Progetto robot swarm

METODOLOGIE DI PROGRAMMAZIONE

ARCANGELI LORENZO

A.A. 2022/2023

Sommario

Introduzione
Il progetto richiedeva la realizzazione di una libreria Java che consenta la simulazione di uno sciame di robot all'interno di uno spazio bidimensionale infinito. Si identificano i robot come punti dello spazio. All'interno di questo sono presenti sotto aree (di forma rettangolare o circolare) che identificano delle particolari condizioni dell'ambiente.
Ogni robot presente all'interno di esso deve avere la possibilità di:

- percepire l'ambiente che li circonda
- segnalare una condizione
- muoversi nella direzione dei robot ad una certa distanza
- muoversi verso una certa direzione
- fermarsi

L'insieme di queste possibilità sono dettate da un programma, il quale verrà eseguito in maniera indipendente da ogni robot in relazione alle condizioni proprie e quelle percepite nell'ambiente.
La velocità di movimento del robot è espressa in metri al secondo, tenendo presente che l'esecuzione di un comando richiede un secondo di tempo. Questa velocità può essere modificata in base alle specifiche dell'utente.
Il programma eseguito dai robot, oltre a contenere istruzioni per: movimento, movimento random, segnalazione di una condizione, fine segnalazione di una condizione, seguire la posizione di altri robot, continuare il proprio movimento e stopparsi, include anche comandi di loop.
In particolare, si vuole gestire le ripetizioni di una o più istruzioni:

- per un numero limitato di volte
- fino a quando una condizione non viene percepita
- per sempre

Si dà la possibilità di inserire dei loop nidificati all'interno del programma.
Questo può essere eseguito step by step o per un determinato periodo di passi.

Responsabilità individuate
In base alle specifiche del progetto, sono state individuate le seguenti responsabilità.

Entity
Rappresenta una entità all'interno dell'ambiente a cui sono associate coordinate direzione e velocità. Ogni entità potrà segnalare o smettere di segnalare una condizione. Può eseguire una serie di comandi la cui gestione non è parte di questa responsabilità.

Environment
La seguente responsabilità permette di identificare lo spazio in cui sono presenti i robot come descritto all'interno delle specifiche. Tramite questa responsabilità si vuole mettere in relazione le entità tra di esse e rispetto alle condizioni dello spazio.

Comandi
Indica lo stato successivo dell'ambiente una volta applicato un comando. Tramite questa responsabilità si va a modificare lo stato delle entità in base ai comandi eseguiti.

Condizione
Rappresenta le condizioni presenti all'interno dell'ambiente. Identificando il proprio centro e l'estensione (relativa alla forma). In questo modo si riesce a determinare se una entità è presente al suo interno o meno.

Classi individuate
Una volta individuate le responsabilità queste sono state tradotte in classi ed interfacce. In particolare, le responsabilità sono diventate rispettivamente le interfacce: Entity, Environment e Command. La responsabilità della Condizione dello spazio è stata tramutata nella classe ShapeData.
Le interfacce Command ed Environment sono parametrizzate rispetto ad Entity in modo da poter gestire sia robot come in questo caso che eventuali future implementazioni di Entity.

RobotFollowMeParserHandler
Ha il ruolo di effettuare il parse dei comandi letti dalle linee del file. Va quindi a convertire ogni istruzione in un nuovo oggetto (rappresentante un comando), aggiungendolo alla lista delle istruzioni. Una volta terminata la lettura di tutto il file, ottenendo quindi la lista completa dei comandi, questa viene caricata all'interno dei Robot.

Classi dei comandi
Dall'interfaccia Command, si ricavano undici classi. Una per ogni tipologia di comando eseguibile. In base alla tipologia, si richiede la collaborazione o meno con l'ambiente. Ad esempio, per quanto riguarda i comandi di movimento, verrà calcolata la direzione e quindi sommata rispetto alle coordinate che indentificano il robot in relazione alla velocità di moviemento.
Tramite questa implementazione è possibile modificare lo stato del robot e agire sull'instruction pointer del programma di ogni robot.

InstructionPointerHandler
Questa classe permette di pilotare l'instruction pointer del programma che i robot devono eseguire. Avendo un insieme lineare di comandi, permette di gestire eventuali loop o loop nidificati tramite il salvataggio degli indici dei comandi rilevanti per la gestione. Ovvero i comandi di loop (repeat, until e do forever) e i comandi di Done, che rappresentano rispettivamente la fine e l'inizio di ogni ciclo.

Coordinate
Viene usata per semplificare la rappresentazione delle coordinate dei vari elementi presenti all'interno dello spazio.

Robot
L'interfaccia Entity è implementata dalla sola classe robot, questa contenendo le informazioni descritte precedentemente, permette lo scorrimento dei comandi da eseguire. Si noti che la gestione dell'esecuzione dei comandi e quale comando eseguire non rientra tra le caratteristiche di questa classe. Le quali sono gestite dalle classi InstructionPointerHandler e le classi dei comandi.

RobotEnvironment
Implementa l'interfaccia Environment parametrizzandola per i Robot. Mette in relazione i robot tra loro e con le condizioni dello spazio. I comandi di Follow e Until fanno uso delle informazioni messe a disposizione da questa classe. Ovvero verificare quali robot all'interno di una certa distanza segnalano una condizione e verificare se un robot si trova all'interno di una sotto area o meno.
È compito inoltre di questa classe dare il via all'esecuzione dello step successivo, la cui istruzioni corrispondente, come detto prima, è indipendente per ogni robot e gestita dalle classi precedentemente descritte.
Mette inoltre a disposizione una rappresentazione scritta di esso.

Normalizer
Le specifiche del progetto segnalano una limitazione per quanto riguarda i valori del comando Move. Questi non possono superare il valore di 1 (o -1). Nei comandi Follow e Move Random non si verificano questi vincoli, è dunque necessario andare a normalizzare i valori all'interno del range che va da -1 a 1 e rispettare quindi l'idea dei comandi di movimento.

RobotSwarmExecutor
Anche questa classe è parametrizzata sulle Entity. Permette di gestire lo swarm di robot durante l'esecuzione di un programma.
Step by step viene salvato l'ambiente e inserito all'interno di una cronologia di ambienti.
Tramite questa implementazione è possibile sia eseguire sia un nuovo step che tornare indietro nella cronologia degli ambienti precedentemente salvati e mostrare quindi l'ambiente.

Controller
Il funzionamento della classe controller è del tutto analogo rispetto alla classe RobotSwarmExecutor. Essendo usata per la gestione dell'interfaccia grafica, questa mette a disposizione la possibilità di lavorare con file scelti dall'utente per quanto riguarda i comandi da eseguire e le condizioni dell'ambiente

Interfaccia grafica
Per la realizzazione dell'interfaccia grafica è stata usata principalmente la seguente classe.

RobotSwarmController
La classe RobotSwarmController tramite l'utilizzo della classe Controller permette di effettuare le operazioni de step forward e backward. Inoltre, permette di modificare la velocità di esecuzioni di ogni istruzione (modificando il tempo dello spostamento) e di eseguire solo un certo numero di istruzioni. Presenta anche la possibilità di effettuare le operazioni di zoom in/out e lo scroll nelle quattro direzioni.

Istruzioni da eseguire
Per eseguire correttamente il progetto, è necessario impostare il numero di robot che si vuole creare e premere il pulsante start. Si richiede quindi di selezionare dei file riguardanti i comandi dei robot e le condizioni ambientali. Si possono inoltre impostare velocità di movimento e numero di istruzioni massimo da poter eseguire.