

FONDAMENTI DI INTELLIGENZA ARTIFICIALE-M

A.A. 2015/2016

Gioco del "Mulino"

Prof. Paola Mello, Ing. Federico Chesani

Il gioco del “Mulino”

- 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...)
- Definizione e regole:
http://it.wikipedia.org/wiki/Mulino_%28gioco%29
- **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” – parte ludica

- Girone all'italiana con andata/ritorno
- Ogni vittoria in uno scontro: 2 punti
- Ogni giocatore mossa ha "tempo limitato":
 - 1 minuto per ogni mossa per ogni giocatore
 - Timeout deciso dall'arbitro (Chesani o un server); in caso di timeout, il giocatore col turno perde la partita
- Stati ripetuti: in caso di ripetizione consecutiva per tre volte dello stesso stato, perdono entrambi (nessuno fa punti).

Il gioco del “Mulino” – parte software

- Rappresentazione dello stato fornita da Chesani (non obbligatoria, ma comoda...)
- Comunicazione tra i due giocatori tramite un servizio fornito da Chesani (verifica delle regole, logging della partita)
- L'agente dovrà supportare due interfacce:
 - Human-oriented: per quanto semplice, dovrà supportare un essere umano nel giocare
 - Server-oriented: Chesani fornirà piccolo server di sincronizzazione/scambio di messaggi
- Per evitare differenze originate da hardware: l'agente software dovrà poter essere eseguito su una architettura linux debian 64 bit. Linguaggio suggerito: Java.

Il gioco del “Mulino” – parte software

Esiste un progetto 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)
- Un client (classe MulinoClient.java) che offre due primitive: una per inviare al server la propria mossa, una per leggere dal server lo stato corrente.

Il gioco del “Mulino” – parte software

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.

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: "g7a1" 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 a1
2. Fase 2: analoga alla prima, ma con regole leggermente diverse
3. Fase finale: analoga alla seconda ma senza le restrizioni varie imposte nelle prime due fasi

Il gioco del “Mulino” – parte software

- 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” – parte software

- La classe `MulinoClient` può essere eseguita con parametro "White" o "Black" e offre la possibilità di giocare tramite console. Per poter fare una partita di prova, è necessario eseguire `Engine.java`, e poi a parte due istanze differenti di `MulinoClient.java`, una eseguita con parametro di avvio "White" e l'altra con parametro di avvio "Black". 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...

Il gioco del “Mulino” – parte software

- 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.
- 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

- Tutte le regole del gioco potranno essere comunque modificate da Chesani/Mello per opportuni motivi, in qualsiasi momento, e senza preavviso...

Il gioco del “Mulino” – parte organizzativa

- Le sfide si terranno verso la fine di maggio 2016 (data da decidersi)
- Iscrizioni con una mail a Chesani/Mello
- In ogni caso, Chesani ha sempre ragione.

Il gioco del “Mulino” – dubbi...

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