

Basketball simulator

Lisa Cattalani
Università di Cesena
Cesena, Italy
lisa.cattalani@studio.unibo.it

July 28, 2019

Contents

1	Introduzione	3
2	Netlogo	3
2.1	Storia	3
3	Analisi del sistema	3
3.1	Semplificazione del sistema	4
4	Modellazione del sistema	4
4.1	Agenti e Artefatti	4
4.2	Agenti a stati mentali	4
4.3	Modello del sistema	5
4.3.1	Comportamenti	6
4.3.2	Comportamento tattico	6
4.3.3	Comportamento aggressivo	6
5	Scelte implementative	6
6	Conclusioni	7

1 Introduzione

Il progetto implementa un sistema multi agente che simula una partita di basket. Vi sono due squadre che si fronteggiano: una squadra rossa ed una blu, ognuna composta da 5 giocatori. Il sistema è stato costruito modellando il comportamento delle singole parti (i giocatori) e le azioni da loro intraprese sono semplici, visibili e giustificabili. Lo scopo del progetto non è quello di costruire un sistema che simula la realtà ma quello di analizzare l'andamento della partita (comportamento globale) avendo codificato solamente il modo d'agire (comportamento locale) dei giocatori. Il risultato ha superato di gran lunga le mie più rosee aspettative, i giocatori: si muovono, si affrontano, si passano la palla, rubano la palla e riescono a segnare. La varietà del gioco è ampia e si possono vedere diversi passaggi, tiri e movimenti.

Il progetto è stato implementato in NetLogo, un ambiente integrato per lo sviluppo e la simulazione di sistemi mediamente complessi che mi ha permesso da un lato di sperimentare l'autonomia di agenti semplici come parti di un sistema, dall'altro di osservare, analizzare e studiare i comportamenti emergenti di tale sistema. NetLogo si è dimostrato uno strumento potente, con un linguaggio flessibile e potente per costruire simulazioni. La semplicità di NetLogo lo rende ideale per simulazioni veloci, ma quando si devono affrontare situazioni molto complesse che richiedono migliaia di righe di codice, la mancanza di strutture più rigide lo rendono uno strumento inadatto.

2 Netlogo

2.1 Storia

Il software NETLOGO, sviluppato originariamente da Uri Wilensky nel 1999 presso il Center for Connected Learning and Computer-Based Modeling della Northwestern University <http://ccl.northwestern.edu/netlogo/>, offre un ambiente di sviluppo ideale per la realizzazione di modelli di simulazioni ad agenti, di networks e di sistemi dinamici. All'interno di NETLOGO è possibile riprodurre molte delle caratteristiche di un sistema complesso, studiare la sua evoluzione nel tempo e visualizzarla in tempo reale all'interno di un laboratorio virtuale 2D o 3D. NETLOGO è stato scritto in JAVA ma è completamente programmabile per mezzo di un meta-linguaggio ad oggetti semplificato basato sulla sintassi del Logo (S.Papert, anni 60) ed introdotto originariamente nell'ambiente StarLogo, il progenitore di NetLogo creato all'MIT nel 1990 (M.Resnick). L'ambiente di sviluppo di NETLOGO interpreta direttamente il codice senza bisogno di compilarlo, quindi al suo interno è possibile, in tempo reale:

- interagire con il sistema attraverso pulsanti e sliders per modificare i parametri di controllo;
- visualizzare variabili, grafici e istogrammi relativi alla simulazione;
- importare/salvare immagini e dati da/su files esterni;

E' possibile inoltre:

- effettuare set di esperimenti al variare delle condizioni iniziali o dei parametri di controllo;
- lanciare le simulazioni più pesanti in modalità off-line su computer remoti;
- usufruire di una ricca libreria di modelli di esempio da cui prendere spunto e di un help pratico ed ipertestuale;
- salvare le proprie simulazioni come applet JAVA e visualizzarle immediatamente dentro un web browser.

3 Analisi del sistema

Analizzando il sistema proposto risulta subito evidente che vi è un' entità fondamentale e sono i giocatori.

