

Soccer-SCD
Simulatore di calcio - Progetto di Sistemi
Concorrenti e Distribuiti

SEBASTIANO CATELLANI
SEBASTIANO GOTTARDO
ALESSANDRO SFORZIN

Universita' degli Studi di Padova
June 7, 2014

Contents

Introduzione	3
Scopo del progetto	3
Funzionalità' del software	3
La partita	3
Il software	3
Obiettivi	3
Entita' coinvolte	4
I giocatori	4
L'arbitro	4
La palla	4
Gli allenatori	4
Lo stato	4
Il modello	5
Interazione tra giocatori: il controllore	5
Verifica sullo stato di gioco: l'arbitro	6
La palla in movimento: l'entita' e l'agente di movimento	8
Modifiche sulle squadre: gli allenatori	9
Gestione delle fasi di gioco: entita' di gioco	10
Analisi architetturale	11
Gli eventi	11
Concorrenza	13
Stato	13
Controller ed arbitro	13
Giocatori	13
Palla	13
Distribuzione	14
IA dei giocatori	15
Implementazione	16
Concorrenza	16
Distribuzione	16
Compilazione ed esecuzione	17
Conclusioni	18

Introduzione

Paragrafo introduttivo.

Scopo del progetto

Scopo del progetto.

Funzionalità del software

Funzionalità del software.

La partita

La partita.

Il software

Il software.

Obiettivi

Obiettivi.

Entita' coinvolte

Paragrafo introduttivo.

I giocatori

I giocatori.

L'arbitro

L'arbitro.

La palla

La palla.

Gli allenatori

Gli allenatori.

Lo stato

Lo stato.

Il modello

Nel Capitolo sono state individuate ed analizzate le entita' che caratterizzano il sistema. La fase successiva, su cui questo capitolo si focalizza, consiste nel definire le interazioni che legano le suddette entita', in maniera tale da poter derivare un modello per il sistema.

Interazione tra giocatori: il controllore

I giocatori sono i principali attori del sistema. Un giocatore si sposta dentro e fuori dal campo, e' in grado di compiere diversi tipi di azioni (Sezione), che decide sia sulla base del proprio stato che dello stato della propria squadra e della partita. Un aspetto fondamentale vede i giocatori, cosi' come avviene nel mondo reale, operare in maniera parallela e concorrente tra di loro: questo implica che i risultati delle loro azioni debbano riflettersi sullo stato di gioco, che come descritto in Sezione , detiene tutte le informazioni che definiscono la partita in un dato istante. Si pone pero' un problema: i giocatori, per poter conoscere le informazioni che permetteranno loro di decidere la prossima azione da compiere, i giocatori devono accedere allo stato, che tuttavia e' unico. E' dunque necessario regolamentare l'accesso a tale risorsa, in maniera tale da preservarne la consistenza e la correttezza.

L'organo che si occupa di permettere l'interazione tra diversi giocatori e' il *controllore*. In quanto entita' centrale di controllo, i suoi molteplici compiti possono essere raggruppati come segue:

1. Permettere ai giocatori di accedere allo stato, sia per leggerne le informazioni sia per poterle modificare
2. Sequenzializzare gli accessi allo stato da parte dei giocatori, al fine di non creare inconsistenze sulle informazioni in esso contenute
3. Ricoprire il ruolo di arbitro di gioco "onnisciente" (verra' dettagliato in Sezione)

Come gia' accennato precedentemente, un giocatore decide la sua prossima mossa sulla base dello stato corrente della partita: ad esempio, trovandosi in possesso della palla nell'area avversaria e senza nessun giocatore a marcarlo, e' ragionevole che il giocatore decida di tirare per provare a realizzare un goal. Al tempo stesso, l'azione che egli compie si ripercuote sullo stato della partita, andandone a modificare una parte: di nuovo, se il giocatore segna, il punteggio della partita cambia e i giocatori ritornano in posizione di partenza per ricominciare il gioco. Questo meccanismo mette in luce una necessita' tale per cui le operazioni che alterano lo stato siano quanto piu'

possibile sequenziali ed atomiche. Sebbene questo argomento verra' trattato ampiamente nel Capitolo , e' importante capire il motivo dell'importanza di queste due caratteristiche.

L'assunzione di atomicita' si rivela particolarmente critica quando si assume di operare in condizioni di prerilascio dei processi. Sotto questa condizione, un processo correntemente in esecuzione puo' essere "temporaneamente fermato" (prerilasciato) in favore di un altro processo, che entra quindi in esecuzione al suo posto; una simile condizione si verifica, ad esempio, se il processo corrente ha una priorita' inferiore di quello che vuole subentrare. Per spiegare come il prerilascio minacci la consistenza dello stato della partita, si consideri una situazione dove due giocatori avversari si contendono il possesso della palla, che giace inerte in mezzo a loro. Entrambi i giocatori leggono lo stato e avviano il processo di decisione della prossima azione. Tuttavia, il processo relativo al primo giocatore viene prerilasciato. Nel frattempo, il secondo giocatore conquista la palla e si muove verso la porta avversaria. Quando il primo giocatore torna in esecuzione si trova in uno stato inconsistente, in quanto lo stato e' cambiato senza che lui lo sappia. Ancora peggio, se il primo giocatore procede con la scrittura della sua azione, lo stato verra' modificato erroneamente, potenzialmente togliendo il possesso di palla al secondo giocatore.

[TODO] aggiungere la parte di mutua esclusione

Verifica sullo stato di gioco: l'arbitro

Affiche' una partita si svolga secondo le regole e le modalita' stabilite dal gioco del calcio c'e' bisogno di un arbitro che regoli l'andamento del gioco. Le mansioni dell'arbitro sono molteplici:

- Sancire l'inizio e la fine dei tempi di gioco (inizio primo tempo - fine primo tempo - inizio secondo tempo - fine partita)
- Fermare il gioco e farlo riprendere a seguito (e.g. una rimessa laterale)
- Segnalare eventuali irregolarita' da parte dei giocatori (e.g. un fallo)
- Tenere il conto dei gol segnati da entrambe le squadre, cosi' da decretare il vincitore alla fine della partita
- Gestire le richieste di sostituzione e di cambio di formazione da parte degli allenatori

L'arbitro deve quindi essere in grado di controllare tutte le mosse dei giocatori, cosi' come lo stato e la posizione della palla e la durata della

partita fino a quel momento. Nella realtà, il ruolo dell'arbitro è assegnato ad un essere umano, che quindi non è infallibile: si pensi ad esempio ad un fallo che viene commesso irregolarmente alle sue spalle mentre lui è impegnato ad assegnare un calcio d'angolo. In questa simulazione si assume più semplicemente che l'arbitro sia "onnisciente", ovvero abbia la facoltà di analizzare ogni singola mossa di ciascun giocatore e della palla, in maniera da poter segnalare immediatamente ogni irregolarità oppure fermare il gioco all'occorrenza.

Si ha così che il controllore, descritto nel paragrafo precedente, ricopre anche il ruolo di arbitro. Questa decisione ha delle ripercussioni non solo nello svolgersi del gioco (l'arbitro è onnisciente), ma anche nell'assegnazione delle risorse di calcolo. Infatti, se l'arbitro fosse soggetto agli stessi vincoli di esecuzione dei giocatori, andrebbe a concorrere assieme a loro per l'esecuzione sulla CPU come task a se stante; questa situazione non si verifica invece nel caso in cui sia il controllore ad essere anche arbitro, essendo l'entità centrale che si occupa di eseguire a tutti gli effetti le mosse dei giocatori. Maggiori dettagli sull'implementazione dell'arbitro verranno esposti in Sezione .

L'elenco delle funzioni che l'arbitro deve espletare nasconde tuttavia un punto critico su cui è opportuno soffermarsi. I primi quattro punti sono strettamente legati alla componente concorrente della simulazione, ovvero quella che vede l'interazione dei giocatori con il controllore e, in alcuni casi, l'agente di movimento. L'ultimo punto fa invece riferimento alla componente distribuita della simulazione, ovvero quella che si occupa di ricevere e gestire i comandi impartiti dagli allenatori ed eventualmente dalla finestra principale di controllo (che, come verrà esposto successivamente, coincide con la componente che mostra il campo di gioco e lo svolgersi della partita). La differenza sostanziale tra queste due tipologie di eventi è racchiusa nel fatto che gli eventi provenienti dalla componente distribuita non sono deterministici e non seguono nessuna regola di generazione, a differenza degli eventi relativi alla parte concorrente. La presenza di due sorgenti di eventi introduce un problema significativo, ovvero l'ordine in cui l'arbitro deve processare/consumare quegli eventi. Ad esempio, si consideri una situazione dove si ha una richiesta di sostituzione per il giocatore 1 in favore del giocatore 3 e, allo stesso tempo, sia stato commesso un fallo commesso dal giocatore 1 sul giocatore 2. L'ordine in cui vengono processati questi eventi determina lo stato successivo, che diverge a seconda che si consideri prima uno oppure prima l'altro. Bisogna tenere conto, ad ogni modo, che gli eventi che vengono generati dalla componente distribuita

