

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 varie classi di test, per assicurarsi che tutto il codice scritto sia corretto. 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.

SwarmRobot è dotato anche di una specie di terminale che permette di seguire il flusso di esecuzione di tutti i robot, così da poter visualizzare in tempo reale quello che stanno facendo i robot. Questo terminale è sviluppato in C# tramite il framework .NET MAUI, che permette di realizzare applicazioni desktop, mobile e web per tutte le piattaforme esistenti.

3 Responsabilità assegnate

- La gestione del flusso di esecuzione e delle varie aree è assegnata alla classe Controller, essa 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.
- Il parsing delle figure e del programma viene assegnato alla classe ParserHandler, che diviene un semplice compilatore che verifica la correttezza sintattica del programma.
- La gestione della lista di istruzioni viene assegnata alla classe Program che avrà un program counter per indicare quale riga il robot deve eseguire, ogni robot ha la propria copia del programma.
- La rappresentazione delle varie figure presenti all'interno della simulazione viene assegnata alle classi CircularShape e RectangularShape che implementano l'interfaccia IShape. Esse implementano funzioni per ottenere le dimensioni e le coordinate delle figure.
- La rappresentazione dello stato di un robot viene assegnata alla classe Robot che si occupa di fornire funzioni per muoversi e segnalare condizioni. Al suo interno ha una propria copia del programma da eseguire.
- 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

Il programma utilizza Gradle e Java, perciò per effettuare una compilazione del programma basta eseguire il comando *gradle build* nel terminale (posizionandosi nella cartella principale del progetto). In seguito per eseguire il programma è sufficiente eseguire il comando *gradle run*.

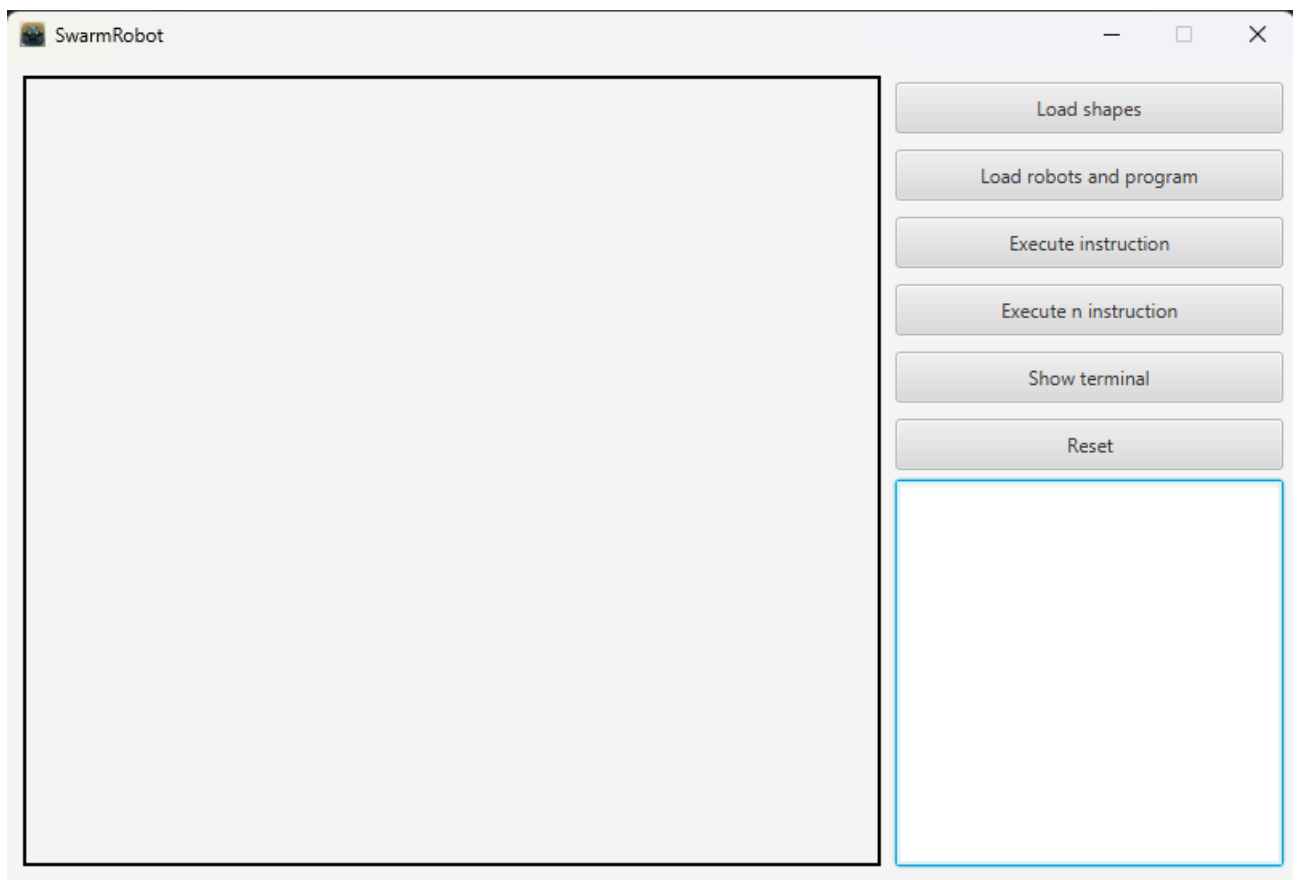
Se nel proprio pc non si è installato gradle, si può usare il gradle wrapper con i seguenti comandi:

- `.\gradlew build`
- `.\gradlew run`

Questi comandi permetteranno l'esecuzione del programma senza dover installare gradle nel proprio pc.

Per quanto riguarda il terminale integrato non c'è bisogno di eseguire nessun'altro comando, ci penserà direttamente SwarmRobot a mandarlo in esecuzione per l'utente.

Una volta eseguito il comando *run* si avrà una schermata di questo tipo:



Da questa situazione l'utente avrà a disposizione diverse azioni da fare:

- Caricare le aree di interesse
- Caricare i robot e il programma

- Eseguire una istruzione (anche se senza aver caricato i robot e il programma non farà nulla)
- Eseguire n istruzioni (vale lo stesso anche qui)
- Mostrare il terminale
- Reimpostare la situazione iniziale

Le aree e i il programma andranno caricati tramite file (nel progetto sono già presenti 2 file di esempio) con estensioni .rshape e .rprogram, mentre i robot possono essere generati casualmente.

Quando si apre il terminale si avrà una situazione del genere:

