

IL FUORIGIOCO A QUATTRO
IN UNA SQUADRA DI CALCIO

USER GUIDE

Progetto di Sistemi Intelligenti
Distribuiti

Docente: Andrea Omicini

A CURA DI STEFANO SEVERI

Indice

Indice	3
1 Getting Started	5
2 Output su linea di comando	7
3 Output grafico	11



Capitolo 1

Getting Started

Come prima cosa è necessario procurarsi il file zippato di nome `sw.zip`;

quindi è necessario decomprimerne il contenuto in una cartella generica. I file presenti all'interno dovrebbero essere i seguenti

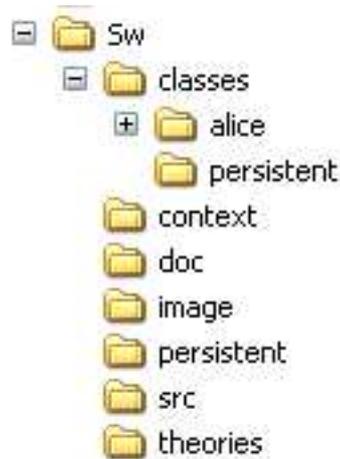


Figura 1.1: Cartelle pacchetto `sw.zip`

Directory importanti ai fine dell'installazione sono:

- **classes** contiene i file java `*.class`, il file `tuProlog 2p.jar` (versione 1.2.1), mentre la sottocartella `Alice` contiene i file di TuCSoN versione 1.4.0. Contiene il file `lancia.bat` per iniziare la simulazione

- **src** contiene i file sorgente `*.java`
- **theories** contiene le teorie Prolog in formato `*.pl`
- **context** contiene il file `*.src` che setta il contesto

Giunti a questo punto vi sono due modi alternativi per iniziare la simulazione: o da riga di comando del prompt oppure con il file

```
lancia.bat
```

contenuto nella directory **class**.

La simulazione necessita di un nodo TuCSoN attivo in locale, ovvero sulla macchina sulla quale avviene l'esecuzione del programma.

Se si usa il file `lancia.bat` per far partire la simulazione, non è necessario attivare prima il nodo TuCSoN, in quanto è messo in esecuzione in automatico attraverso il comando:

```
start java -cp ..\classes alice.tucson.runtime.Node
```

Quindi si passa a lanciare l'esecuzione vera e propria del programma.

Ancora una volta se si è scelto di usare il file

```
lancia.bat
```

tutto è eseguito in automatico: non resta che osservare ed analizzare i dati della simulazione.

Da riga di comando è invece necessario lanciare il file

```
FabrizioCastori.class
```

che contiene il main. Il comando è il seguente:

```
java -cp dir \classes FabrizioCastori
```

La simulazione ha inizio ed in pochi secondi si conclude, producendo due tipi di output: grafico e su linea di comando.

Capitolo 2

Output su linea di comando

Su linea di comando sono illustrati passo-passo i movimenti degli agenti giocatori e relative considerazioni degli stessi.

Innanzitutto ogni agente osserva la bandierina del fuorigioco, cioè il flag `fine(X)` e ne stampa a video il contenuto. Se non vi è offside in partenza di simulazione l'output sarà il seguente:

```
fine vale: fine(0)
```

```
Il flag vale: 0
```

```
ripetuto 6 volte.
```

Quindi vengono illustrate le posizioni di partenza dei 6 giocatori.

```
Posizione di partenza di Rea: (367,288)
```

```
Posizione di partenza di Bocchini: (465,228)
```

```
Posizione di partenza di Ficagna: (373,152)
```

```
Posizione di partenza di Ambrogioni: (259,64)
```

```
Posizione di partenza di Cavalli: (282,291)
```

```
Posizione di partenza di Bernacci: (75,157)
```

Per ulteriore chiarezza, è stampata a video anche la percezione istante per istante che ogni difensore ha di sé, dei propri compagni ed avversari. Naturalmente, salvo errori, si devono produrre 4 output uguali (uno per difensore), che sono però il frutto di 4 diverse osservazioni dello spazio di tuple contenente tutte le posizioni in campo. La percezione degli attaccanti non interessa in questo frangente.

```
fuorigioco(rea(367,-,288), bocchini(465,-,228),
```

```
ficagna(373,-,152), ambrogioni(259, -,64),
cavalli(282,291,0), bernacci(75,157,1)).
```

E' importante notare però che mentre per i difensori la tupla difensore(X,-,Y) esprime le coordinate sugli assi X e Y del difensore, nel caso degli attaccanti la tupla è la seguente: attaccante(X,Y,Palla) dove X e Y hanno il significato precedente, mentre Palla indica se l'attaccante è in possesso o meno della palla. In questa simulazione si è deciso che Bernacci avrà sempre la palla, ma i difensori questo non lo sanno, e si limitano a leggere il contenuto della variabile Palla. Ogni giocatore decide quindi come muoversi in base alle proprie considerazioni ed esplicita a video tale scelta:

```
*****
Bernacci e' avanzato da: 75 a 105
La posizione di Bernacci vale: (105,157)
*****
Cavalli e' avanzato da: 282 a 312
La posizione di Cavalli vale: (312,291)
*****
Rea e' avanzato da: 367 a 367
Posizione di Rea: (367,288)
*****
Ficagna e' avanzato da: 373 a 373
Posizione di Ficagna: (373,152)
*****
Ambrogioni e' avanzato da: 259 a 259
Posizione di Ambrogioni: (259,64)
*****
Bocchini e' avanzato da: 465 a 435
Posizione di Bocchini: (435,228)
```

Dopo essersi mossi tutti i giocatori leggono la tupla fine(X) per sapere se continuare a muoversi. Ogni agente stampa a video il valore della variabile X del flag(X). Se il contesto non decreta né il fuorigioco né determina la fine della simulazione perchè i difensori hanno deciso di andare a contrasto, l'output sarà il seguente

0

ripetuto sei volte. Varrà invece

1

in caso di fuorigioco ripetuto 6 volte. Sempre per 6 volte, in caso di contrasto si leggerà:

2.

