

Simulazione Quiddich

Realizzato nel corso di Sistemi Autonomi

lisa.cesario greta.sasso

July 2017

1 Introduzione

Lo scopo di questo progetto è quello di modellare e di implementare un sistema multi-agente in grado di simulare una partita del gioco magico del Quiddich.

L'obiettivo della simulazione non è quello di costruire un ambiente che simuli la realtà, bensì di analizzare il comportamento globale dei giocatori che emerge, avendo modellato solo comportamento locali.

Durante questa breve relazione metteremo in risalto aspetti fondamentali della teoria che abbiamo cercato di applicare nella simulazione

il risultato è stato ...

La tecnologia scelta per la parte implementativa del progetto è NetLogo,

Questi sono i passi che abbiamo seguito per lo studio di simulazione

- Analisi del sistema reale, con l'obiettivo di comprenderne le caratteristiche
- Costruzione di un modello dal sistema reale, evitando aspetti non rilevanti per la simulazione
- Costruzione di una rappresentazione eseguibile del modello
- Analizzare gli output della simulazione comparandoli con il sistema reale

2 Analisi del sistema reale

2.0.1 Il Quiddich

Il Quiddich è un gioco magico inventato da con la palla che si gioca a cavallo di manici di scopa volanti. L'ambiente di gioco è costituito da un campo di forma ovale e tre grandi cerchi [2] posti sui pali agli estremi del campo. Al centro c'è un cerchio dove l'arbitro libera le quattro palle all'inizio della partita. Per giocare sono necessarie tre tipi di palle:

- La **Pluffa** è una palla di cuoio rosso con diametro pari a 30 centimetri. La Pluffa viene utilizzata dai giocatori di tutte e due le squadre per segnare i goal nei tre cerchi posti sui pali agli estremi del campo. Ogni goal vale 10 punti.
- I **Bolidi** sono delle palle stregate di pesante ferro nero dal diametro di 25,4 centimetri che svolazzano da sole per il campo cercando di disarcionare i giocatori dai loro manici di scopa (di solito puntano al giocatore più vicino). Sono 2.
- Il **Boccino d'Oro** è una palla incantata dorata dal diametro di una noce e dotata di ali. Ogni squadra mette in campo un proprio giocatore, il Cercatore, il cui compito è darle la caccia e cercare di prenderla. È molto veloce ed è così piccola che a stento la si vede, e per questo motivo i Cercatori sono solitamente i giocatori più piccoli ed agili. La cattura del Boccino segna la fine della partita, ed alla squadra del Cercatore che è riuscito a catturarlo vengono assegnati 150 punti. La cattura del Boccino è l'unica causa che pone fine alle partite, che hanno quindi una durata potenzialmente illimitata

Esattamente come nel calcio, si affrontano due squadre che hanno come obiettivo quello di accumulare più punti possibili prima della fine della partita. Ogni squadra è composta da 7 giocatori:

- **3 Cacciatori** si passano la Pluffa e tentano di farla entrare negli anelli avversari per segnare un gol. Questi giocatori devono essere molto forti fisicamente e avere ottime capacità visive. Devono inoltre stare attenti ai Bolidi che potrebbero spuntare all'improvviso e disarcionarli dalle loro scope.
- **2 Battitori** sono giocatori muniti di una corta mazza di ferro che usano per svolgere il loro compito: allontanare i Bolidi dai loro compagni di squadra. Possono colpire i Bolidi e indirizzarli verso gli avversari per disarcionarli.
- **1 portiere** ha il compito di difendere i tre anelli, impedendo che i Cacciatori avversari segnino dei gol. I Portieri non escono quasi mai dall'area contenente gli anelli.
- **1 cercatore** è il giocatore che ha il compito di acciuffare il Boccino d'Oro. Sono soggetti a molti falli.

La partita termina quando un Cercatore prende il Boccino d'Oro. Vince la squadra che ha accumulato più punti

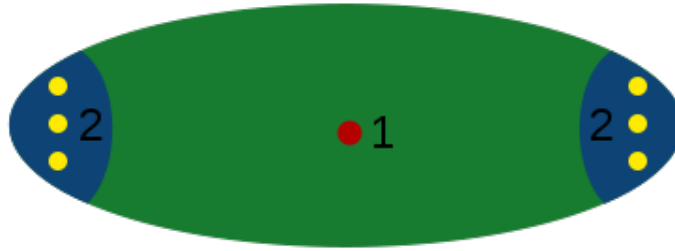


Figura 1: Struttura del campo di Quiddich. Al centro in rosso indicato dal numero 1 troviamo la Pluffa; ai lati i tre cerchi gialli indicano i tre anelli in cui fare gol

2.0.2 Semplificazioni del sistema

Per poter mantenere le dimensioni e la complessità del progetto adeguati, abbiamo preso in considerazione alcune semplificazioni del gioco. In particolare:

- Il Quiddich originale si svolge su 3 dimensioni, in quanto i giocatori volando si muovono anche in altezza, noi abbiamo scelto di lavorare in un mondo bidimensionale;
- Non prenderemo in considerazione i falli;
- Il campo sarà rettangolare e non ovale;
- I bolidi sono entità autonome che possono creare falli verso alcuni giocatori, senza un motivo;
- Lo scopo dei giocatori detti “Battitori” è quello di difendere i membri della propria squadra dai bolidi, e di attaccare gli avversari indirizzando i bolidi verso questi. Per semplicità abbiamo implementato solamente il comportamento dei bolidi, tralasciando quello dei battitori;
- Quando la Pluffa esce dal campo, il gioco si interrompe riportando i giocatori nello stato di partenza del gioco / azione;
- Quando il Boccino o i Cercatori si avvicinano alla fine del campo di gioco abbiamo supposto l’esistenza di un muro magico, che chiameremo "passaporta", attraverso il quale queste entità vengono riportate in una posizione casuale compresa nell’area di gioco;
- Il portiere si muove solo sull’asse delle y.

3 Modello del sistema

Prima di calarci nella modellazione del sistema simulato è necessario effettuare un'analisi dei requisiti che il nostro sistema dovrà soddisfare, considerando il quadro teorico concettuale che vogliamo seguire.

3.1 Autonomia e Agenti

Nella scienza dei sistemi ad agenti l'autonomia è l'unica fondamentale feature di un agente. Esistono tuttavia diverse definizioni del concetto di "autonomia" in base al contesto di cui ci si occupa. Di seguito sono riportate le due definizioni che a nostro parere rispecchiano maggiormente la visione del sistema ad agenti da noi definito.

- Un agente si definisce *socialmente autonomo* se :
 - Ha un obiettivo globale, e dei sotto goal per raggiungerlo;
 - è capace di prendere decisioni riguardanti la scelta di certi obiettivi, considerando casi di obiettivi non compatibili, o che creano conflitti con altri agenti;
 - Gli obiettivi di un agente possono essere modificati solo dall'agente stesso;
 - Le modifiche ai belief di un agente, così come i piani e le azioni, devono essere decisi a design-time per poter essere poi attuati in autonomia dall'agente a run-time.
- Un agente si definisce *computazionalmente autonomo* se :
 - incapsula il thread di controllo;
 - il controllo non passa attraverso i confini dell'agente, ma solo la conoscenza del mondo;
 - non ha un'interfaccia, non può essere controllato o invocato.

Inoltre possiamo definire un Multi Agent System come un aggregato di entità di controllo multiple e distinte, che interagiscono tra di loro scambiandosi informazioni.

In letteratura un agente intelligente è un sistema computazionale in grado di svolgere azioni autonome e di avere una qualche percezione in un qualche ambiente. In questa definizione emergono due elementi cardine delle proprietà di un agente :

- La percezione è quell'abilità di osservare l'ambiente. L'output di una funzione di percezione è un processo attraverso il quale l'agente riconosce lo stato dell'ambiente, e si prepara per adattare il suo comportamento ad esso;

- Le azioni sono selezionate all'interno di un piano, scelto per raggiungere un sotto-goal o un goal, e hanno l'obiettivo di cambiare l'ambiente . La funzione di azione rappresenta il processo di decisione.

Sulla base di queste premesse concettuali, abbiamo facilmente individuato nei giocatori della partita gli agenti, entità autonome, in quanto: tramite una conoscenza parziale del mondo, cercano di soddisfare l'obiettivo di fare più punti possibili, cioè di portare la pluffa nella porta avversario, e nell'evitare il goal della squadra avversaria. Questo avviene con il raggiungimento di sotto goal scelti come conseguenza dell'analisi dello stato conosciuto del mondo, in un dato momento, e eseguiti tramite delle azioni che definiscono un piano per il raggiungimento dei sotto-goal selezionati. In base a queste premesse è spontaneamente nata la decisione di sviluppare il nostro sistema multi-agente secondo il framework BDI.

3.2 Architettura BDI

Ogni entità attiva del nostro sistema è modellata attraverso il framework Belief-Desires-Intentions.

BELIEF rappresentano la conoscenza corrente del mondo, includendo :

- Informazioni Strutturali: sono informazioni relative alla posizione e alle caratteristiche fisiche dell'agente
- Settaggi di gioco : Informazioni che influiscono nelle decisioni che l'agente prenderà per raggiungere i sotto-goal e i goal
- Conoscenza dell'ambiente : informazioni sullo stato corrente dell'ambiente inferite dai mezzi di percezione, in particolare attraverso il mezzo "visivo". Gli agenti avranno una parziale conoscenza dell'ambiente, in quanto, come accade nella realtà non riescono ad avere una visione totale del campo di gioco.
- Stato Interno: Informazioni relative alla storia passata che l'agente ha memorizzato

DESIRES Rappresentano lo stato del mondo che l'agente sta cercando di raggiungere

INTENTIONS Sono scelte per raggiungere il desiderio dell'agente e sono generalmente implementate come piani (Insieme di azioni) e post-condizioni

3.3 Entità Attive

Qui di seguito saranno elencate tutti gli agenti attivi del sistema e per ciascuna di loro verrà fornita una breve descrizione più una tabella con i *belief* i *desires* e le *intentions*.

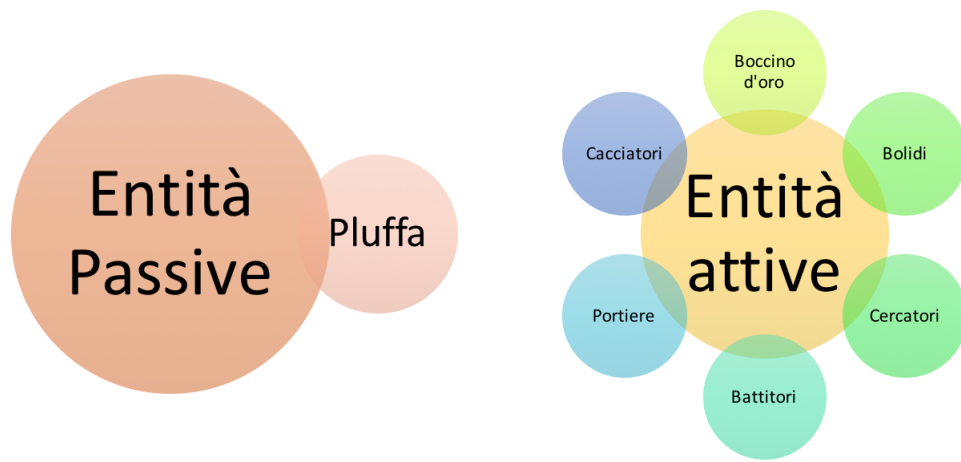


Figura 2: Entità del nostro sistema

3.3.1 Observer

L'Observer avrà il ruolo dell'arbitro di gioco. Essendo una entità attiva esterna, conscia sia di quello che succede nell'ambiente, ma anche dei comportamenti e delle posizioni di tutti i giocatori, sarà responsabile di diversi compiti.

- In primo luogo dovrà occuparsi di monitorare la Pluffa. Questa, essendo una entità passiva, non è consapevole del fatto di essere all'interno dell'area goal di una squadra, o di essere fuori dal campo di gioco. Pertanto sfruttiamo l'arbitro, per tre situazioni :
 - il controllo della presenza di un goal;
 - il controllo della fuoriuscita della Pluffa dal campo di gioco;
 - il movimento della Pluffa in base all'angolo e alla velocità settati dalle altre entità del gioco ;
- In secondo luogo, avrà il compito di controllare anche la cattura del Boccino d'Oro da parte di uno dei due Cercatori. Questo avvenimento comporta anche la fine della partita, l'accreditamento dei punti, e di conseguenza il decretamento del vincitore. L'azione di cattura del Boccino è un evento che prescinde dai ruoli dell'arbitro: sono i Cercatori che si contendono la presa in modo del tutto indipendente da ogni altra entità del sistema. Nasce quindi la necessità di una forma di comunicazione tra il Cercatore e l'arbitro. – forse questo lo metterei in implementazione. Questo si attua tramite una interrogazione da parte dell'arbitro nei confronti dei cercatori, a cui viene chiesto se hanno catturato il boccino;
- Inoltre ha il ruolo di controllare l'andamento della partita, in particolare affidando il “turno” di gioco a ogni giocatore, permettendo loro di eseguire il comportamento

3.3.2 Portieri



Figura 3: Icona che rappresenta un Portiere

I Portieri sono entità attive che hanno il compito, in modo analogo ai colleghi del calcio, di proteggere la porta da eventuali tiri avversari.

Name	Description
Proprietà Strutturali	
team	- Valori: - o "SERPEVERDE" "GRIFONDORO" - Corrisponde al nome della squadra di appartenenza
starting-x , starting-y	- Il valore della coordinata della posizione iniziale, espressa come numero intero
Starting-heading	- Il valore iniziale di orientamento del giocatore
Angle	- Direzione che il giocatore prende durante la partita per muoversi
pos-y	- Coordinata y del portiere, in quanto esso si muove solo lungo quest'asse
Settaggi di gioco	
eyes	- È la visuale del giocatore. Ogni giocatore infatti ha una conoscenza parziale dell'ambiente. Anche questo valore è modificabile tramite lo slider "visual-range" presente nell'interfaccia grafica
Conoscenza dell'ambiente	
see-pluffa	- Assume valore 1 quando la pluffa è nel raggio visivo, quindi quando la distanza del giocatore dalla pluffa è minore del belief eyes. Altrimenti assume valore 0.
n-near-teammates-range , n-near-opponents-range	- Conteggio degli altri giocatori (escluso sé stesso) della propria squadra, o avversari, che sono presenti in un range ridotto rispetto alla visuale del cacciatore
n-patch	- Conteggio del numero di giocatori (escluso sé stesso) che sono immediatamente vicini al portiere attuale.
bolide-near	- Rappresenta la presenza di un bolide nella stessa posizione del portiere, e quindi se è stato colpito da esso
pluffa-here	- Belief che può essere presente o meno. Il portiere deve sapere se la pluffa è nella sua stessa posizione, così da poterla parare o meno

Figura 4: Belief Portieri

INDIVIDUAZIONE DELLO STATO	DELIBERAZIONE	INTENTIONS [PIANI → INSIEME DI AZIONI]
• Possibilità DI PARATA	• PARARE: stabilito tramite una percentuale di successo •	• Stop-pluffa
• PLUFFA NEL PROPRIO RAGGIO VISIVO	• MOVIMENTO VERSO LA PLUFFA	• Follow-pluffa
• PLUFFA NON VISIBILE	• ATTESA	• Waiting

Figura 5: Intentions e Plan Action Portieri

3.3.3 Cacciatori



Figura 6: Icona che rappresenta un Cacciatore

I Cacciatori sono entità autonome che hanno il compito di segnare più punti possibili portando la pluffa nell'area goal avversaria. Si interfacciano principalmente tra di loro e con la Pluffa. Vista la complessità dell'entità abbiamo scelto di raggruppare in una tabella tutti i Belief di cui dispongono.

Name	Description
Proprietà Strutturali	
team	- Valori: - o "SERPEVERDE" "GRIFONDORO" - Corrisponde al nome della squadra di appartenenza
Mygoal	- L'insieme di patches rappresentante l'area in cui il giocatore deve fare goal
starting-x , starting-y	- Il valore della coordinata della posizione iniziale, espressa come numero intero
Starting-heading	- Il valore iniziale di orientamento del giocatore
Speed	- Velocità del giocatore, questa varia in base allo slider presente nell'interfaccia utente "basic-speed" e si modifica in base all'azione selezionata dal cacciatore.
Angle	- Direzione che il giocatore prende durante la partita per muoversi
Settaggi di gioco	
play-team	- Indica il livello del gioco di squadra. Prende il valore del corrispondente slider e varia per ognuna delle due squadre. Più il valore è alto e più il giocatore sarà spinto ad azioni collettive, ad esempio avrà più probabilità di passare la palla invece che continuare a muoversi con la pluffa.
eyes	- È la visuale del giocatore. Ogni cacciatore infatti ha una conoscenza parziale dell'ambiente. Anche questo valore è modificabile tramite lo slider "visual-range" presente nell'interfaccia grafica
foot	- Corrisponde al piede preferito del giocatore, destra [1] o sinistra [-1]. Questo è utilizzato per decidere l'angolo di rotazione quando un cacciatore deve effettuare un cambio di direzione, con probabilità di scelta favorevole rispetto alla preferenza memorizzata.
Stato Interno	
energy	- Ogni cacciatore ha una sua energia che varia da 0 a MAX-ENERGY. Questo per poter simulare lo sforzo di un cacciatore nell'effettuare determinate azioni, e quindi affrontare anche la necessità di risposarsi o di rallentare in una azione
time-for-recovery	- Quando il cacciatore esaurisce completamente l'energia, dovrà effettuare alcuni turni di riposo, e il numero di turni corrisponde a questo belief
Conoscenza dell'ambiente	
see-pluffa	- Assume valore 1 quando la pluffa è nel raggio visivo, quindi quando la distanza del giocatore dalla pluffa è minore del belief eyes. Altrimenti assume valore 0.
i-have-pluffa	- Se il giocatore non si sta riposando controlla se ha la pluffa sotto ai piedi, altrimenti risparmia energie ed evita questo controllo. Più giocatori, alla fine dell'aggiornamento della conoscenza possono avere questo belief attivo [con valore 1, altrimenti 0]. Quando si verifica questa situazione si effettua un "contrasto", gestito e risolto dall'observer.
n-near-teammates-range , n-near-opponents-range	- Conteggio degli altri giocatori (escluso sé stesso) della propria squadra, o avversari, che sono presenti in un range ridotto rispetto alla visuale del cacciatore
nearest-teammate, nearest-opponent	- Selezione del chasers (escluso sé stesso) più vicino al giocatore attuale , della propria squadra , e della squadra avversaria
n-opponents-patch, n-teammates-patch	- Conteggio del numero di giocatori avversari, o della propria squadra (escluso sé stesso) che sono immediatamente vicini al giocatore attuale.
pluffa-with-any	- Assume valore 1 quando attualmente un giocatore della propria squadra [incluso sé stesso] ha la pluffa sotto ai piedi . questo aggiornamento comporta una interrogazione , e quindi una comunicazione, con gli altri giocatori della propria squadra

Figura 7: Belief Cacciatori

INDIVIDUAZIONE DELLO STATO	DELIBERAZIONE	INTENTIONS [PIANI → INSIEME DI AZIONI]
FASE DI RECUPERO	RIPOSO	stop-me
ENERGIE ESAURITA	RIPOSO: avvio la fase di recupero per un certo numero di turni	stop-me
IL GIOCATORE POSSIEDE LA PLUFFA	DECISIONE secondo un certo criterio	- Scelta fra: - passaggio - possession-move "walking" "run" "normal" - tiro - Stop-me
LA PLUFFA È DEL MIO TEAM, MA NON MIA, ED È NEL MIO RAGGIO VISIVO	- COMPORTAMENTO di RECUPERO della la pluffa se in fase di passaggio/tiro, - altrimenti se in fase di possesso di un giocatore, COMPORTAMENTO D'ATTACCO verso la porta avversaria	simple-move
LA PLUFFA È DEGLI AVVERSARI, ED È NEL MIO RAGGIO D'AZIONE	- COMPORTAMENTO di ATTACCO verso la pluffa. - Nel caso in cui nono sono uno dei più vicini alla pluffa, torno a casa	simple-move
NON VEDO LA PLUFFA	RICERCA DELLA PLUFFA	simple-move

Figura 8: Intention e Plan Action Cacciatori

3.3.4 Cercatori



Figura 9: Icona che rappresenta un Cercatore

I Cercatori sono entità attive che si occupano della ricerca e della cattura del Boccino d'Oro. Non hanno interazioni con i propri compagni di squadra, nè con i membri della squadra avversaria

Name	Description
Proprietà Strutturali	
team	- Valori: - o "SERPEVERDE" "GRIFONDORO" - Corrisponde al nome della squadra di appartenenza
starting-x, starting-y	- Il valore della coordinata della posizione iniziale, espressa come numero intero
Starting-heading	- Il valore iniziale di orientamento del giocatore
Angle	- Direzione che il giocatore prende durante la partita per muoversi
Speed	- Velocità del giocatore, questa varia in base allo slider presente nell'interfaccia utente "basic-speed"
Settaggi di gioco	
eyes	- È la visuale del giocatore. Ogni cercatore infatti ha una conoscenza parziale dell'ambiente. Anche questo valore è modificabile tramite lo slider "visual-range" presente nell'interfaccia grafica
Conoscenza dell'ambiente	
boccino-captured	- Belief inizialmente non presente. È aggiunto nel momento in cui un cercatore cattura il boccino
boccino-near	- Belief aggiunto quando il boccino è nel raggio visivo del cercatore
wall-near	- Belief aggiunto quando il cercatore si avvicina a uno dei confini del mondo, detto "Muro"
bolidi-near	- Assume un valore pari al numero di bolidi che sta colpendo il cercatore.

Figura 10: Belief Cercatore

INDIVIDUAZIONE DELLO STATO	DELIBERAZIONE	INTENTIONS [PIANI → INSIEME DI AZIONI]
• RAGGIUNTO CONFINE DEL MONDO	• ATTIVAZIONE PASSAPORTA	• Passaporta-move
• BOCCINO NEL RAGGIO VISIVO	• CATTURA DEL BOCCINO	• Follow-boccino
• BOCCINO NON VISIBILE	• ESPLORAZIONE	• Exploration

Figura 11: Intention e Plan Action Cercatore

3.3.5 Il Boccino d'oro

Il Boccino d'Oro è un'entità attiva che ha il compito di non farsi catturare dai Cercatori.

Name	Description
Proprietà Strutturali	
hide	- Assume valore 1 quando il boccino non è visibile ai cercatori
starting-x, starting-y	- Il valore della coordinata della posizione iniziale, espressa come numero intero
Starting-heading	- Il valore iniziale di orientamento del boccino
Angle	- Direzione che il giocatore prende durante la partita per muoversi
Speed	- Velocità del boccino, questa varia in base allo slider presente nell'interfaccia utente "basic-speed"
Conoscenza dell'ambiente	
closest-seeker	- Individuazione del cercatore più vicino al boccino
wall-near	- Belief aggiunto quando il boccino di avvicina a uno dei confini del mondo, detto "Muro"

Figura 12: Belief Boccino

INDIVIDUAZIONE DELLO STATO	DELIBERAZIONE	INTENTIONS [PIANI → INSIEME DI AZIONI]
• RAGGIUNTO CONFINE DEL MONDO	• ATTIVAZIONE PASSAPORTA	• Passaporta-boccino
• CERCATORE NELLE VICINANZE	• EVITARE LA CATTURA	• Avoid-catching
• NESSUN CERCATORE O MURO NELLE VICINANZE	• MOVIMENTO CASUALE	• Simple-move-boccino

Figura 13: Intention e Plan Action Boccino

3.3.6 I Bolidi

Il Bolide è un'entità attiva interna all'ambiente di gioco che ha il compito di disturbare i giocatori. Essendo un'entità attiva il Bolide è consapevole della sua posizione e può osservare la posizione degli altri giocatori.

Name	Description
Proprietà Strutturali	
starting-x, starting-y	- Il valore della coordinata della posizione iniziale, espressa come numero intero
Starting-heading	- Il valore iniziale di orientamento del giocatore
Speed	- Velocità del giocatore, questa varia in base allo slider presente nell'interfaccia utente "basic-speed" e si modifica in base all'azione selezionata dal cacciatore.
Angle	- Direzione che il giocatore prende durante la partita per muoversi
Settaggi di gioco	
Attack-range	- È la visuale del bolide. Ogni cacciatore infatti ha una conoscenza parziale dell'ambiente. Anche questo valore è modificabile tramite lo slider "visual-range" presente nell'interfaccia grafica
Stato Interno	
Attack?	- Booleano, inizialmente settato a true. Esso indica lo stato del bolide, cioè se è in fase di attacco o meno
has-target?	- Booleano, inizialmente settato a false. Indica se il bolide ha individuato un target da attaccare
Target	- Il target selezionato che il bolide vuole colpire.
Conoscenza dell'ambiente	
chasers-near, keepers-near, seekers-near	- Conteggio del numero di giocatori visibili al bolide

Figura 14: Belief Bolide

INDIVIDUAZIONE DELLO STATO	DELIBERAZIONE	INTENTIONS [PIANI → INSIEME DI AZIONI]
• IL BOLIDE DEVE ATTACCARE MA NON HA UN OBIETTIVO	• SELEZIONE DEL TARGET DA ATTACCARE	• Select-target • Attack-target
• OBIETTIVO DA COLPIRE SELEZIONATO	• COLPIRE IL TARGET	• Attack-target
• TARGET / OBIETTIVO COLPITO	• SELEZIONARE UN NUOVO TARGET: alternando un target patch casuale, ad una entità per evitare che venga selezionato immediatamente la stessa entità	• decide-type-target • wait-for-select

Figura 15: Intentions e Plan Action Bolide

3.4 Entità Passive

Le entità passive non hanno autonomia, e quindi non hanno il concetto di architettura BDI. Queste entità vengono modificate da agenti esterni, che possono essere l'Observer o le altre entità attive del gioco

3.4.1 La Pluffa

La Pluffa è una palla contesa dalle entità attive "Cacciatori", i quali hanno lo scopo di condurla, attraverso movimenti, passaggi, e tiri, nell'area goal avversaria, per poter ottenere i punti relativi a un goal.

- il movimento della Pluffa è attivato dall'Observer che ne modifica la velocità e l'orientamento.
- l'angolo e la velocità della Pluffa vengono modificati dalle entità attive del gioco, in particolare dai Cacciatori e dai Portieri, tramite le azioni di passaggio, movimento, tiro, e parata.

4 Implementazione

4.1 NetLogo

4.1.1 Che cos è NetLogo?

L'obiettivo di questa sezione non è quello di fornire una documentazione dettagliata sulla sintassi e le procedure di NetLogo. Offriremo invece una panoramica generale su questo potente strumento di modellazione. NetLogo nasce nel 1999

da Uri Wilensky, un professore di Computer Science e Sistemi Complessi alla NorthWestern University, ed è l'ultimo di una serie di linguaggi per la modellazione multi-agente che includono StarLogo e StarLogoT. Distribuito sotto licenza GPL, questo software opensource è stato scritto principalmente in Scala, con alcune parti in Java, ed è eseguibile sulle principali piattaforme (Mac, Windows, e Linux). NetLogo è un ambiente di sviluppo nato per simulare fenomeni naturali e sociali. Gli sviluppatori possono dare istruzioni a centinaia di migliaia di agenti, tutti operanti in maniera indipendente: questo permette di esplorare la connessione tra il comportamento emerso dalle singole entità del micro-livello e i pattern risultanti dalla loro interazione (macro-livello). Inoltre fornisce un'ampia documentazione e una vasta gamma di modelli già pronti all'uso, così da poter essere studiati anche da neofiti della simulazione. Tra le aree ricoperte dai modelli già pronti troviamo le scienze sociali e naturali, la medicina, la biologia, la fisica, l'informatica e la psicologia sociale.

4.1.2 Agenti in NetLogo

In NetLogo ci sono tre tipi di agenti:

- **TURTLES** Le *turtles* sono agenti che si muovono nel mondo. Il mondo è bidimensionale ed è diviso in una griglia di *patches* e ogni turtle ha delle coordinate rispetto alle patches (`pxcor`, `pycor`) che assumono valori interi , o proprie che assumono valori decimali (`xcor`, `ycor`) .
- **OBSERVER** L'observer non osserva passivamente, può dare istruzioni alle turtles. Non ha una propria locazione, ed è colui che osserva e controlla il mondo dall'esterno e l'evoluzione del modello.
- **PATCHES** Ciascuna Patch è un rettangolo quadrato che compone la superficie sul quale le turtles possono muoversi. Le patch hanno delle coordinate `pxcor` e `pycor`.

4.1.3 L'interfaccia grafica

Oltre a fornire una serie di procedure per simulare un sistema multi-agente, NetLogo fornisce anche un supporto per la creazione di un'interfaccia grafica ad hoc per il sistema che si sta modellando. Grazie ad una toolbar è possibile con pochi click aggiungere bottoni, slider, switch, monitorare variabili in tempo reale e plottare grafici. In questo modo si agevola notevolmente il processo di analisi del modello.

