Lab ISS | boundaryWalk using BasicStepRobotActor in kotlin

Introduction

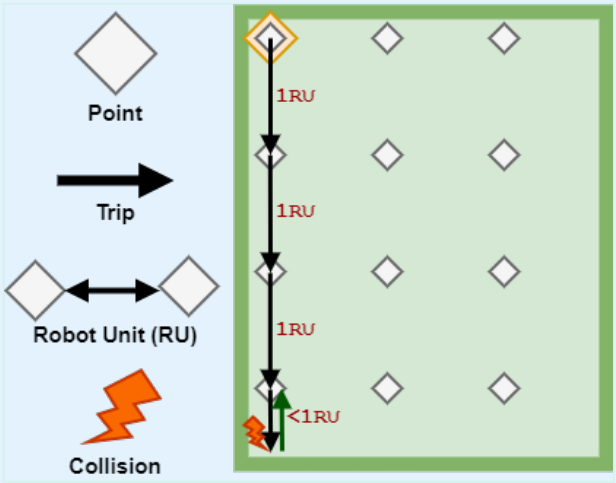
Requirements

Design and build the *BoundaryWalk.html* using *BasicStepRobotActor*.

Requirement analysis

The requirements are the same analized in *BoundaryWalk.html* with only one new that consists into the use of *BasicStepRobotActor* with Kotlin language to build the system. This actor has a large explanation in *ActorWithKotlinSupport.html*.

Problem analysis



Using the *BasicStepRobotActor* the robot can move **step-by-step**. Then, we called **Robot Unit (RU)** the space traveled by the robot in a **Time Unit (TU)** from a single step: **all steps have the same duration (1TU).**
It means that in the last step of a *side walking* it is possible that the robot has no possibility to do the entire step because he collides with the **wall** before concluding. In addition to this, by default, *BasicStepRobotActor* move backward the robot when he collides, returning him to last cell (green arrow in the image).
This can be a very important problem if the application is built to measure the boundary of the room: its lenght can only be in **integer RU** (then, an approximation if the real measure of the leght is not a multiple of RU). Fortunately, *StepRobotActor* has a timer that let us to measure the exact length of

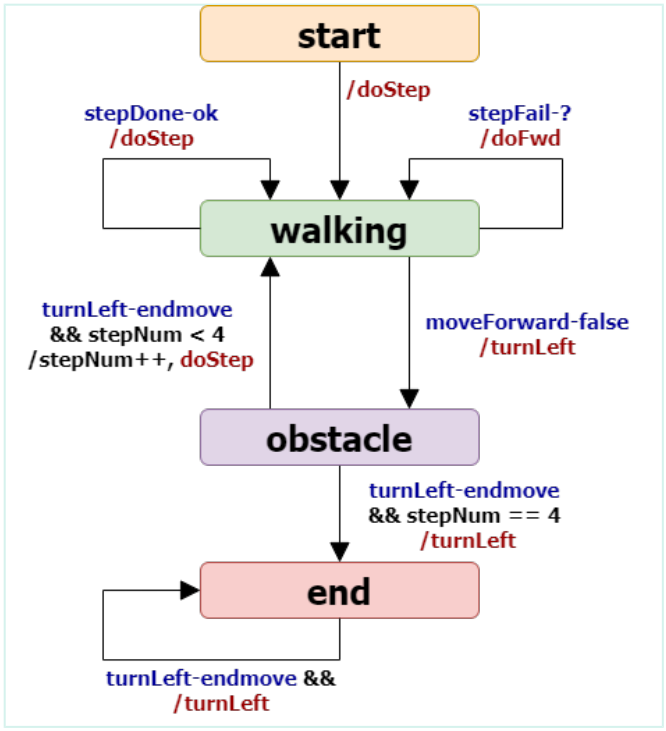
	the last step, so we can measure the boundary in a good way. However, the problem persists if the consumer requires that the robot must walk **exacltly flanking the boundary side** so it is needed to **correct the moveBackward with a moveForward** that place the robot in a correct way.

In order to make the robot walk exactly flanking all the side of the boundary, when he collides and the *BasicStepRobotActor* moves him backward, to send a *moveForward message*, the project-manager shoud directly use a *robotmove command* instead of a *step message* then, when robot newly collides, it is possible to send the *turnLeft command*.

Test plans

Project

To realize the **boundary logic** we will create a new actor *BoundaryWalkerActor* that incapsulates the *StepRobotActor* to move the robot. In addition to this, we can reuse the **FSM** explained in wssupportAsActorJava for the *BoundaryWalk* problem (java class BoundaryWalkerActor.java).



BoundaryWalkerActor.kt:

the new **FSM** is very similar to the one used in Java. However:

- the fsm is activated with an input each time the **handleInput** receives ApplMessages from the **main** to start and also from the *BasicStepRobotActor*;
- the actions related to a state transition are performed when the specified input-message is received and the conditions are true;
- it's been added a new functionality that measure the length of the entire boundary.

Testing

Deployment

Maintenance

By student
Name: Luca Marchegiani
Email: luca.marchegiani3@studio.unibo.it
Git Repo: https://github.com/LM-96/MarchegianiLuca