hanno come preconditione il gioco fermo: quindi c'è una stretta dipendenza unidirezionale tra un evento singolo della componente concorrente e gli eventi della componente distribuita. L'approccio da seguire per garantire il massimo livello di correttezza temporale è quindi il seguente: l'arbitro dovrà prima processare l'evento singolo della componente concorrente (che può causare il gioco fermo, nel caso non lo fosse già) e solo poi processare gli eventi della componente distribuita, se le condizioni sono opportune.

La palla in movimento: l'entità e l'agente di movimento

La palla è una parte fondamentale del modello della simulazione, in quanto il suo possesso viene conteso dai giocatori che devono tirarla in porta, segnando un gol per la loro squadra.

In ogni momento della partita, la palla occupa una delle celle del campo. Nel caso sia posseduta da un giocatore, essi condividono la stessa cella; in caso contrario, la palla occupa la cella in cui si trova. Inoltre, la palla può trovarsi solamente in due stati: inerzia e moto. Una palla in movimento si può avere quando il giocatore che la controlla si sposta con essa; inoltre, si ha una palla in movimento anche quando un giocatore la passa verso un altro giocatore oppure effettua un tiro verso la porta avversaria. Al contrario, una palla è inerte se non è controllata da nessun giocatore, solitamente quando un passaggio o un tiro mancano il bersaglio (e.g. un passaggio troppo debole).

Ad ogni modo, la palla è un'entità passiva, che non compie azioni proprie ma che subisce azioni di altre entità attive (i giocatori). Di conseguenza, quando un giocatore effettua un passaggio oppure un tiro, la palla deve essere spostata da un'entità che però non può essere il giocatore stesso: in altre parole, si tratta di simulare l'impressione di un moto alla palla a seguito di un'azione del giocatore che la controlla. Questa mansione è ricoperta dal cosiddetto *agente di movimento*. L'agente di movimento è un'entità concorrente agli altri giocatori, il cui unico compito è quello di spostare la palla in una determinata direzione fino a che la potenza ad essa impressa è sufficiente a farla avanzare alla cella successiva. Una volta completato il suo compito, smette di eseguire in attesa del prossimo spostamento da effettuare. In alcuni casi, l'agente di movimento viene volutamente bloccato attraverso l'arbitro, ad esempio nel caso in cui la palla esca dal campo e sia necessario assegnare una rimessa oppure un calcio d'angolo.

Modifiche sulle squadre: gli allenatori

Nel gioco del calcio i giocatori di ciascuna squadra vengono coordinati dal proprio allenatore, che ha il compito di scegliere come disporre i giocatori in campo (la formazione) e di effettuare delle sostituzioni, come conseguenza di un infortunio o piu' semplicemente per una scelta tattica. In questa simulazione i due allenatori rappresentano due componenti distribuite separate, che comunicano con l'unita' centrale di controllo. Ciascun allenatore ha la facolta' di prendere le seguenti decisioni:

- effettuare un cambio di formazione per la propria squadra
- effettuare una sostituzione di un giocatore con un suo compagno presente in panchina

Le decisioni prese da un allenatore hanno come preconditione necessaria il gioco fermo: sara' pertanto l'arbitro a dover esaminare e successivamente accettare le richieste di un allenatore solo quando le condizioni lo permettono. Nel caso di un semplice cambio di formazione, la decisione dell'allenatore viene applicata sulla squadra, cosicche' i giocatori al proprio turno successivo sappiano che la loro posizione di riferimento e' cambiata. Diverso e piu' complesso e' invece il caso della sostituzione. La sequenza di operazioni che seguono una richiesta di sostituzione si possono schematizzare come segue:

1. L'arbitro riceve la richiesta di sostituzione e, non appena il gioco e' fermo, procede a notificarlo ai giocatori
2. Ciascun giocatore, prima di decidere la propria mossa, controlla se l'allenatore ha deciso di sostituirlo con un compagno
3. Nel caso del giocatore interessato alla sostituzione, esso si dirige verso la panchina (eventualmente lasciando la palla dove si trova)
4. Una volta giunto in panchina, il suo compagno prende il suo posto in campo e si dirige verso la propria posizione di riferimento
5. L'arbitro, ad ogni turno, controlla se il giocatore entrante ha raggiunto la posizione di riferimento e, in quel caso, sancisce la ripresa del gioco

Il comportamento sopra descritto vale anche nel caso di piu' sostituzioni simultanee, che vedono i giocatori uscire ed entrare nel campo allo stesso tempo.

Ciascun allenatore ha inoltre una visione meno precisa del gioco e riceve pertanto un sottoinsieme degli eventi che caratterizzano una partita: dal

punto di vista decisionale e' infatti poco interessante per un allenatore conoscere ogni singolo movimento di ogni giocatore. Si ha quindi che l'allenatore viene notificato solo in caso di eventi "salienti", che come descritto nel Capitolo , sono stati denominati *Game Events*.

Gestione delle fasi di gioco: entita' di gioco

[secondo me non va messa qui, e' prettamente implementativa]

Analisi architetturale

In questo capitolo verranno presentate le decisioni architetturali inerenti la struttura e l'organizzazione del software di simulazione. L'analisi comincia con una visione generale della gerarchia di eventi, per poi suddividersi nelle scelte che riguardano la parte di concorrenza e le scelte che riguardano la parte di distribuzione.

Gli eventi

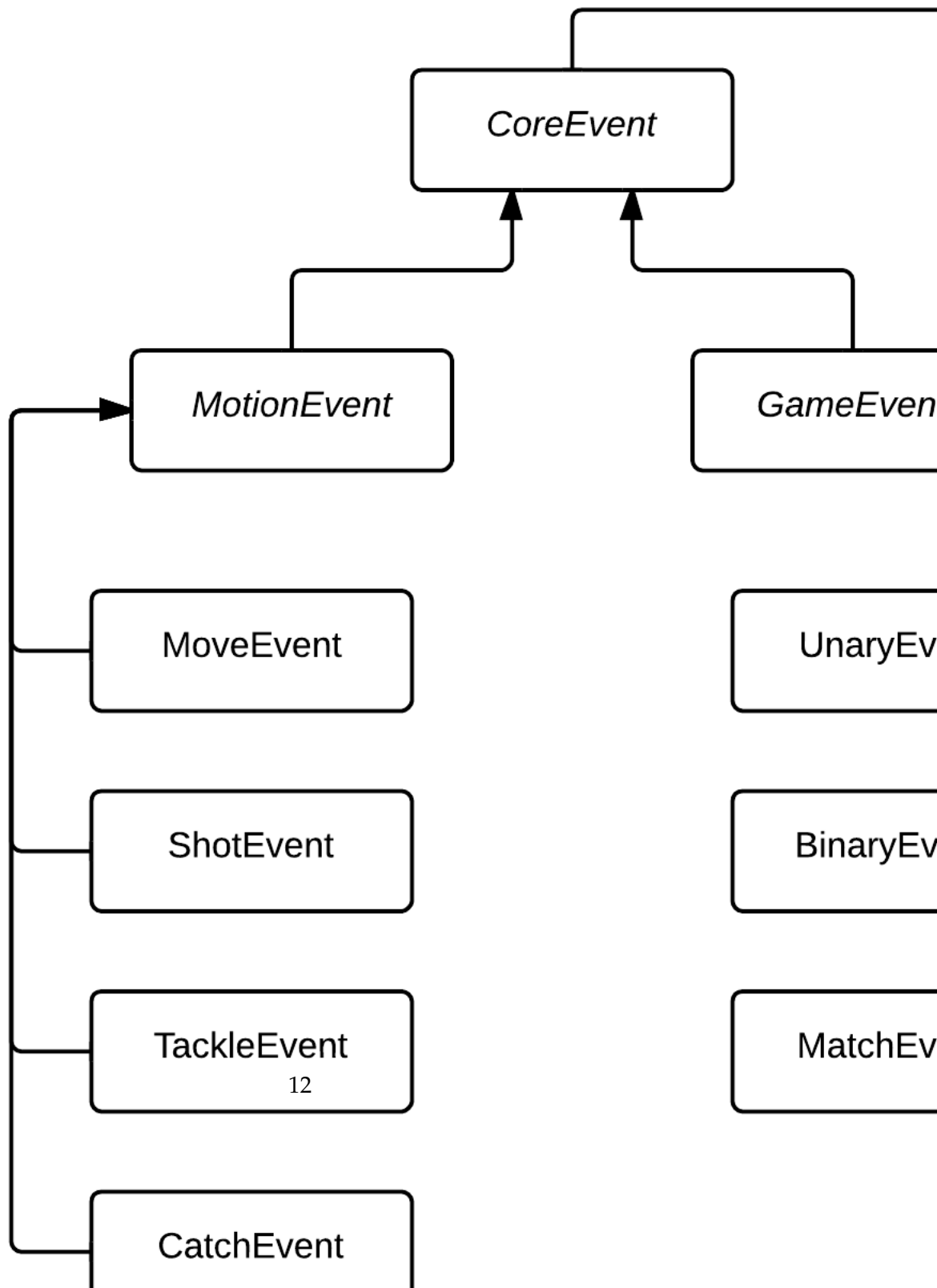
Gli eventi sono l'elemento costituente di una partita. Essi possono essere generati dalle diverse entità e possono essere raggruppati in diverse categorie. Inoltre, alcuni di questi eventi sono interessanti solo per la parte concorrente, mentre altri eventi possono viaggiare dalla parte concorrente a quella distribuita e viceversa.

