*draw*)
- E' un gioco *risolto* per alcune configurazioni. Ad esempio:
 - $(3,3,3)$ e $(4,4,4) \Rightarrow \text{draw}$
 - $(4,5,4)$ e $(5,4,4) \Rightarrow \text{draw}$
 - $(4,3,3)$ e $(3,4,3)$ e $(6,5,4) \Rightarrow \text{win}$
 - $(m,4,4)$ $m \leq 8 \Rightarrow \text{draw}$, $m \geq 30 \Rightarrow \text{win}$, $8 < m < 30 \Rightarrow ??$
 - ...

Organizzazione della relazione

- **Intestazione**

- Indicare il nome del progetto, nomi, cognomi e numero di matricola dei componenti del gruppo

- **Problema**

- Descrivere il problema computazionale affrontato
- Non è necessario fare un tema. Introdurre il problema ed indicare i punti salienti che hanno portato alle scelte progettuali adottate

- **Scelte progettuali**

- Descrivere ad alto livello l'implementazione del metodo `selectCell(...)` (eventualmente con aiuto di pseudocodice)
- Citare esplicitamente le strutture dati e gli algoritmi noti utilizzati (se applicabile) e sottolineare le strategie *originali* nell'implementazione
- Fornire un'analisi della complessità computazionale

Oltre al voto ...

- I progetti consegnati e funzionanti saranno utilizzati per un torneo tutti-contro-tutti per determinare una classifica delle implementazioni più forti di giocatori (M,N,K)
- La classifica verrà aggiornata di volta in volta
- E' solo per nostro divertimento interno, non avrà alcuna influenza sul voto
- Il nome che comparirà nella classifica è quello che assegnerete al vostro giocatore tramite il metodo `playerName()`
- Vi aggiornerò prima delle consegne sulle regole delle sfide che determineranno la classifica