**CS321**

**Robotic Simulation**

**Sakshi Gupta**

**Abstract**

In this project, I am going to present a two-dimensional mobile robot that has been written using Java. The Robot will be tasked to perform a navigation which will be based on the use of sensors. Therefore, will compile the sensor range and the angle so that it can be adjusted in real-time. I will also implement the Obstacle detection and avoidance algorithm in Java. The high number of growing of the application of the autonomous mobile robots hence the need for dedicated simulators. I am also going to use the computer tools to tests my algorithm. The simulation of this robot is such that it will move on a square tabletop which has dimensions of 5 units x 5 units. In the table, I am going to place no other obstruction on the table surface. In this project, I am going to perform tests that are focused on the reduction of time and cutting down the number of issues.

**The Autonomous Mobile Robot Simulator**

In this project, I am going to present the 2D autonomous mobile robot simulator that was written using the Java Language. In the simulator we will need a standard JDK (Java Development Kit 1.6.0 or a higher version). This toolkit will be able to run on any platform which provides this component for either Windows or Mac OS. The main objective that this Simulator needs to fulfill is the ability to multi-task. This was done using the java.util. concurrent packages. This is a package that will support the concurrent programming.

**The Features Implemented in the Java Simulator**

The Java simulators has the following features that makes it provide its functionality.

* It’s a single robot simulation algorithm code
* It has both moving and static obstacles.
* It has detectors for Collision and obstacle.
* It has simulation for the multi-sensors.
* It has the path drawing for the robot.
* It has an easily extensible graphical user interface.

During the design and coding stage we will make the assumptions that the simulator will be able to simulate a single differential drive mobile robot. The graphic user interface is flexible to allow them to edit the robot parameters such as the linear velocity and the platform dimensions. We also have the Obstacle detection sensors that can be adjusted in a wide range. We also have the sensor range and the radiation cone. When talking about its flexibility, the obstacle position and its orientation is not fixed and can be changed during the simulation.

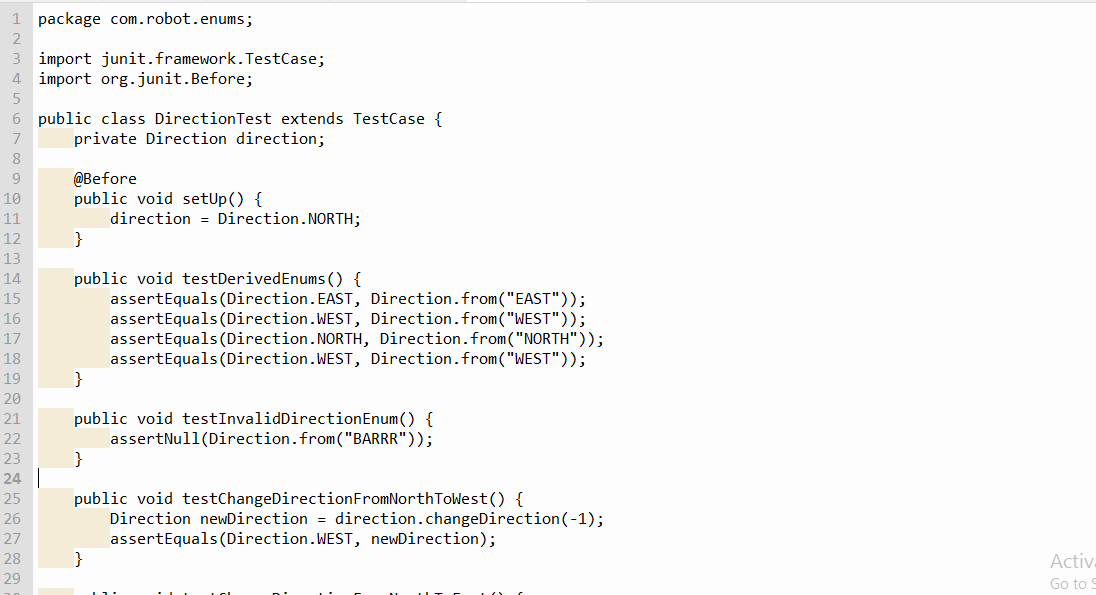
**Collision detection**

This java program that is implemented is designed to recognize only two types of interactions between the obstacle and the robot. For instance, for the Obstacle detection, we have the Infrared sensor. For this every sensor has a visible radiation cone. If the radiation cones are intersected by the obstacle surfaces, then the intersection points are circled.

**Simulations**

During the simulation the robot was free to roam around the surface of the table and it was also prevented from falling to destruction or obstruction. We used the valid movement commands which includes:

* PLACE X, Y , F
* Move
* Left
* Right



**Conclusion**

From this research we have seen the use of the Mobile robot simulator that has the fast control algorithm. The robot had sensors as said earlier on. There simulation was done very fast, and it worked at the best level. The high number of growing of the application of the autonomous mobile robots hence the need for dedicated simulators. I also used the computer tools to tests our algorithm. The simulation of this robot is such that it moved on a square tabletop which has dimensions of 5 units x 5 units. In the table, I placed no other obstruction on the table surface. In this project I performed tests that are focused on the reduction of time and cutting down the number of issues.