La struttura globale degli eventi è schematizzata in Figura 1. Data la sua complessità ed estensione, verranno ora trattati singolarmente i diversi tipi di eventi al fine di spiegarne il loro significato e il contesto in cui vengono utilizzati.

Event *Event* rappresenta l'evento generico ed è alla base della gerarchia. Da esso si diramano due macro-categorie di eventi: i *CoreEvent* e i *ManagerEvent*. Essi sono rispettivamente gli eventi che vengono generati nella parte concorrente (il cosiddetto "Core") e gli eventi che vengono generati nella parte distribuita (fondamentalmente, dagli allenatori). *Event* può essere considerata come un'entità astratta, che non viene concretizzata se non da uno dei suoi derivati.

CoreEvent Gli eventi di tipo *CoreEvent* sono un insieme di eventi che vengono generati dalla parte concorrente del sistema, ovvero il Core. La loro generazione tuttavia non vincola il loro utilizzo nella sola parte concorrente: essi vengono infatti inviati alla parte distribuita per notificare gli allenatori (e l'interfaccia grafica del campo) sullo svolgersi della partita. Vi sono due tipologie di *CoreEvent*: i *MotionEvent*, che rappresentano le possibili azioni dei giocatori, e i *GameEvent*, che invece rappresentano tutti quegli eventi che influiscono sullo stato di gioco. Anche in questo caso, i *CoreEvent* sono astratti e trovano una concretizzazione nei loro discendenti.

MotionEvent Tutte le azioni che un giocatore può compiere sono definite dai *MotionEvent*. Sono stati definiti quattro tipi di *MotionEvent*, elencati di seguito.



- *MoveEvent* - Descrivono i movimenti di un giocatore, dal punto in cui si trova al punto in cui si vuole spostare. Questi eventi vengono altresì usati per descrivere gli spostamenti della palla.
- *ShotEvent* - Rappresentano il tiro/passaggio effettuato da un giocatore, e sono caratterizzati da alcune informazioni quali la posizione del giocatore e la potenza impressa alla palla.
- *TackleEvent* - Corrisponde al tentativo di contrasto verso un altro giocatore.
- *CatchEvent* - Questo evento descrive il gesto di prendere possesso di una palla, sia essa inerte sul campo (non in possesso) oppure come intercettazione di una palla in movimento.

Ciascuno di questi eventi viene sottoposto all'attenzione dell'arbitro, che ne valida la correttezza nel rispetto delle regole del gioco. Inoltre, come già anticipato, questi eventi vengono anche inviati alla parte distribuita, così da poter aggiornare la visualizzazione della partita e per permettere agli allenatori di prendere decisioni tattiche.

GameEvent Gli eventi che regolano lo svolgimento del gioco rientrano nella categoria di *GameEvent*.

Concorrenza

La concorrenza.

Stato

Lo stato.

Controller ed arbitro

Controller ed arbitro.

Giocatori

Giocatori.

Palla

Palla.

Distribuzione

Distribuzione.

IA dei giocatori

IA dei giocatori.

Implementazione

Paragrafo introduttivo.

Concorrenza

Concorrenza.

Distribuzione

Distribuzione.

Compilazione ed esecuzione

Compilazione ed esecuzione.

Conclusioni

Conclusioni.