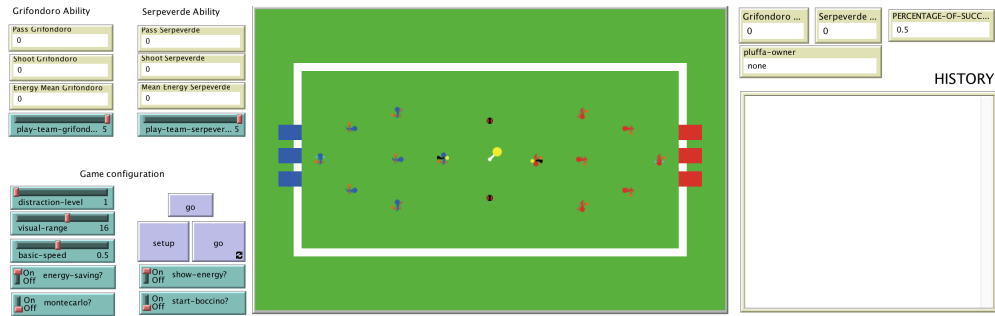


Figura 16: L'interfaccia di gioco

4.1.4 Sviluppo della libreria BDI.nls

Prima di addentrarci nella spiegazione delle scelte implementative che riguardano lo sviluppo dei comportamenti degli agenti è doveroso soffermarci un attimo a parlare della libreria sviluppata e successivamente utilizzata da noi per implementare l'architettura BDI delle entità. Infatti NetLogo, per quanto estremamente utilizzato per la simulazione ad agenti, non possiede una sua libreria ad hoc per modellare gli agenti secondo l'architettura BDI. In rete abbiamo rinvenuto qualche esempio di librerie sviluppate da utenti singoli, poco commentate e documentate. Per questi motivi abbiamo preferito implementare una semplice libreria creata *ad hoc* per le nostre esigenze. La libreria `BDI.nls` implementa il concetto di belief attraverso una mappa di coppie (chiave, valore), memorizzata in una tabella. Ogni entità possiede la propria tabella dei belief che viene creata al momento del setup del gioco. Abbiamo provveduto a fornire funzioni per aggiungere, rimuovere, aggiornare e persino stampare i belief di ciascun entità, gestendo anche gli eventuali errori.

4.2 Scelte implementative riguardo allo svolgimento della partita

La partita di svolge a turni, scanditi dai tick.

Ad ogni tick gli agenti osservano il mondo e determinano l'azione da eseguire per raggiungere l'obiettivo comune di fare goal e di proteggere la propria area. Ad ogni tick vengono inoltre aggiornati i parametri riguardanti al possesso palla, l'energia dei Cacciatori e i relativi grafici.

In caso di goal, di parata oppure di palla uscita dal campo, l'Observer ha il compito di far sì che le due squadre si riposizionino secondo la formazione iniziale e che la Pluffa sia disposta al centro del campo. Durante questa operazione tutte gli agenti, ad eccezione del Boccino d'oro e dei Cercatori, attendono due tick di pausa. Questa scelta è stata fatta per evidenziare il fatto che il comportamento degli attori Cercatori, dei Bolide e del Boccino è completamente legato dalle vicende relative alla Pluffa e ai goal segnati in campo. Le azioni che coinvolgono il bolide, e la cattura del boccino, prescindono dalle azioni relative alla Pluffa.

Per rendere ancora più interessante lo studio del modello abbiamo permesso all'utente di interagire con il modello tramite alcuni pulsanti posti nel menu di configurazione di gioco.

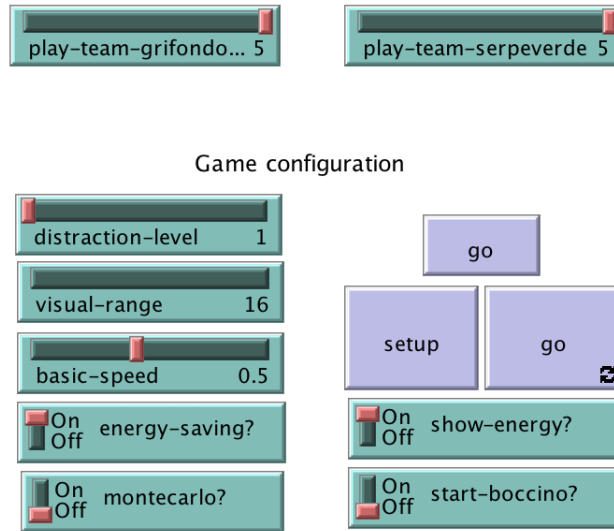


Figura 17: Menu di configurazione

- La prima cosa da notare è la presenza di due pulsanti "go" nell'interfaccia: tramite essi è possibile scegliere se eseguire un singolo turno di gioco senza proseguire con il turno successivo, utile per il debug, oppure se optare per un'esecuzione sequenziale di più turni di gioco, per osservare quindi l'esecuzione del modello.
- Esiste inoltre un pulsante di *switch* **show-boccino?** che permette all'utente di stabilire quando rendere visibile il Boccino d'Oro. La scelta di implementare questa opzione è derivata dal fatto che la cattura del Boccino d'Oro da parte dei Cercatori comporta la fine immediata della partita; rendere il Boccino sempre visibile e catturabile avrebbe compromesso lo studio accurato dei comportamenti degli altri attori, in quanto la loro azione sarebbe potuta terminare da un momento all'altro, anche nei primi turni di gioco.
- Per un migliore monitoraggio del sistema in tempo reale è stato inserito uno *switch* **show-energy?**, che permette di visualizzare una label collegata alle icone dei Cacciatori con l'energia che hanno a disposizione; Questo permette di analizzare le scelte comportamentali prese con determinate situazioni ambientali e con determinati livelli di energia. In condizioni normali, cioè con lo switch **show-energy?** disattivato, viene visualizzato l'id della *turtle* corrispondente.

- Esistono altri due *switch*, **greedy?** e **montecarlo?**, di cui parleremo più approfonditamente nelle prossime sezioni.
- A disposizione dell'utente finale sono stati predisposti anche alcuni *slider* che permettono di influenzare il comportamento degli agenti:
 - **visual-range** è lo *slider* che permette di modificare il campo visivo dei giocatori (Cercatori, Cacciatori, Portieri) . Esso inoltre corrisponde all'area entro cui il bolide seleziona i suoi obiettivi da attaccare, e l'area in cui il boccino individua i Cercatori da cui scappare.
 - **basic-speed** è lo *slider* che permette di modificare la velocità minima dei giocatori. L'utente può aumentare o diminuire per un intervallo compreso da 1 a 9.
 - **team-play** è lo *slider* che condiziona il comportamento dei Cacciatori nei termini di tendenza al gioco di squadra. Ne parleremo in modo più approfondito nella sezione dedicata ad essi.
 - **distraction-level** è lo *slider* che condiziona la "concentrazione" degli agenti in merito al possesso della Pluffa. Ne approfondiremo l'utilizzo e l'utilità nelle sezioni successive.

La scelta di cambiare i parametri può essere effettuata dall'utente in qualsiasi momento della partita, senza alcun vincolo. L'aggiornamento sarà propagato ai belief dei giocatori nel turno (tick) successivo . Infine, accanto alla finestra di gioco, è stato predisposto un monitor di output che simula la telecronaca di quello che succede realmente nella partita, delle azioni ritenute più importanti.

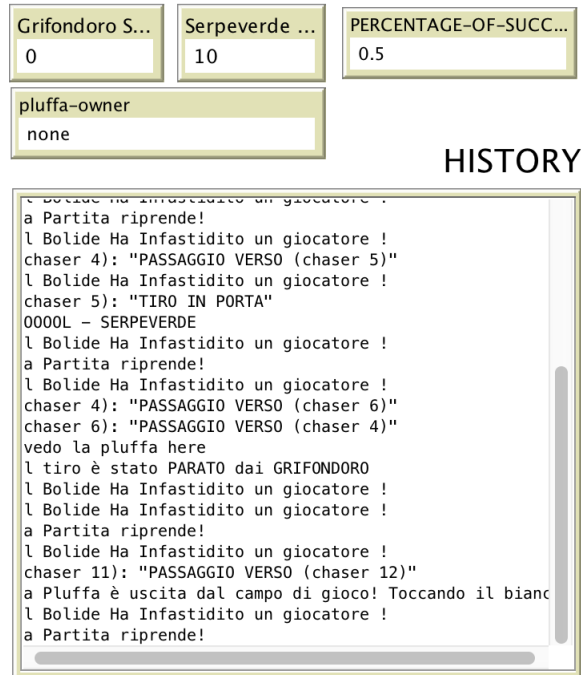


Figura 18: Telecronaca della partita e monitor segnapunti

4.3 Il comportamento delle entità

Seguendo il modello suggerito dal framework BDI abbiamo suddiviso il comportamento dei singoli agenti in due fasi:

- **observe-update-belief**, fase in cui l'entità osserva il mondo circostante aggiornando i beliefs;
- **deliberate-planning-executing** nella quale l'entità, analizzando i dati raccolti dal mondo nella fase precedente, sceglie il piano di azione a seconda del modello definito, e lo esegue.

4.3.1 Il comportamento dei Portieri

Il comportamento del portiere è stato semplificato, in quanto non può effettuare passaggi al seguito di una parata, e non sono previsti retro-passaggi dai giocatori dello stesso team. Inoltre il suo movimento è limitato all'asse y, vincolandolo in quanto privato della possibilità di uscire dalla porta, ricordando in questo modo il comportamento dei portieri del Calcio Balilla.

L'insieme delle azioni che un Portiere può eseguire è composto da:

- **waiting-keepers** E' l'azione in cui il portiere è semplicemente fermo, in attesa di poter aggiornare nuovamente il mondo al turno successivo. Infatti, non vedendo la Pluffa non può far altro che fermarsi e aspettare.
- **follow-pluffa** E' l'azione in cui il Portiere, una volta individuata la Pluffa cerca di seguirla spostandosi in direzione verticale verso di lei.
- **stop-pluffa** E' l'azione che viene compiuta dal Portiere se e solo se ha la possibilità di parare la palla. La percentuale di riuscita della parata è determinata dal valore della variabile **PERCENTAGE-OF-SUCCESS**, la quale è influenzata da diversi fattori quali: la presenza di Bolidi e Cacciatori avversari nella visuale del Portiere e dallo *slider distraction-level*.

A differenza dei Cacciatori, il valore dello *slider distraction-level* influenza la probabilità di successo della parata, **PERCENTAGE-OF-SUCCESS**, in quanto nel mondo reale la distrazione del portiere comporta una ridotta reattività nel momento dell'arrivo della pluffa.

4.3.2 Il comportamento dei Cercatori

Il comportamento dei Cercatori è assimilabile a quello della fototassi delle mosche: se avvistano il Boccino d'Oro lo inseguono e cercano di catturarlo per primi. Le azioni che i Cercatori possono eseguire sono:

- **exploration** è l'azione base, che viene eseguita ogniqualvolta il Boccino d'Oro non è nel raggio di visivo dei Cercatori.
- **follow-boccino** questa azione viene eseguita quando il Boccino d'Oro viene avvistato. Il Cercatore prova ad inseguirlo.
- **passaporta-move** questa azione è legata alla scelta implementativa di rappresentare i confini del mondo come un muro magico, una passaporta, che teletrasporta gli attori che si ritrovano ai margini del campo di gioco in una posizione casuale all'interno della stessa area.

Per esplorare il comportamento dei Cercatori, oltre al già citato *switch show-boccino?*, l'utente può intervenire modificando gli *slider visual-range* e *basic-speed*. La velocità è ulteriormente influenzata dalla presenza di Bolidi: se un Cercatore viene colpito da un Bolide rallenta la sua velocità. Per non complicare ulteriormente la cattura del Boccino d'Oro, abbiamo preferito non implementare in queste entità il concetto di energia e di concentrazione, in quanto a nostro avviso irrilevanti ai fini dello studio del comportamento.

4.3.3 Il comportamento del Boccino

Il comportamento del Boccino è diametralmente opposto a quello dei Cercatori:

- **simple-move-boccino** E' l'azione in base alla quale il Boccino si muove randomicamente nello spazio;

- **avoid-catching** E' l'azione in base alla quale il Boccino, dopo aver avvistato uno o più Cercatori nel paraggi, prova a scappare innescando un comportamento repulsivo verso di essi, aumentando la velocità e modificando immediatamente la direzione (in maniera randomica).
- **passaporta-boccino** E' l'analoga azione della **passaporta-move** dei Cercatori.

Il comportamento del Boccino d'Oro non è influenzato in alcun modo dalla presenza del Bolide, essendo entrambe due palle del gioco.

4.3.4 Il comportamento dei Bolidi

Il comportamento dei Bolidi è essenzialmente un comportamento di fototassi, con l'obiettivo di disturbare gli altri agenti. Come abbiamo visto nelle precedenti sezioni, la presenza del Bolide condiziona la probabilità di riuscita di un'azione di un agente. La scelta della vittima da attaccare è determinata da questa scala di priorità:

- I Cercatori, In quanto essi sono gli unici possibili artefici della terminazione della partita in corso; come già detto la presenza di un Bolide nella stessa patch del Cercatore conferisce a quest'ultimo un malus sulla velocità, rallentandolo e quindi ostacolando nel suo obiettivo.
- I Portieri; quando un Bolide attacca un portiere, ha la conseguenza di ostacolare la visuale, incidendo pesantemente sulla riuscita della parata.
- I Cacciatori: se un Bolide condivide la patch con un Cacciatore non solo ne condiziona la probabilità di perdere la Pluffa, ma ne diminuisce l'energia influenzando in ogni caso le azioni di quest'ultimo.

Le azioni che un Bolide può compiere sono le seguenti:

- **select-target** Il Bolide sceglie l'obiettivo da attaccare secondo i criteri elencati sopra. Nel caso in cui nel raggio visivo d'attacco non sia presente alcun tipo di giocatore, allora esso selezionerà una zona casuale del campo da raggiungere e in cui ripetere la selezione del target.
- **attack-target** Rappresenta l'azione per cui il Bolide attacca il bersaglio selezionato in precedenza. Questa azione è eseguita a fronte di qualsiasi tipo di target selezionato.
- **decide-type-target** Azione eseguita successivamente al raggiungimento di un precedente target. Infatti, dopo aver colpito il target, nel successivo turno il bolide valuterà la tipologia di target da selezionare. Le tipologie di target sono due: Player e Patches. Queste due tipologie si alternano in modo che il bolide cambi zona, così da avere una buona probabilità di selezionare un diverso player, senza accanimento, per cercare di mantenere un approccio di fairness.

- **wait-for-select** Azione in cui il bolide rimane fermo, per simulare il momento di selezione del prossimo target da attaccare.

Come abbiamo detto all'inizio, per evitare ulteriori complessità del sistema abbiamo scelto di non implementare i Battitori, i quali sarebbero stati i principali agenti a relazionarsi con i Bolidi.

4.3.5 Il comportamento dei Cacciatori

I Cacciatori sono sicuramente gli agenti più complessi modellati dal nostro sistema.

Il concetto di energia A differenza di tutte le altre entità viste in precedenza, i Cacciatori incapsulano il concetto di *energia* che influenza drasticamente il loro comportamento. Analogamente al mondo reale infatti, ogni azione scelta ha un suo costo in termini di consumo energetico: l'uso eccessivo di azioni dispendiose porta ad un rapido azzeramento dell'energia a disposizione, compromettendo l'interazione con il mondo esterno e la possibilità di scegliere l'azione più consona al momento. Infatti, ogniqualevolta l'energia di un Cacciatore si esaurisce, egli è costretto a fermarsi almeno per un turno, durante il quale non può fare altro che osservare i cambiamenti del mondo, e ricaricarsi di energia, senza però poter intervenire attivamente nel gioco. La durata del periodo durante il quale il giocatore è costretto a fermarsi è scandita dalla variabile contenuta nel belief **"time-for-recovery"**.

Risparmio energetico Come abbiamo spiegato nel paragrafo precedente l'energia è un punto nodale del comportamento dei Cacciatori. Per rendere ancora più evidente la differenza tra un comportamento parsimonioso e un comportamento dispendioso è stato messo a disposizione dell'utente uno slider **"energy-saving?"**, che permette di selezionare una delle due strategie di approccio al gioco. Nel caso in cui questo slider sia attivo, i Cercatori cercheranno di mantenere un comportamento il più possibile parsimonioso, al fine di conservare le energie per le situazioni decisive. L'attivazione del comportamento **"energy-saving?"** incide in particolar modo nella decisione sulla velocità da utilizzare.

La gestione della velocità Abbiamo già illustrato nel modello che tutte le entità attive possiedono una velocità di base che può essere modificata dall'utente tramite lo slider **basic-speed**. Tuttavia, proprio come nella vita reale, è possibile modificare la velocità in base al mondo osservato e all'energia a disposizione. Infatti, accelerare la propria andatura comporta un dispendio di energie pari allo sforzo richiesto e, naturalmente, lo sforzo usato non può superare l'energia che il giocatore ha a disposizione. Di conseguenza, ogni volta che si rivela necessario decidere la velocità del movimento del giocatore, sarà prima di tutto calcolato uno **speed-up**. La scelta del valore dello **speed-up** è

randomica, secondo una probabilità condizionata dall'energia che il Cacciatore possiede in quel determinato momento. Dopo che il cacciatore ha stabilito in questo modo quanta energia investire nella velocità, si può procedere con la scelta della velocità da seguire. Sono presenti tre gradi di velocità.

- **default** corrisponde alla velocità base di un giocatore, e non richiede alcuno sforzo aggiuntivo.
- **walking-speed** è la velocità ottenuta sottraendo lo **speed-up** ottenuto alla velocità base dei Cacciatori. E' stata implementata per simulare un'andatura paragonabile a quella della camminata del giocatore. Camminando il Cercatore si muove più lentamente rispetto alla sua **basic-speed**, e nel frattempo può recuperare energia, in base allo **speed-up** scelto.
- **scatto-speed** è la velocità ottenuta aggiungendo lo **speed-up** calcolato alla **basic-speed**. L'obiettivo, in questo caso, è simulare lo scatto del giocatore, che accelera aumentando la velocità, a discapito di un dispendio di energia superiore alle altre andature.

In linea di massima quando lo slider "**energy-saving?**" è attivo, i Cacciatori tenderanno a privilegiare la camminata. Esistono anche delle situazioni che prevedono l'utilizzo della semplice **basic-speed**, senza alcun incremento o decremento della velocità stessa.

La gestione del contrasto Un altro punto rilevante del sistema è la gestione dei contrasti tra giocatori. Per mantenere un certo grado di semplicità abbiamo scelto di affidare la scelta del risultato del contrasto ad un'entità esterna: l'Observer. Ad ogni turno i Cacciatori osservano l'ambiente circostante ed esprimono eventualmente la consapevolezza di avere il possesso della Pluffa tramite la modifica del belief "**i-have-pluffa**", con un ragionamento analogo al seguente: "se la Pluffa è immediatamente vicino a me, modifico il belief con il valore 1, in tutti gli altri casi il belief assume valore 0". Dopo che tutti i Cacciatori hanno osservato il mondo entra in gioco l'Observer: questa entità controlla se ci sono più giocatori che hanno rilevato la Pluffa nella loro stessa posizione, e in caso affermativo si occupa di decidere equamente il vincitore dello scontro. In questo modo siamo in grado di impostare probabilità coerenti. I Cacciatori che perdono lo scontro guadagnano tramite l'Observer la consapevolezza di non avere più il controllo della Pluffa e, per permettere al vincitore di divincolarsi agevolmente, dato che lo scontro richiede un certo sforzo, perdono tutta l'energia a disposizione. Il vincitore invece guadagna un bonus di energia che gli permette di eseguire almeno un'azione.