I giocatori, decidono in autonomia le azioni da intraprendere, e cioè se attaccare, se difendere, se passare la palla, segnare, cercare di rubare la palla o commettere fallo provandoci. Ovviamente la scelta terrà conto anche degli elementi esterni, e cioè la presenza di altri giocatori o l'eventuale possesso palla. Un'altra entità è la palla che è sempre legata ad un giocatore e che si muove con esso.

3.1 Semplificazione del sistema

Al fine di analizzare propriamente il comportamento emergente del sistema siamo costretti a modellare una versione semplificata della realtà, scartando dettagli che turberebbero le nostre osservazioni (e conclusioni). Il tema, quello del basket, ci permette di mantenere la complessità dei singoli agenti molto bassa. Un giocatore è molto semplice: esso ha una visione parziale del campo da gioco e il suo comportamento è un insieme di meccanismi di coordinazione e interazione locale. Vogliamo capire se il comportamento globale (a livello di squadra) è più articolato e interessante rispetto alla somma dei singoli componenti locali (a livello di singolo giocatore). Per mantenere bassa la complessità del progetto abbiamo eliminato molti dei dettagli e delle regole del basket. I cambiamenti significativi adottati sono:

- assenza della stragrande maggioranza delle infrazioni di gioco, quali : fallo d'attacco (sfondamento), fallo tecnico, fallo antisportivo, passi, doppio, tre secondi, ventiquattro secondi, otto secondi, campo.
- semplificazione del fallo di difesa: il fallo divensivo esiste e viene commesso quando un giocatore si trova vicino ad giocatore avversario (sia esso provvisto o sprovvisto di palla), ma non dà diritto ad una eventuale rimessa o tiro libero, e il giocatore può commettere anche più di 5 falli.
- assenza di rimesse: ogni qualvolta viene fatto canestro, i giocatori ripartiranno direttamente da metà campo.
- assenza di cambi: vi saranno solo 5 giocatori per ogni squadra che giocheranno sempre durante la partita.
- rimbalzo: nel caso di un tiro sbagliato, la palla viene automaticamente assegnata ad un giocatore avversario (di preferenza quello più vicino alla palla).
- la palla non esce mai dal campo ed appartiene sempre ad un giocatore.

4 Modellazione del sistema

Per effettuare la modellazione vera e propria del sistema, si è partiti individuando il framework concettuale che più si adattasse al sistema in esame, così da poter riutilizzare strumenti e modelli già noti e adeguatamente testati.

4.1 Agenti e Artefatti

Il sistema può essere visto come un insieme di entità computazionali autonome (i giocatori) che si muovono a seconda dei propri obiettivi in un campo da gioco. In questo caso il framework concettuale per Agenti& Artefatti ben si presta a descrivere tale sistema. In questo framework si distinguono tre principali entità. Gli **agenti** sono componenti pro-attive che svolgono autonomamente qualche attività, in questo caso rappresentano i giocatori; gli agenti possono modificare l'ambiente (ad esempio tirando). Un agente vive immerso in un MAS (Sistema Multi-Agente), dove interagisce con gli altri agenti e l'ambiente in cui sono situati. Gli **artefatti** sono le componenti passive del sistema e che un agente può decidere di utilizzare a proprio vantaggio per portare a termine i propri obiettivi, tale ruolo si è deciso di assegnarlo alla palla; questo elemento è trasparente agli agenti, che ne possono usufruire tramite un'interfaccia, e permette agli agenti di effettuare modifiche all'ambiente. La palla definisce chi delle due squadre ne è in possesso e quindi chi sta attaccando o chi sta difendendo. Il campo da gioco costituisce l'**ambiente** fisico in cui sono presenti e si muovono sia i giocatori che la palla, fornendo così il concetto di canestro avversario, proprio canestro, metà campo e posizione nello spazio che tutti gli agenti condividono: in questo modo essi possono intraprendere azione in base al proprio essere e in relazione ai propri obiettivi.

4.2 Agenti a stati mentali

Un ulteriore aiuto nella modellazione del sistema può pervenire dalla modellazione degli agenti tramite stati mentali. Un agente intelligente è un sistema computazionale in grado di percepire

l'ambiente e di effettuare azioni autonomamente, principalmente per cambiarlo secondo i propri obiettivi: spiegando un agente in termine di stati mentali, è possibile fornirgli una rappresentazione simbolica del mondo in cui vive e permettergli di ragionarci sopra. Le azioni che egli compirà saranno in base alla sua percezione attuale del mondo. Gli stati mentali di un agente possono essere di tipo epistemico (ciò che conosce l'agente) o motivazionale (ciò che vuole realizzare l'agente). Per mantenere la semplicità del sistema, gli agenti saranno modellati come teleo-reattivi, ovvero gli obiettivi da raggiungere non vengono direttamente esplicitati all'agente ma emergono dal suo comportamento, che viene pianificato dal programmatore.

4.3 Modello del sistema

Come già detto, il programma simula la sfida tra due squadre: la squadra rossa e la squadra blu. Ci sono cinque giocatori per parte che si possono muovere liberamente nel campo. Per rendere più veritiero il gioco, ogni giocatore conosce solamente ciò che succede in un raggio di pochi metri, detto raggio d'azione, e prende decisioni sulla base di questa sua visione parziale del mondo. Al di fuori di quel che accade nel suo raggio d'azione un giocatore non ha informazioni sullo stato della partita. Nella figura sottostante



è esplicitato questo aspetto: i due avversari non sono a conoscenza della presenza l'uno dell'altro perchè al di fuori dei reciproci raggi d'azione. Immaginiamo che i giocatori abbiano due obiettivi. Il primo punta al proprio canestro, il secondo punta al canestro avversario. Queste due informazioni, anche se possono sembrare eccessive, servono per costruire un comportamento verosimile: nella realtà, i giocatori non si muovono lungo direzioni casuali, ma cercano di avvicinarsi il più possibile al canestro avversario; quando invece deve difendere, tornerà ad una posizione prestabilita e cioè il proprio canestro o andare verso il possessore di palla. La partita si svolge a turni successivi. Ad ogni turno, un giocatore osserva ciò che sta accadendo intorno a lui e sceglie un comportamento tra i 7 possibili. Di seguito una tabella riepilogativa:

Comportamento	Descrizione
steal-ball:	la decisione di rubare la palla all'avversario, o , in mancanza, avvicinarsi all'avversario che ha la palla.
attack	attaccare il canestro.
shoot	tirare a canestro, se si tira fuori dall'area di tre punti, il canestro varrà 3, altrimenti 2.
move	muoversi, se si ha la palla si tenderà a muoversi verso il canestro.
defense	si tenderà o ad andare incontro al possessore di palla avversaria oppure ad andare verso il proprio canestro per difenderlo.
foul	verrà commesso un fallo nei confronti del giocatore avversario.
pass	verrà passata la palla di preferenza al compagno più vicino, altrimenti ad un compagno qualsiasi.

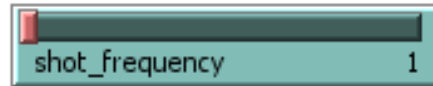
La scelta avviene tramite il metodo Montecarlo: ad ogni comportamento viene assegnato un punteggio e quello con il punteggio più alto ha più probabilità di essere eseguito. In generale, la vicinanza di avversari o la prossimità del canestro rendono un comportamento più possibile di un altro. Ad esempio, se il giocatore che possiede la palla nota intorno a lui (nel suo raggio d'azione) molti avversari, un passaggio potrebbe essere un'ottima scelta.

Oppure quando un giocatore si trova al limite dell'area avversaria (il canestro si trova quindi all'interno del suo raggio d'azione) dovrebbe invece provare il tiro. Abbiamo classificato i comportamenti principali in due categorie: comportamenti aggressivi e comportamenti tattici. Nelle sezioni seguenti descriveremo in dettaglio ognuno di questi.

4.3.1 Comportamenti

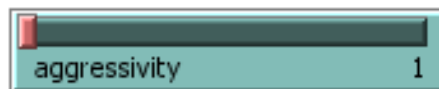
Le varie tipologie di comportamenti che possiamo trovare dipendono molto anche dagli slider che applichiamo ai giocatori.

