

FONDAMENTI DI INTELLIGENZA ARTIFICIALE-M

A.A. 2017/2018

Gioco del "Mulino"

Nine Men's Morris Competition

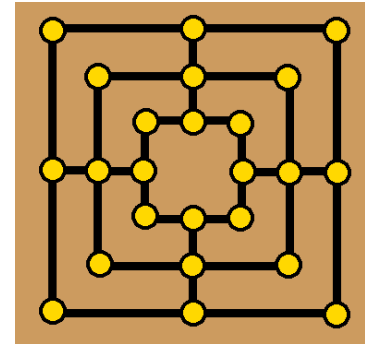
Prof. Paola Mello, Ing. Federico Chesani, Dott. Andrea Galassi

Il gioco del “Mulino” – scopo della competizione

- Stimolare la comprensione e la discussione sugli algoritmi di base per la risoluzione di giochi, nell’ambito della disciplina di AI (lato docente...)
- Vincere! (lato studente...)
- **Soluzioni accettabili:** qualunque soluzione che sfrutti algoritmi che si possono riferire all’area dell’Intelligenza Artificiale
 - Risoluzione di giochi come esplorazione nello spazio degli stati
 - Genetic algorithms and swarm optimization
 - Reti neurali
 - Approccio a vincoli
 - Prolog-based solutions
 - ...

Il gioco del “Mulino” – regole del gioco

- Due giocatori (Bianco e Nero) hanno ognuno 9 pedine a disposizione
- La tavola di gioco è composta da 3 quadrati concentrici, connessi tra loro nei punti medi dei lati. Le pedine possono essere posizionate negli angoli e nei punti medi dei quadrati
- Allineando 3 pedine lungo un segmento il giocatore completa un mulino e ha diritto a rimuovere dal gioco una pedina avversaria
 - Pedine allineate in un mulino possono essere rimosse solo se non vi sono pedine non allineate
- Scopo del gioco: eliminare 7 pedine avversarie o impedire all'avversario di potersi muovere



Il gioco del “Mulino” – regole del gioco

- Durante la prima fase di gioco, a turno, i giocatori posizionano una pedina sulla scacchiera
- Terminate le pedine, comincia la fase 2 e i giocatori ad ogni turno muovono una pedina lungo i segmenti in una posizione adiacente
- Quando un giocatore rimane con sole 3 pedine comincia la fase finale, in cui ad ogni turno può muovere liberamente una pedina in qualsiasi posizione
- Per altre informazioni:
 - consultare Wikipedia: http://it.wikipedia.org/wiki/Mulino_%28gioco%29
 - contattare Chesani/Galassi

Il gioco del “Mulino” – regole della competizione

- Girone all'italiana con andata/ritorno
- Ogni vittoria in uno scontro: 3 punti
- Ogni giocatore ha "tempo limitato":
 - 1 minuto per ogni mossa per ogni giocatore
 - Timeout deciso dall'arbitro (Chesani/Galassi o un server); in caso di timeout, il giocatore col turno perde la partita
- Stati ripetuti: in caso di ripetizione di uno stato, la partita è patta, 1 punto a giocatore.

Il gioco del “Mulino” – cosa realizzare

- Gli studenti devono realizzare un agente software in grado di giocare una partita comunicando con un piccolo server di sincronizzazione/scambio messaggi (fornito)
- Comunicazione tra i due giocatori tramite un servizio fornito da Chesani/Galassi (verifica delle regole, logging della partita)
- Rappresentazione dello stato fornita da Chesani/Galassi (non obbligatoria, ma comoda...)
- Per evitare differenze originate da hardware: l’agente software dovrà poter essere eseguito su una architettura linux debian 64 bit.