Il possesso della Pluffa è un vincolo essenziale nel determinare l'azione da compiere: risulterebbe irragionevole e surreale che un giocatore senza la Pluffa possa scegliere come azione un passaggio o un tiro, i quali per definizione richiedono una stretta interazione con essa. Per illustrare meglio le azioni che possono essere scelte da un Cacciatore, le suddivideremo in due grandi macro-gruppi: le azioni con il possesso della Pluffa e le azioni senza la Pluffa.

Le azioni senza la Pluffa

- **simple-move** E' l'azione che indica il movimento del Cercatore che non possiede la Pluffa. Velocità e direzione di spostamento sono caratterizzate dalle condizioni che vengono osservate nella percezione del mondo. Difatti, gli stati del mondo che possono richiedere l'esecuzione di questa azione sono:
 - *STATO: Il Cacciatore vede la Pluffa e il possesso della Pluffa è di un suo compagno di squadra.*
DELIBERAZIONE: Il Cacciatore a questo punto valuta se il compagno ha fisicamente la Pluffa nella sua stessa patch. In caso affermativo, dato che il compagno con la pluffa cercherà di avvicinarsi alla porta avversaria, il player avrà l'*INTENZIONE* di aiutare il compagno a fare goal, e si muoverà in attacco anch'esso. In caso contrario, è molto probabile che sia stato effettuato un passaggio, o un tiro, quindi il Cacciatore assume come *INTENZIONE* quello di recuperare la pluffa "a rischio" e si muove verso la Pluffa con la volontà di intercettarla.
 - *STATO: Il Cacciatore vede la Pluffa e il possesso della Pluffa è della squadra avversaria.*
DELIBERAZIONE: Il Cacciatore guarderà se è lui il più vicino alla Pluffa, adottando l'*INTENZIONE* di recuperarla, e quindi muovendosi in quella direzione. Questo si verifica anche se il cacciatore ha una distanza minima dal più vicino compagno di squadra alla pluffa, o se l'azione avviene in prossimità dell'area goal. Altrimenti, assumerà l'*INTENZIONE* di allontanarsi dall'azione, cercando di tornare alla sua posizione iniziale, per non intralciare i compagni di squadra e per supportarli non appena la pluffa sia recuperata. Questa scelta implementativa è stata fatta per evitare ammassamenti di Cercatori nella stessa patch, cosa che nel mondo reale avviene con estrema rarità.
 - *STATO: Il Cacciatore non vede la Pluffa.*
DELIBERAZIONE: Il Cacciatore semplicemente assumerà l'*INTENZIONE* di cercare la Pluffa. Non subisce alcun tipo di accelerazione in quanto sarebbe uno spreco di energie.
- **stop-me** Riguardo a questa azione ne abbiamo già discusso indirettamente in precedenza. Solo fermandosi il Cacciatore è in grado di recuperare parzialmente l'energia che ha consumato nei turni precedenti.
STATO: Energia Esaurita , Oppure Turno di riposo
DELIBERAZIONE: Il player decide di riposarsi con l'*INTENZIONE* di fermarsi, e quindi di non effettuare alcun movimento.

Le azioni con la Pluffa *STATO:* Le seguenti azioni vengono eseguite a nel caso in cui sia rilevato uno stato del mondo in cui il cacciatore in questione

possiede la pluffa.

DELIBERAZIONE: Nel momento in cui il player possiede la pluffa dovrà deliberare per scegliere l'*INTENZIONE* che possa essere la migliore da intraprendere, per massimizzare il risultato ottenuto, per realizzare il macro obiettivo di fare goal. Infatti ognuna di queste azioni tende a traslare il gioco di una certa squadra verso l'area avversaria.

Le azioni che possono essere intraprese sono:

- **possession-move** Rappresenta il movimento del cacciatore, e della pluffa al suo seguito. In base alla velocità da utilizzare scelta si suddivide in:
 - **possession-move "walking"**
 - **possession-move "run"**
 - **possession-move "normal"**

Ogni azione di movimento è intrinsecamente connessa alla decisione della direzione da prendere. Sebbene la direzione ultima da intraprendere è sicuramente la porta avversaria, i percorsi per raggiungerla sono molteplici, tenendo in considerazione il fatto che incontrare altri avversari sul proprio cammino incide sulla probabilità di perdere il possesso della Pluffa. Per questo motivo è stato introdotto il concetto di "**foot**" ovvero il piede preferito del giocatore. Ogni Cercatore, ad ogni movimento ha il 98 per cento di probabilità di utilizzare il suo piede preferito. Tuttavia nello specifico caso dello scatto non si tiene conto del piede preferito in quanto considerata una scelta immediata. Tutto questo è racchiuso nella funzione **decide-direction-move**, chiamata al termine di ogni **possession-move**.

- **pass** E' l'azione che corrisponde al passaggio della Pluffa. Non è stato implementato nessun meccanismo esplicito che permetta ad un Cacciatore di riconoscere il passaggio del compagno, e neppure quello di richiedere il passaggio della Pluffa. Questo per mantenere il sistema il più semplice possibile. Tuttavia i compagni di squadra assumono l'intenzione di recuperare la pluffa non appena vedono che nessun compagno di squadra la possiede effettivamente.
- **shoot** E' l'azione che corrisponde al tiro in porta.
- **stop-me** Anche in possesso di Pluffa i giocatori possono decidere di fermarsi.

A differenza delle azioni senza la Pluffa, le quali sono quasi scelte obbligate per ottenere un comportamento ragionevole e accettabile, le azioni con la Pluffa vengono deliberate seguendo un algoritmo di decisione più complesso.

L'algoritmo decisionale Arriviamo quindi alla parte più interessante del comportamento dei Cacciatori, ovvero come avviene la deliberazione della scelta di un'azione con la Pluffa. Abbiamo individuato due approcci decisionali differenti:

- *Approccio casuale*, diviso a sua volta in:
 - con *Selezione Montecarlo*
 - con l'utilizzo del generatore di *pseudo-random number* pre-esistente in Netlogo

L'utente può selezionare l'approccio casuale che desidera attraverso lo *switch montecarlo?*.

Per un confronto diretto tra i due approcci decisionali, indipendentemente dall'attivazione dello slider, sono stati inseriti due grafici che plottano i risultati delle due computazioni.

- Approccio basato sulla *scelta della migliore e utile azione* tra tutte quelle possibili in quel turno. Per poter implementare questo approccio si è reso necessario creare delle funzioni attraverso le quali i Cacciatori potessero assegnare ad ogni azione un punteggio, *score*, in base ad alcune percezioni raccolte nella fase di osservazione del mondo.

I fattori che influenzano gli *score* sono:

- *Il numero di compagni presenti nelle immediate vicinanze* `get-belief "n-teammates-patch"`: La presenza di più compagni di squadra affianco scoraggia il passaggio, e di conseguenza influenza positivamente lo *score* relativo al movimento del cacciatore con la pluffa.
- *Il numero di avversari presenti nelle immediate vicinanze* `get-belief "n-opponents-patch"`: La presenza di più compagni di squadra affianco influenza positivamente il passaggio, e di conseguenza influenza negativamente lo *score* relativo al movimento del cacciatore con la pluffa.
- *La distanza dal compagno più vicino* `distance get-belief "nearest-teammate"`: Questo influenza negativamente il passaggio, essendoci un compagno troppo vicino, che potrebbe intercettare un qualsiasi passaggio.
- *Il numero di compagni presenti in un certo raggio visivo* `get-belief "n-near-teammates-range"`: Se il numero di compagni è maggiore del numero di avversari attorno a un cacciatore, questo avrà meno volontà di effettuare un passaggio. Inoltre, se c'è almeno un compagno intorno al cacciatore, esso avrà una leggera positività nei confronti del movimento, avvertendo l'appoggio della squadra.
- *La presenza dell'area goal avversaria vicina* `has-belief "goal-near"`: Quando il cacciatore si avvicina all'area in cui ha la possibilità di fare goal, sarà meno volenteroso di effettuare un passaggio, in quanto aumenterà il desiderio di effettuare un tiro. Con il crescere della vicinanza alla porta, inoltre aumenterà in modo proporzionale lo *score* relativo al tiro.
- *Il livello del gioco di squadra* `has-belief "play-team"`: Parametro influenzato dallo slider citato nelle precedenti sezioni. Questo valore

incrementa il punteggio relativo al passaggio, se possibile, e invece scoraggia le azioni individuali, e quindi il movimento.

- *Il livello di energia* `get-belief "energy"`: Alcune azioni come il passaggio e il tiro richiedono più energia rispetto ad una semplice corsa con velocità di base. Per questo influisce nella scelta dell'azione da intraprendere.
- *Il posizionamento di un player sulla linea bianca, confine del campo di gioco* Questo fattore ha una altissima influenza sulla decisione del giocatore di fermarsi con la Pluffa, e di non effettuare alcuna azione.

Gli *score* assegnati a ogni azione, sono rappresentativi della maggiore probabilità di essere utili per la squadra, per raggiungere l'obiettivo globale di fare goal. Il giocatore potrà scegliere quindi l'azione da intraprendere che ha maggior punteggio, seguendo quindi un approccio cosiddetto Greedy, o una qualsiasi senza badare all'utilità che si otterrà dall'esecuzione dell'azione scelta.

Il distraction-level La scelta tra i due approcci è determinata dal valore contenuto nello slider `distraction-level`, anch'esso modificabile a runtime dall'utente. Questo slider indica il grado di distrazione del giocatore: più il valore è basso più il Cacciatore sarà concentrato sulla scelta e avrà meno probabilità di effettuare una scelta randomica; al contrario, più è alto, più aumentano le probabilità di incappare in una scelta randomica e potenzialmente meno ragionevole/meno adatta al contesto.

5 Analisi del sistema simulato e conclusioni

5.1 Analisi del comportamento emerso

- In primo luogo abbiamo constatato che, nonostante non sia stato implementato esplicitamente, i Cacciatori si comportano come se avessero l'obiettivo di segnare più punti possibili.
- Altro aspetto emergente è la presenza di errori commessi dai giocatori nell'eseguire le intenzioni deliberate.