In ogni caso alla fine della simulazione i giocatori agenti ristampano sia posizione di partenza che posizione di arrivo nel seguente formato.

Posizione di partenza di Rea: (367,288)

Posizione di arrivo di Rea: (367,288)

Posizione di partenza di Bocchini: (465,228)

Posizione di arrivo di Bocchini: (345,228)

Posizione di partenza di Ficagna: (373,152)

Posizione di arrivo di Ficagna: (345,152)

Posizione di partenza di Ambrogioni: (259,64)

Posizione di arrivo di Ambrogioni: (259,64)

Posizione di partenza di Cavalli: (282,291)

Posizione di arrivo di Cavalli: (402,291)

Posizione di partenza di Bernacci: (75,157)

Posizione di arrivo di Bernacci: (195,157)

Quando il controllo passa alla parte grafica, vengono stampati a video i parametri ricevuti per ogni giocatore: Xstart, Ystart, Xend e Yend.

Dentro il Manuzzi: 367 288 367 288

Dentro il Manuzzi: 465 228 345 228

Dentro il Manuzzi: 373 152 345 152

Dentro il Manuzzi: 259 64 259 64

Dentro il Manuzzi: 282 291 402 291

Dentro il Manuzzi: 75 157 195 157

In tal maniera è possibile rileggere passo-passo l'evoluzione del sistema e catturare eventuali errori.

Capitolo 3

Output grafico

Appare l'immagine di una metacampo di un rettangolo di gioco da calcio, con il cerchio di centrocampo sulla sinistra e l'area di rigore a destra. L'origine degli assi ($x=0, y=0$), è fissata nell'estremo superiore sinistro del pannello.



Figura 3.1: Marco Ambrogioni, terzino destro, maglia numero 2

La legenda che indica i giocatori è la seguente: il pallino con numero indica la posizione di partenza, la maglietta con il numero indica le coordinate di arrivo.

I difensori sono rappresentati con maglietta bianca e numeri neri, colorati gli attaccanti. Nel dettaglio Bernacci, portatore di palla, maglia rossa e numero bianco, Cavalli, senza palla, maglia blu e numero bianco.



Figura 3.2: Riccardo Bocchini, difensore centrale, maglia numero 5



Figura 3.3: Daniele Ficagna, difensore centrale, maglia numero 6



Figura 3.4: Angelo Rea, terzino sinistro, maglia numero 13



Figura 3.5: Simone Cavalli, attaccante senza palla, maglia numero 10

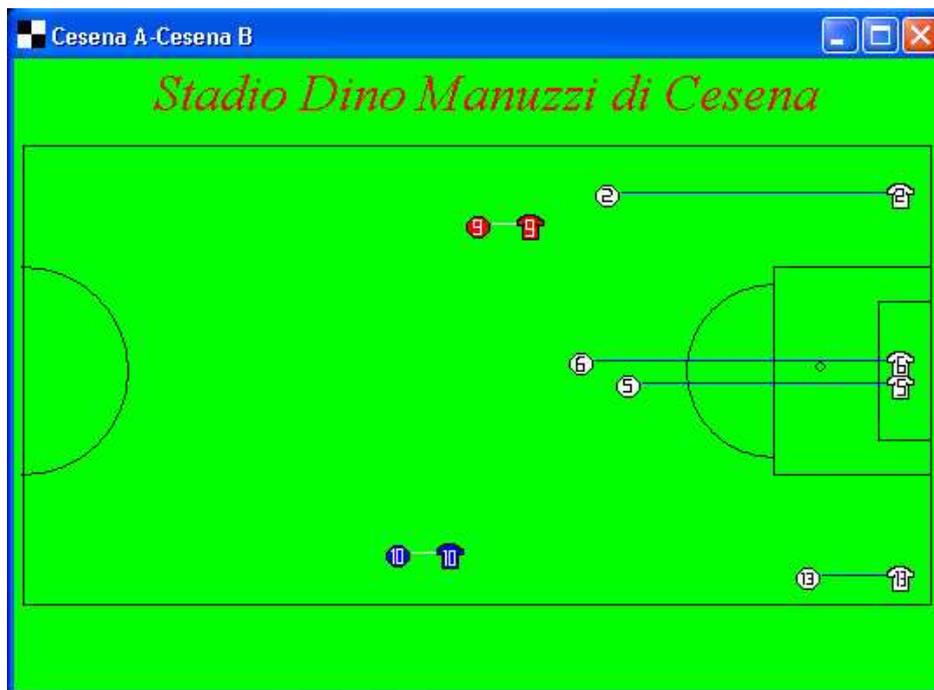


Figura 3.8: I difensori si preparano al contrasto

Una linea colorata, azzurra per i difensori e bianca per gli attaccanti, esplicita la traiettoria seguita. Nel caso in cui i difensori decidano di andare a contrasto, la loro posizione per default sarà quella di fondocampo. Non si tratta ovviamente di una posizione reale, che ha il solo scopo di comunicare il particolare esito della simulazione.