Il gioco del “Mulino” – strumenti forniti (1/2)

Esiste un progetto java Eclipse disponibile su svn all'url:

https://svn.ing.unibo.it/svn/ai/didattica_ai/mulino/Mulino

Tale progetto contiene:

- Un server che mantiene lo stato centrale del gioco (Stato in *State.java*, motore di esecuzione in *Engine.java*)
- Una classe astratta (*MulinoClient.java*) che offre due primitive: una per inviare al server la propria mossa, una per leggere dal server lo stato corrente.
- Un client che implementa un'interfaccia testuale per permettere a giocatori umani di giocare (*MulinoHumanClient.java*) che è possibile usare come riferimento.
- Due classi che eseguono la precedente nei diversi ruoli: *MulinoHumanWhiteClient.java* e *MulinoHumanBlackClient.java*

Il gioco del “Mulino” – strumenti forniti (2/2)

- Per poter fare una partita di prova, è necessario eseguire *Engine.java*, e poi a parte eseguire, in ordine, *MulinoHumanWhiteClient.java* e *MulinoHumanBlackClient.java*. In totale, tre processi distinti (server e due giocatori).
- Stampando a video il risultato di *State.toString()* si ottiene una rappresentazione ascii-art della scacchiera e dello stato attuale...
- La funzione *State.toCompactString()* produce invece una rappresentazione più compatta.

Il gioco del “Mulino” – comunicazione tra i processi

All'avvio il server invia a entrambi i giocatori lo stato attuale. Da quel momento in poi si comincia a giocare, per primo sempre il bianco (State.Checker.WHITE). Le mosse cambiano a seconda della fase e sono rappresentate come stringhe.

1. Fase 1 (State.Phase.FIRST): ogni mossa è una stringa di due lettere che indicano dove si vuole mettere la propria pedina. Ad esempio "a1". Se la propria mossa dà adito al diritto di rimuovere la pedina avversaria, bisogna anche indicare la posizione da cui rimuovere la pedina avversaria. Ad es: "a1g7" significa che il giocatore mette la propria pedina in a1, e che poiché tale mossa crea una tripletta, si deve rimuovere la pedina avversaria in posizione g7
2. Fase 2: le stringhe sono composte da 4 lettere che indicano la posizione di partenza della pedina e quella in cui muoverla. Ad esempio "a1a4" significa che la pedina che si trova in a1 va mossa in a4. Di nuovo, se la mossa permette di rimuovere una pedina avversaria, bisogna indicarne la posizione. Es: "a1a4g7"
3. Fase finale: rappresentazione analoga alla Fase 2

Il gioco del “Mulino” – comunicazione tra i processi

- Una volta inviata la propria mossa, il server risponde subito al giocatore comunicandogli il nuovo stato del gioco: a tal scopo, dopo la scrittura al server della mossa il giocatore deve subito leggere dal server il nuovo stato attuale, modificato dalla propria mossa.
- Dopo aver fatto ciò il turno passa all'avversario; il giocatore può effettuare una read bloccante per aspettare dal server di avere notizie: in particolare, non appena l'avversario effettua la sua mossa, il server comunica a entrambi i giocatori il nuovo stato

Riassumendo, supponendo di essere il giocatore di turno, il ciclo di gioco è simile a:

1. invio la mia mossa;
2. leggo dal server il nuovo stato, modificato a causa della mia mossa;
3. ri-eseguo lettura dal server, e mi blocco aspettando la mossa del mio avversario: appena questa viene fatta, ricevo dal server lo stato aggiornato, e posso riprendere (o continuare) il mio ragionamento;

Il gioco del “Mulino” – consigli sul linguaggio

- Che linguaggio utilizzare?
- JAVA: facilissimo, dato che tutto il progetto è già in Java
- Altro: leggermente più complicato, poiché è necessario realizzare un client di interfaccia tra il server e il giocatore:
 1. L'interfaccia riceve l'oggetto `State` dal server, lo traduce in stringa (se volete usate `State.toCompactString`) e lo manda al giocatore
 2. Il giocatore legge la stringa, elabora la mossa e la manda all'interfaccia
 3. L'interfaccia legge la mossa e la manda al server
 - È esattamente come funzionano le classi `MulinoHumanClient`! Sono interfacce tra il server e un giocatore umano!