Un esempio lo possiamo vedere con passaggi terminanti fuori dall'area di gioco. Questo si verifica principalmente perché il Cacciatore che ha il possesso della palla è a conoscenza della posizione dei suoi compagni al momento della deliberazione dell'azione, ma non tiene conto della velocità e della direzione dei medesimi e che questi nel frattempo potrebbero dirigersi altrove. Inoltre, esiste anche la possibilità che i passaggi possano essere intercettati dai Cacciatori avversari, anche se la decisione di effettuare un passaggio, con determinate configurazioni, cerca di evitare le situazioni critiche con alta probabilità di errore/intercettazione.

- L'ennesimo aspetto emergente è il comportamento di aggirare i Cacciatori avversari simulando un rudimentale meccanismo di "dribbling".

5.2 Comparazione teamplay

Il gioco di squadra non è un concetto che il Cacciatore conosce: il valore dello slider **play-team** corrisponde semplicemente all'aumento o alla diminuzione della probabilità che il giocatore decida di passare la palla (per questo si chiama gioco di squadra) , andando ad influenzare il punteggio ottenuto nella `texttpass-score`, come mostrato nelle figure seguenti. Si prenderanno in considerazione degli esempi con livello di distrazione minimo, per poter analizzare il comportamento ragionato degli agenti, che scelgono le azioni più utili per raggiungere sotto-goal selezionati runtime.

Nella figura 19 si vuole evidenziare il cambiamento riscontrato nella decisione di un cacciatore del team Grifondoro con livello team-play minimo, e successivamente nella figura 20 con livello team-play massimo. Si noti che in entrambe le situazioni, essendo due turni consecutivi, il player che ha la *éluffa* ha un avversario in attacco e un compagno nelle vicinanze. Malgrado questo, nella prima situazione decide comunque di proseguire da solo, mentre nel secondo caso si può notare come la decisione cambi nei confronti del gioco di squadra, in quanto consapevoli del rischio di perdere la *Pluffa*, decide di passarla a uno dei compagni. Nella riquadro rosso nella parte inferiore dello schermo, il command center, sono visualizzati i punteggi assegnati a ciascuna azione da intraprendere-



Figura 19: Decisione with team-play-grifondoro = 1



Figura 20: Decisione with team-play-grifondoro = 5

Una squadra con un **play-team** maggiore sarà più incline a scegliere l'azione passaggio.

5.2.1 Simulazione di partite con variazione del play-team

In questo paragrafo mostreremo tramite esempi di partite simulate quanto influenza il valore dello slider **play-team** in fase di gioco.

Entrambe le squadre con playteam = 1 Come si può notare da entrambe le simulazioni, entrambe le squadre effettuano pochi passaggi e un discreto numero di tiri. Non c'è una differenza rilevante tra numero di passaggi effettuato da ciascuna squadra.

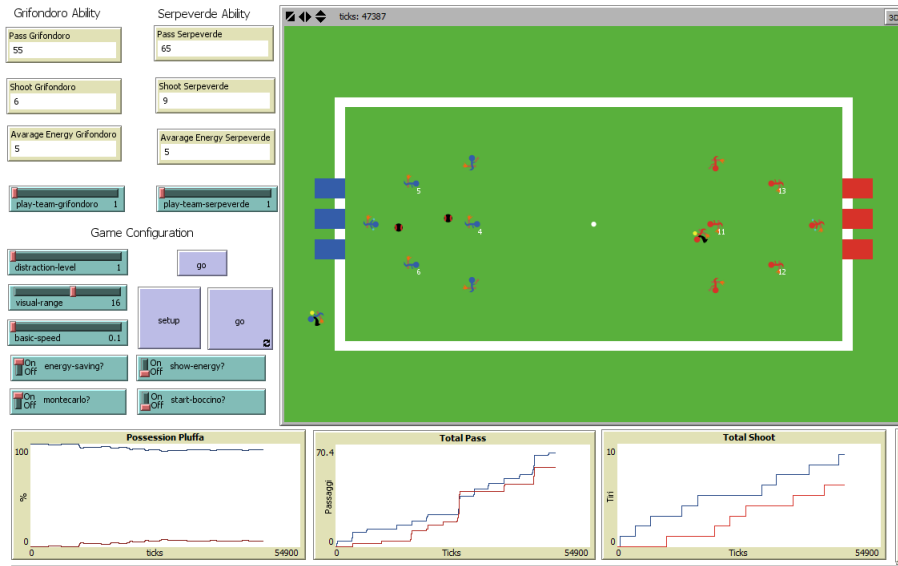


Figura 21: Esempio 1: play-team Grifondoro 1 - play-team Serpeverde 1

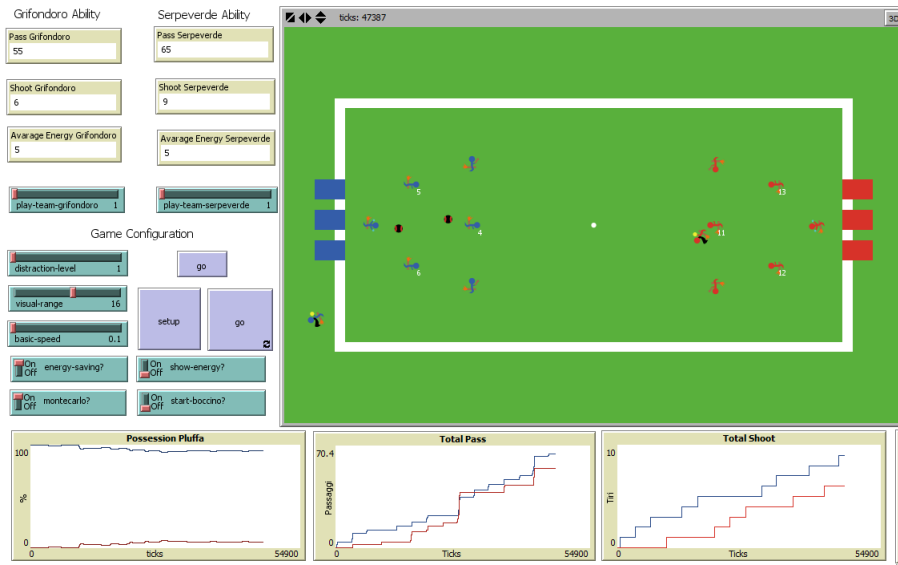


Figura 22: Esempio 2: play-team Grifondoro 1 - play-team Serpeverde 1

Entrambe le squadre con playteam = 5 Nel caso in cui il valore `play-team` sia massimizzato per entrambe le squadre, si nota un incremento sostanziale

delle azioni di passaggio a discapito del numero di tiri effettuati che diminuisce drasticamente.

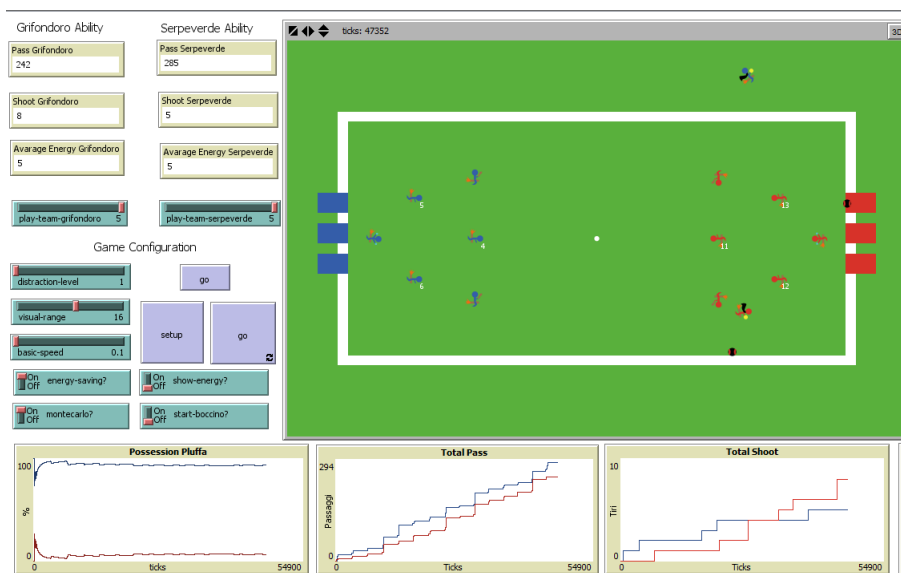


Figura 23: Esempio 1: play-team Grifondoro 5 - play-team Serpeverde 5

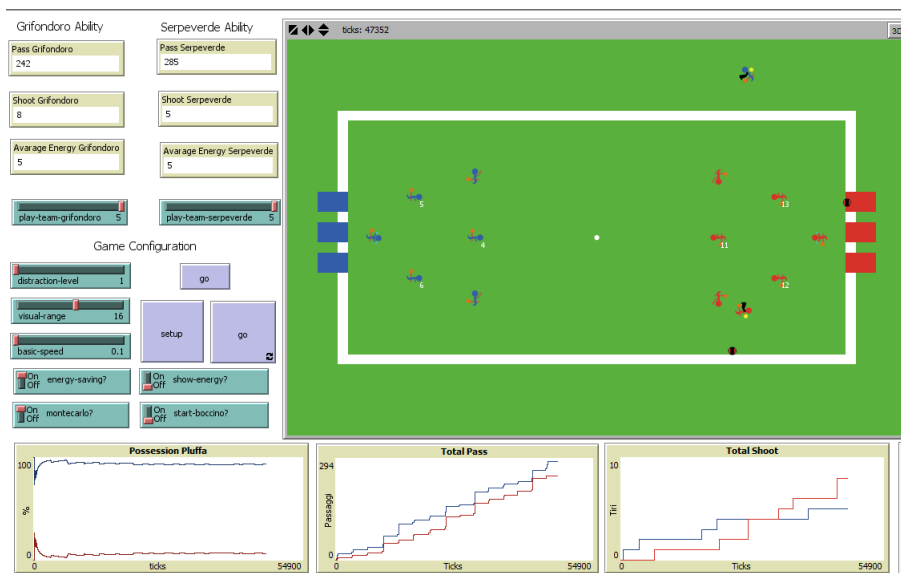


Figura 24: Esempio 2: play-team Grifondoro 5 - play-team Serpeverde 5

Entrambe una squadra con play team massimo e una con playteam minimo Ponendo il massimo dislivello tra il valore del **play-team** della squadra Grifondoro e quello del **play-team** della squadra Serpeverde la differenza di decisione è sostanziale. Come illustrato sia dai monitor che dai grafici di gioco mostrati nelle figure seguenti, i Cacciatori che appartengono alla squadra che ha il valore di **play-team** più elevato tenderanno ad effettuare un elevato numero di passaggi massimizzando il gioco di squadra; al contrario, un Cacciatore appartenente alla squadra con il valore di **play-team** più basso tenderà a massimizzare le azioni individuali, prediligendo azioni di movimento e conclusioni personali.



