Gestione del movimento di Boid tramite libreria ActoDes

Progetto di sistemi distribuiti

Porta Michele

2019/2020

Obiettivo del progetto

L'obiettivo del progetto è sviluppare una API per implementare le diverse regole che gestiscono il movimento di boid presentate durante il corso, il tutto utilizzando la libreria software Java ad attori ActoDeS.

La libreria Actodes

ActoDeS è utilizzabile usando la JDK 11 o 12 e JavaFX 12.

In ActoDeS un sistema si basa su una serie di attori interagenti che svolgono compiti contemporaneamente.

Un attore è un oggetto concorrente ed autonomo, che interagisce con altri attori scambiando messaggi asincroni.

In risposta ad un messaggio, un attore può inviare messaggi ad altri attori o a sé stesso, creare nuovi attori, aggiornare il proprio stato locale, cambiare comportamento, killarsi o prepararsi per ricevere ed elaborare il messaggio successivo.

Un attore è composto da quattro componenti principali: reference, mailbox, behavior e state.

Actodes offre già un esempio di simulazione del movimento di particelle che mostra l'usuale comportamento di un attore che simula dei boid.

Un attore non può accedere allo stato degli altri attori e quindi non può determinare il suo movimento in base allo pseudocodice con cui vengono normalmente presentate le regole che guidano il movimento.

Ad ogni passo di simulazione, ogni attore manda in broadcast un messaggio contenente le informazioni sul suo stato e calcola i valori associati alle regole con i messaggi ricevuti dagli altri attori fino a ricevere un messaggio di conclusione del passo di simulazione in cui aggiorna la propria posizione e velocità e rinvia le informazioni del proprio nuovo stato.

L'esecuzione di ogni boid genera un file log che contiene le informazioni sui passi della simulazione. Questo file log può essere processato dalla "*SimulationGui*" che genera una visualizzazione grafica dei passi di simulazione in cui è possibile agire da interfaccia visualizzandone un passo singolo alla volta. Inoltre, via codice è possibile impostare la velocità di esecuzione di un ciclo di simulazione che per questo scopo è stata impostata a 0.1 secondi.

L’idea è quella di impostare un obiettivo situato in una zona lontana dal gruppo di boid generati inizialmente. I boid si muoveranno casualmente all’inizio ma dovranno a mano a mano seguire alcune regole.

Nel caso i boid, muovendosi in modo random, entrino nel raggio di conglomerazione imposto via codice di un altro boid entreranno a far parte automaticamente di un gruppo e si muoveranno all’unisono secondo passi preimpostati alla ricerca del goal.

Quando uno dei boid entra nel raggio di azione dell’obiettivo, simulandone quindi l’individuazione, manda un messaggio in broadcast ad ogni boid presente nella scena per avvisarli dell’individuazione dell’obiettivo finale.

A questo punto tutti i boid si muoveranno all’unisono verso l’obiettivo ad una velocità impostabile dall’esterno fino a raggiungerne la prossimità.

I boid, arrivati tutti ad una certa distanza rispetto al centro dell’obiettivo, si coordineranno fra di essi per posizionarsi in posizioni equidistanti fra di essi per fare in modo di non sovrapporsi e di non creare una massa di boid nello stesso punto. I boid attenderanno vicino al goal fino al termine della simulazione.

Questa implementazione può essere vista anche nell’ambito dell’intelligenza artificiale, e più precisamente nell’ambito della swarm intelligence.

Swarm Intelligence

Si tratta dell’Insieme di tecniche di ottimizzazione ispirate al comportamento collettivo di sciami/branchi/stormi di animali.

Vengono sfruttate le scoperte individuali e collettive per massimizzare alcuni obiettivi, come ad esempio trovare l'ottimale all'interno di uno spazio di ricerca, sulla base di meccanismi di comunicazione impliciti o espliciti, come ad esempio in:

- Ant Colony Optimization (ACO), basato sul modo in cui le formiche comunicano tra loro rilasciando feromone mentre si muovono.
- Ottimizzazione dello sciame di particelle (PSO), che simula il movimento degli stormi di uccelli sui punti in cui il resto dello stormo ha trovato cibo.

Il funzionamento nel dettaglio

Ogni singolo boid viene rappresentato graficamente attraverso un cerchio blu di raggio 4 (sia colore che raggio sono impostabili). L’obiettivo sarà rappresentato da un cerchio molto più grande (raggio 15) e quindi sarà ben visibile agli occhi dell’utente.

Il numero di boid è stato impostato a 20 e vengono generati casualmente all’interno dell’area di simulazione.

È stata presa l’accortezza di farli generare non nell’intera area di simulazione ma su una sua frazione (impostata a 1.4) in modo tale da impedirne la generazione subito in prossimità

dell'obiettivo. Obiettivo che è stato impostato nella parte inferiore del quadrato di simulazione alle coordinate x=400 e y=400.

I boid, dopo essere stati generati, iniziano a muoversi casualmente e alla cieca. Per ottenere questo risultato ad ogni iterazione di simulazione si aggiungono, attraverso il metodo *add* relativo alla classe *doublePairVector*, due numeri generati anch'essi casualmente tramite la classe RANDOM, in modo tale da sommare un numero randomico da 5 a -5.

Ad ogni passo di simulazione i boid si scambiano messaggi. Ogni boid manda in BROADCAST un oggetto denominato BoidInfo, una classe creata custom per questo progetto, che è composto da un doublePairVector, ovvero un oggetto contenente la propria posizione espressa in coordinate x e y, ed un booleano denominato *find* che se impostato a *true* ha il significato che è stato individuato il goal.

Il primo boid che entra nel raggio di azione del *goal,* impostato di dimensione pari ad una sfera di raggio pari al doppio del diametro del goal stesso, setterà il flag *find* a true e procederà con l'inviare il messaggio tramite l'oggetto boidInfo.

Il messaggio, essendo spedito in BROADCAST, arriverà a tutti gli altri boid che leggendone il contenuto si appresteranno immediatamente a muoversi con una velocità maggiore rispetto al normale verso la posizione del gol fino a trovarlo anch'essi ed arrestarsi in tale posizione.

L'obiettivo imposto ai boid di raggiungere il goal si avvale di alcune regole di movimentazione tipiche della teoria di movimentazione dei boid, fra cui ***coesione***, ***separazione*** *ed* ***allineamento***.

All'inizio della simulazione ogni boid risulta un'entità a sé stante e non ha alcun legame con nessuno degli altri boid presenti nella scena.

È stato impostato un cerchio figurativo di raggio 20 che sancisce la distanza per la quale un boid entra in contatto con un altro boid e formano un gruppo. Quando un boid entra od esce dal raggio di azione di un altro boid viene attivato o disattivato un booleano che fa da flag che li associa ad un gruppo o meno.

Una volta in gruppo i boid si muoveranno all'unisono cambiando direzione ogni tot passi di simulazione (anch'essa impostabile dall'esterno) sempre in cerca del GOAL imposto.

Quando un boid esce dalla distanza prefissata, il flag verrà impostato su false ed il boid tornerà a movimentarsi in modo casuale allo stesso modo dell'inizio della simulazione.

La simulazione

Il comportamento dei boid viene visualizzato tramite il simulatore presente all'interno della libreria Actodes, che permette di visualizzare graficamente, al succedersi di un tempo virtuale, l'andamento dei boid e il loro approcciarsi verso il raggiungimento dell'obiettivo imposto.

La simulazione avviene in uno spazio quadrato delle dimensioni di 500 x 500 in cui è possibile visualizzare lo scorrere del tempo virtuale. Il tempo può essere impostato sia in numero di quanti virtualizzati che in durata espressa in millisecondi.

In questo caso la simulazione è stata impostata su 1000 quanti della durata di 100 ms ciascuno, quindi con durata totale massima della simulazione di 100 secondi.

La simulazione deve essere avviata con l'apposito tasto play e può essere interrotta con il tasto stop e successivamente fatta ripartire dallo stesso punto precedentemente interrotto. È possibile scorrere all'indietro passo per passo con l'apposito pulsante di rewind per focalizzarsi su punti precedenti.

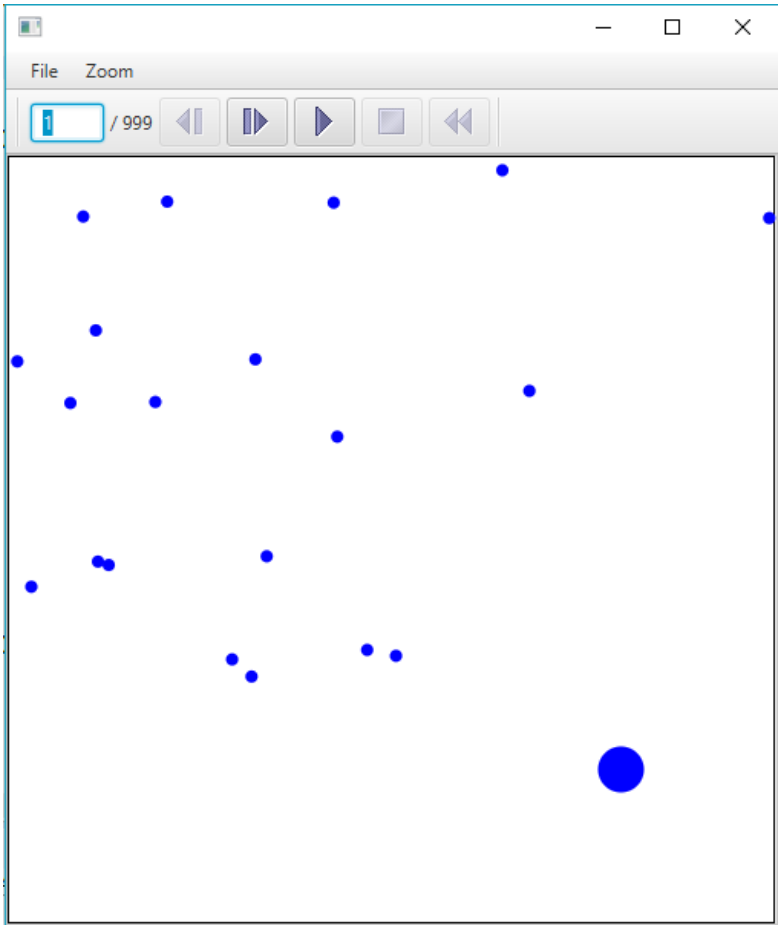


Figura 1: L'inizio della simulazione

Inoltre, è presente una funzionalità di zoom, sia in avanti che indietro.

Importate notare che non è garantito che nel tempo di simulazione impostato i boid riescano a trovare il GOAL in tempo, in quanto si tratta di una simulazione basata sulla casualità degli eventi.

In caso un boid riesca ad entrare nel raggio di prossimità dell'obiettivo, indicato in Figura 2 tramite un cerchio di colore azzurro chiaro, comunica immediatamente in BROADCAST la posizione dell'obiettivo.

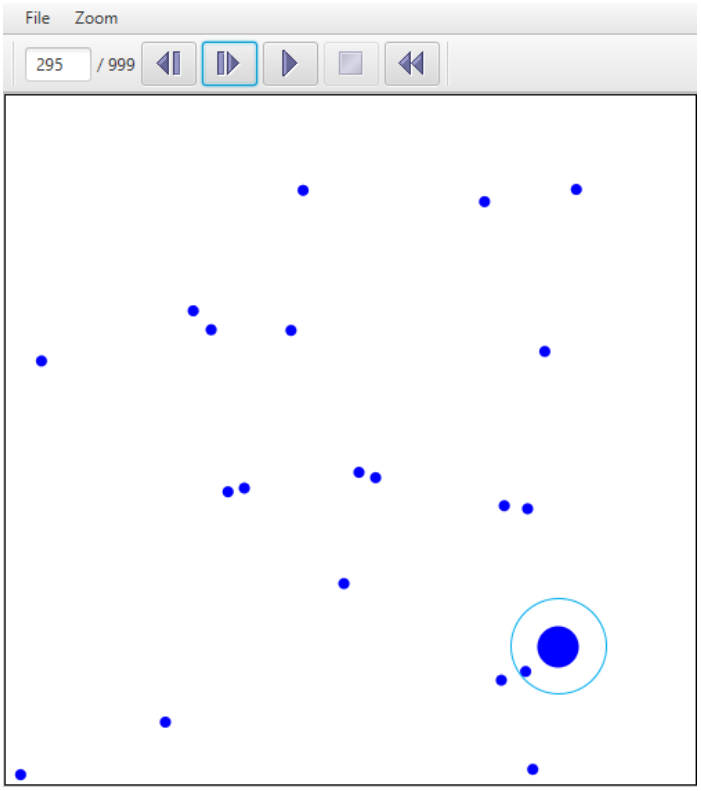


Figura 2: Goal trovato dal primo boid

I restanti boid ignoreranno le regole precedentemente imposte per recarsi in massa verso l'obiettivo a velocità impostabile dall'esterno.

Una volta arrivati in vicinanza del goal i boid si coordineranno fra di essi e si spargeranno in modo tale da non sovrapporsi e rimarranno in tale posizione fino alla fine della simulazione.