Il gioco del “Mulino” – requisiti e vincoli

- Per ogni mossa ogni giocatore ha al massimo 60 secondi di tempo: se sfora il tempo, il giocatore perde. Qualora le sfide risultassero troppo lunghe, il tempo potrà essere abbassato a discrezione di Chesani/Galassi.
- Per motivi di efficienza, ogni software dovrà essere in grado di "giocare" con un limite di 2Gb di RAM
- Se il giocatore propone una mossa non consentita, vengono lanciate specifiche eccezioni (a seconda del tipo di errore), e il giocatore perde
- L'agente deve poter essere avviato specificando come primo parametro "White" o "Black" a seconda del ruolo del giocatore.
- Tutte le regole del gioco potranno essere comunque modificate da Chesani/Mello/Galassi per opportuni motivi, in qualsiasi momento, e senza preavviso...

Il gioco del “Mulino” – spunti di discussione

- Cercate di realizzare il vostro progetto seguendo i principi dell’ingegneria del software: riutilizzabilità, modularità ecc. ecc.
- Se doveste adattare il vostro giocatore per giocare a delle varianti del gioco, cosa dovrete cambiare e cosa invece potreste riutilizzare?
- Esempi di varianti
 - Lasker Morris: 10 pedine e le fasi 1 e 2 sono fuse assieme
https://en.wikipedia.org/wiki/Nine_Men%27s_Morris#Lasker_Morris
 - Morabaraba: 12 pedine e diversa scacchiera
https://en.wikipedia.org/wiki/Nine_Men%27s_Morris#Twelve_Men's_Morris

Il gioco del “Mulino” – presentazione e questionari

- Una volta consegnato il vostro giocatore vi verrà chiesto di compilare un questionario online anonimo relativo al vostro giocatore e alle tecniche che avete utilizzato
- A fine competizione dovrete presentare il vostro giocatore davanti ai professori e agli altri studenti, preparando una breve presentazione powerpoint
- Il giorno della presentazione vi verrà chiesto di compilare un questionario anonimo relativo all'esperienza della competizione, con scopo di ricerca e per migliorare l'esperienza negli anni prossimi

Il gioco del “Mulino” – parte organizzativa

- Le sfide si terranno verso la fine di maggio 2018.
Date ipotetiche:
 - invio giocatori entro 20 maggio 2018 (ore 23:59)
 - presentazione dei progetti, discussione, e vincitore (obbligatorio che si presenti tutta la squadra): prima settimana di giugno 2018
- Iscrizioni con una mail a Chesani && Mello && Galassi entro il 6 Maggio 2018
 - paola.mello *at* unibo.it
 - federico.chesani *at* unibo.it
 - a.galassi *at* unibo.it
- In ogni caso, Chesani e Galassi hanno sempre ragione.

Il gioco del “Mulino” – consegna

- Cosa:
 - Il vostro giocatore, ovvero la classe Client sotto forma di file **NomeGiocatore*.jar*
 - La cartella dell'intero progetto
- Come:
 - usando link a repository (Github, Dropbox, Drive ecc. ecc.) per evitare problemi del server di posta con archivi o eseguibili
- Nel caso il vostro giocatore sia composto da più eseguibili, specificate come lanciaarli da linea di comando. Specificatelo nella mail e mettete un file *Readme.txt* nella cartella del progetto

Il gioco del “Mulino” – dubbi...

- Tutte le regole qui elencate potranno essere modificate a discrezione di Chesani/Mello/Galassi, con notifica sul sito web del corso.
- Tutti gli studenti sono invitati a suggerire/proporre modifiche...
- In ogni caso, Chesani e Galassi hanno sempre sempre sempre ragione.