Figura 25: Esempio 1: play-team Grifondoro 1 - play-team Serpeverde 5

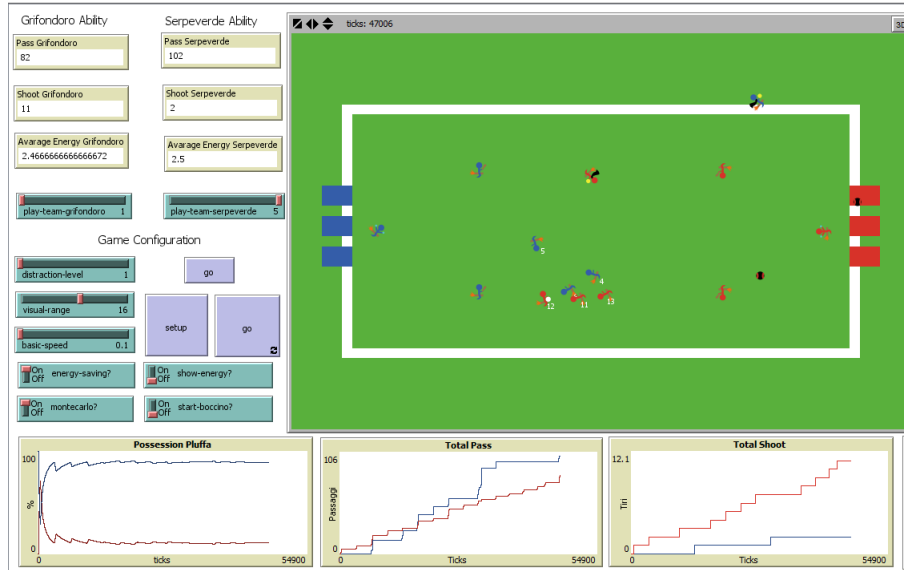


Figura 26: Esempio 2: play-team Grifondoro 1 - play-team Serpeverde 5

5.3 Comparazione livello di distrazione

Anche per quanto riguarda il concetto distrazione, anche in questo caso analizziamo prima il cambiamento riscontrato nella decisione di un cacciatore che possiede la Pluffa. In questo caso, in un primo esempio, figura 21, osserviamo che con un livello di distrazione minimo, il player rimanga concentrato effettuando una azione coerente con la situazione attuale della partita. Dopo soli alcuni tick, cambiando drasticamente il livello di distrazione al massimo, in modo tale da avere una probabilità nulla di ottenere un comportamento greedy, osserviamo come il player selezioni una azione random, in questo caso quella di fermarsi.

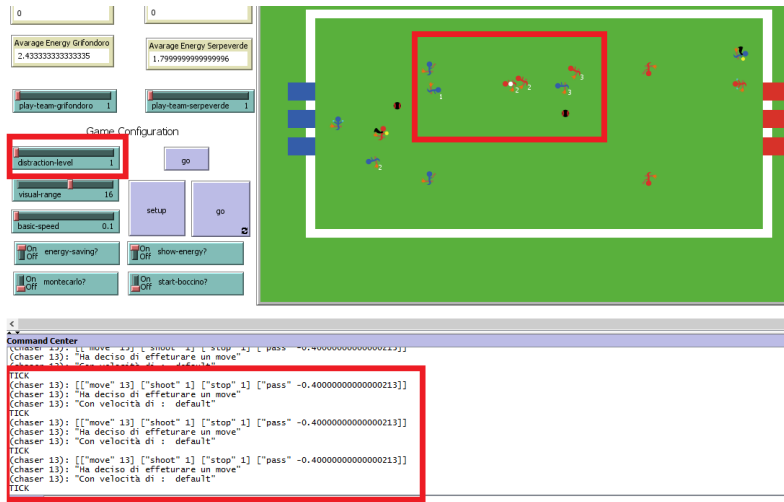


Figura 27: Distraction-level-grifondoro = 1

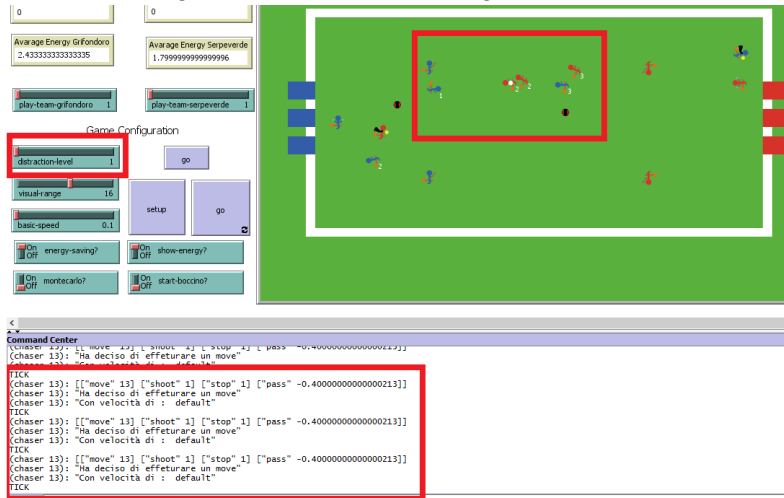


Figura 28: Save Energy non attivo

Figura 29: Distraction-level-grifondoro = 10

Un altro aspetto influenzato dal livello di distrazione, è quello relativo alla scelta della velocità, nella fase di possesso palla di un cacciatore. Infatti, come già spiegato, effettuando scelte ragionevoli, sarà più propenso per un risparmio energetico, anche considerando l'attivazione dello switch energy-saving.

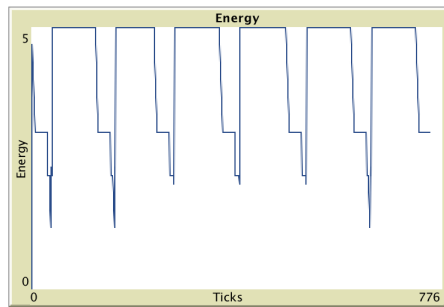


Figura 30: Save energy attivo

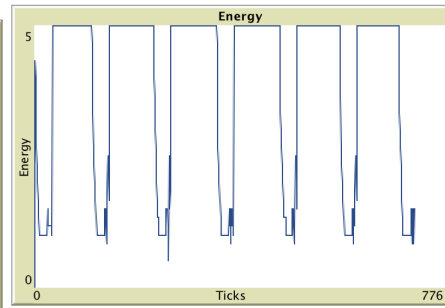


Figura 31: Save Energy non attivo

Figura 32: Risparmio energetico con giocatori concentrati

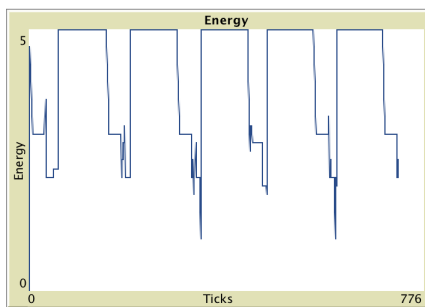


Figura 33: Attivo

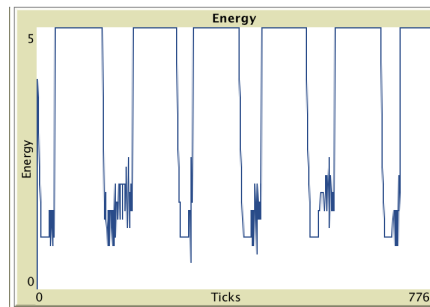


Figura 34: Non Attivo

Figura 35: Risparmio energetico con giocatori con probabilità 0.5 di essere concentrati. Algoritmo di decisione montecarlo

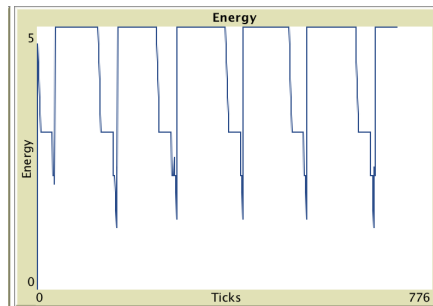


Figura 36: Attivo

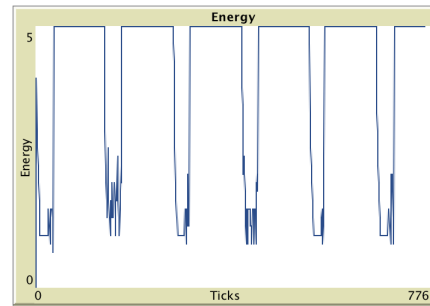


Figura 37: Non Attivo

Figura 38: Risparmio energetico con giocatori con probabilità 0.5 di essere concentrati. Algoritmo di decisione random

Riferimenti bibliografici

<https://ccl.northwestern.edu/netlogo/>

Stewart Russel, Peter Norvig, *"Intelligenza Artificiale, un approccio moderno, volume 2"*, 2a edizione, Pearson - Prentice-Hall, Capitolo 13.1

Slide del corso, in particolare :

Andrea Omicini, *Autonomous Agents as Intentional Systems*, Autonomous Systems, Sistemi Autonomi