SwarmRobot: una libreria Java per la simulazione di uno sciame di robot

Cristiano Aloigi

27 Giugno 2023

**1 Introduzione**

Il progetto SwarmRobot ha come obiettivo la realizzazione di una libreria Java, con interfaccia grafica annessa, per la simulazione di uno sciame di robot. I robot andranno ad operare in un’ambiente bidimensionale di dimensioni illimitate, dove posso muoversi liberamente.

**2 Dettagli Tecnici**

Il progetto è sviluppato in Java 18 ed utilizza il sistema di gestione dei progetti Gradle, assicurando una struttura del codice semplice e molto intuitiva, in più sono anche presenti i vari test. L’interfaccia grafica è realizzata tramite l’utilizzo di JavaFX 20, una libreria grafica che permette di realizzare applicazioni per desktop utilizzando il linguaggio Java.

**3 Responsabilità assegnate**

* La classe Controller si occupa di gestire tutta l’esecuzione della simulazione, dalla gestione delle aree presenti alla gestione di tutti i robot, si occupa anche di inizializzare il parser per poter caricare i vari dati. Durante l’inizializzazione delle aree si avvale dell’utilizzo di uno ShapeParser per poter convertire gli ShapeData in elementi IShape
* La classe ParserHandler va ad implementare il parsing delle figure e del programma, nel caso di quest’ultimo diviene anche un piccolo compilatore andando a verificare la correttezza sintattica del programma, dopo di che carica la lista di istruzioni a tutti i robot.
* La classe Program si occupa di gestire la lista delle istruzioni che ogni robot deve eseguire, ogni robot avrà una copia di questa classe, avendo così ognuno il suo proprio program counter, in modo tale da separare l’esecuzione di ogni robot.
* La classe ShapeParser va a convertire oggetti ShapeData (generati alla lettura del file delle aree) in oggetti IShape, tramite l’utilizzo di metodi factory presenti nell’interfaccia IShapeFactory.
* Le classi CircularShapeFactory e RectangularShapeFactory sono classi che implementano l’interfaccia IShapeFactory e sono utilizzate dalla classe ShapeParser. Esse astraggono le diverse conversioni in base alla figura di riferimento.
* Le classi CircularShape e RectangularShape sono classi che implementano l’interfaccia IShape, sono utilizzate per rappresentare le diverse figure presenti. Implementano funzioni per ottenere le dimensioni delle figure e le loro coordinate, ovviamente ognuna in base al tipo di figura di riferimento.
* La classe Robot rappresenta un singolo robot all’interno della simulazione. Si occupa di gestire il suo stato andando a fornire funzioni per muoversi e segnalare condizioni. Al suo interno ha una copia del programma da eseguire.
* La classe Point si occupa di gestire la posizione di ogni elemento nel piano cartesiano.
* La classe Direction estende la classe Point e si occupa di gestire la direzione verso la quale il robot si deve muovere, infatti ammette solo valori compresi tra -1 e 1.
* La classe CoordinatesTranslator è una classe di utilità usata principalmente per l’interfaccia grafica in quanto si occupa di traslare le posizioni cartesiane in quelle di uno schermo di un pc.
* Le interfacce Direc
* Per la rappresentazione delle varie istruzioni è presente un’interfaccia RobotInstruction che verrà implementata da ogni istruzione, sotto forma di classe, che andrà a eseguire la propria logica di esecuzione.
* La classe astratta IterativeInstruction implementa l’interfaccia RobotInstruction e rappresenta quelle istruzioni che sono iterative, così da gestire le condizioni di ogni iterazione, questa classe verrà estesa da ogni istruzione che rappresenta un ciclo.

**4 Istruzioni**