4.3.2 Comportamento tattico



Se viene aumentato slider del tiro, farà aumentare al giocatore possessore della palla la volontà di tirare. Può essere una buona tattica se si vuole cercare di recuperare punti in breve tempo.

4.3.3 Comportamento aggressivo



Se viene aumentato lo slider dell'aggressività la decisione di fare fallo/cercare di rubare la palla saranno maggiori. Questo comportamento viene classificato come un atteggiamento più aggressivo del giocatore, che cercherà infatti di rubare più volte palla oppure commetterà più falli, ma più si recupera palla più si ha la possibilità di poter attaccare e quindi fare canestro.

Se quindi le due barre venissero utilizzate al massimo, avremo un gran numero di falli e di tiri e quindi di fatto l'azione si concentrerebbe molto verso la metà campo.

In generale, un giocatore è più portato ad attaccare e in caso tirare solo se abbastanza vicino al canestro oppure a passare la palla se ha un avversario vicino, altrimenti continuerà ad avanzare.

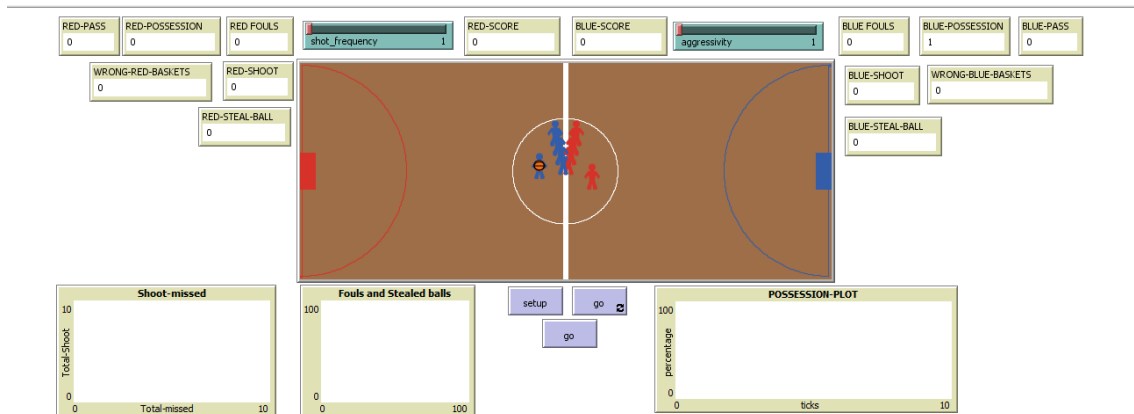
5 Scelte implementative

Come già detto, per implementare il progetto, si è scelto come ambiente di programmazione NetLogo nella versione 6.0.4. In NetLogo l'ambiente è costituito da una griglia di coordinate del tipo XY, sulle quali vivono gli agenti, che sono di 3 tipi:

- le **turtles** sono agenti in grado di muoversi, cambiare colore, forma e proprietà.
- le **patches** sono agenti fissi ad una data posizione, rappresentati da quadrati di medesima dimensione, possono cambiare colore.
- l'**observer** è un agente esterno alla griglia di coordinate, in grado di fornire istruzioni a turtles e patches e col quale osservare l'evoluzione del modello.

Una funzionalità base di NetLogo è lo scandire delle azioni degli agenti tramite tick ("momenti"): un agente viene selezionato in maniera casuale e compie tutte le proprie azioni prima di passare il controllo all'agente successivo; al termine di un tick, tutti gli agenti hanno compiuto quindi uno alla volta le proprie azioni. Questo metodo è stato sfruttato per simulare l'idea di tempo condiviso tra tutti gli agenti: ognuno compie un'unica azione pratica, basata sul proprio stato, e quindi aggiorna anche di conseguenza le variabili globali a seconda dell'azione scelta.