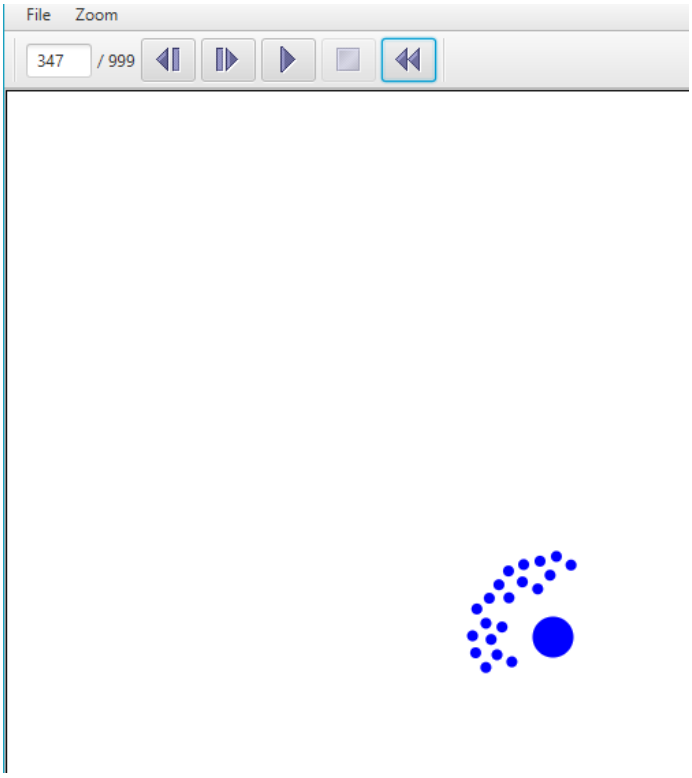
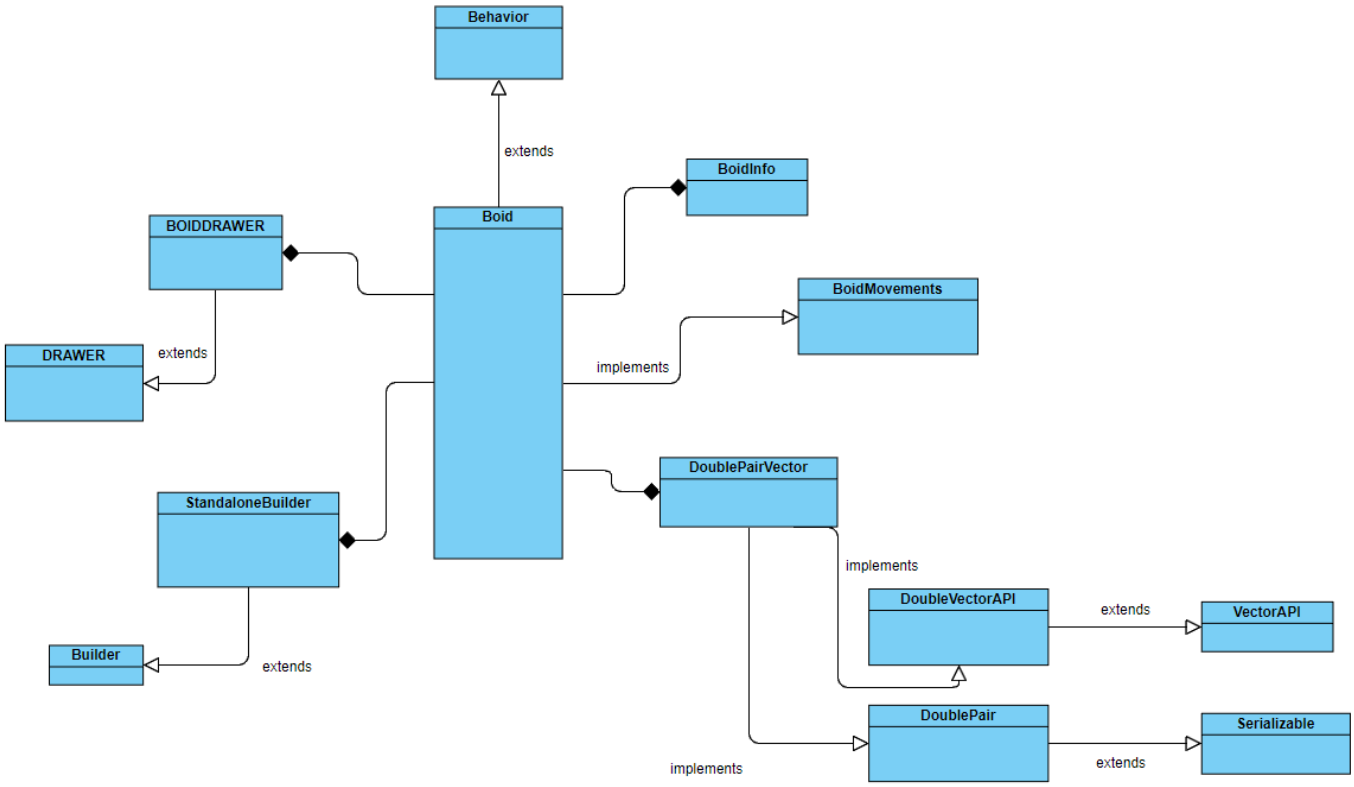


Figura 3:Goal raggiunto da tutti i boid

Il Diagramma UML delle classi dei boid



Il Diagramma UML delle classi del simulatore