Il nostro mondo inizialmente si presenterà nel seguente modo:



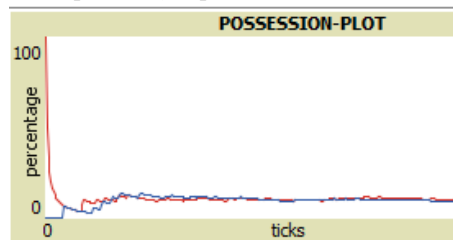
Quando il pulsante setup verrà premuto, permetterà di resettare tutte le variabili globali e assegnare la palla casualmente ad uno dei giocatori delle due squadre. Vi sono due tasti go, il primo, quello conente le due freccette, permette di far partire i tick e aggiornarli in modo continuo; l'altro invece ogni volta che viene premuto corrisponde un tick.

6 Conclusioni

Terminata l'implementazione e il testing, il sistema è stato avviato più volte così da verificare i risultati ottenuti ed annotare eventuali deduzioni.

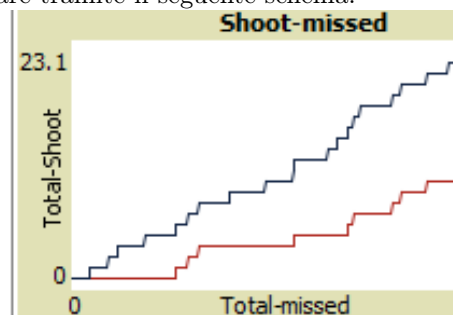
Possiamo notare dei passaggi tra due compagni di squadra, dei tiri e delle azioni in solitaria che si terminano in canestri, oltre a questo, vi sono a riguardo diverse annotazioni da fare.

Primo, si è notato che il possesso palla tra le due squadre è sempre più o meno uguale, era abbastanza prevedibile visto che ad ogni tiro corrisponde un cambio possesso e che comunque, anche se una squadra ruba palla e poi tira, il possesso ritorna comunque alla squadra avversaria.



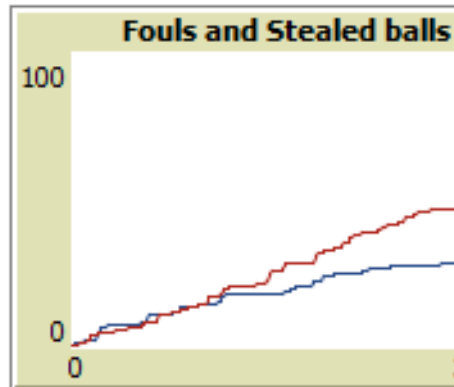
In generale questo comportamento è anche molto simile a quello reale, infatti il possesso palla delle due squadre sarà sempre più o meno uguale, perchè se ad ogni palla rubata corrisponde un canestro, in automatico il possesso ritorna alla squadra avversaria, ciò che invece fa la differenza sono le palle rubate.

Secondo, il numero di tiri fatti e quelli segnati/non segnati, varia, può variare di molto oppure a tratti essere anche uguale, infatti ogni tiro ha il cinquanta per cento di probabilità di essere realizzato, nel complesso abbiamo che ovviamente il numero dei tiri totali sarà maggiore di quelli sbagliati, lo possiamo osservare tramite il seguente schema:



Dove le linee blu indicano il numero di tiri totali e la linea rossa il numero di tiri sbagliati.

Terzo, uno schema dei falli commessi e delle palle rubate anche qui il numero dei falli commessi è maggiore rispetto alle palle rubate, in quanto è molto più facile commettere fallo che rubare palla.



Quarto, se i due slider, dell'aggressività della frequenza di tiro, venissero portati al massimo, avremo una tipologia di gioco che si concentrerà maggiormente a metà campo, questo perchè la scelta di tirare e di commettere fallo/rubare palla sarà maggiore e quindi vi saranno più tiri e meno spostamenti percorsi, incentrando maggiormente il gioco al centro del campo.

Più in generale tutte le varie statistiche si possono anche desumere dai vari monitor presenti, che comunque ci permettono di simulare e di avere valori abbastanza veritieri, infatti in una partita di basket è praticamente impossibile che non vengano commessi falli, che vengano sempre realizzati dei canestri, o che non vengano mai rubate palle agli avversari.