FINAL PROJECT -- MILESTONE 5B

=====================

**Introduction:**

- Name of Assignment: Final Project Milestone 5B – Map a Simple Maze

- Lab team number: 5

- Team Members:

+ Phuoc Nguyen, Khoa Tran (NXT Part)

+ Corey Short, Trevor Davenport (PC Part)

- Approximate number of person hours spent on this: 5 hours

**New Classes and Roles:**

For this Milestone 5-B, we implement 2 more values for the enum Message which are *WALL* and *SEND\_MAP*.

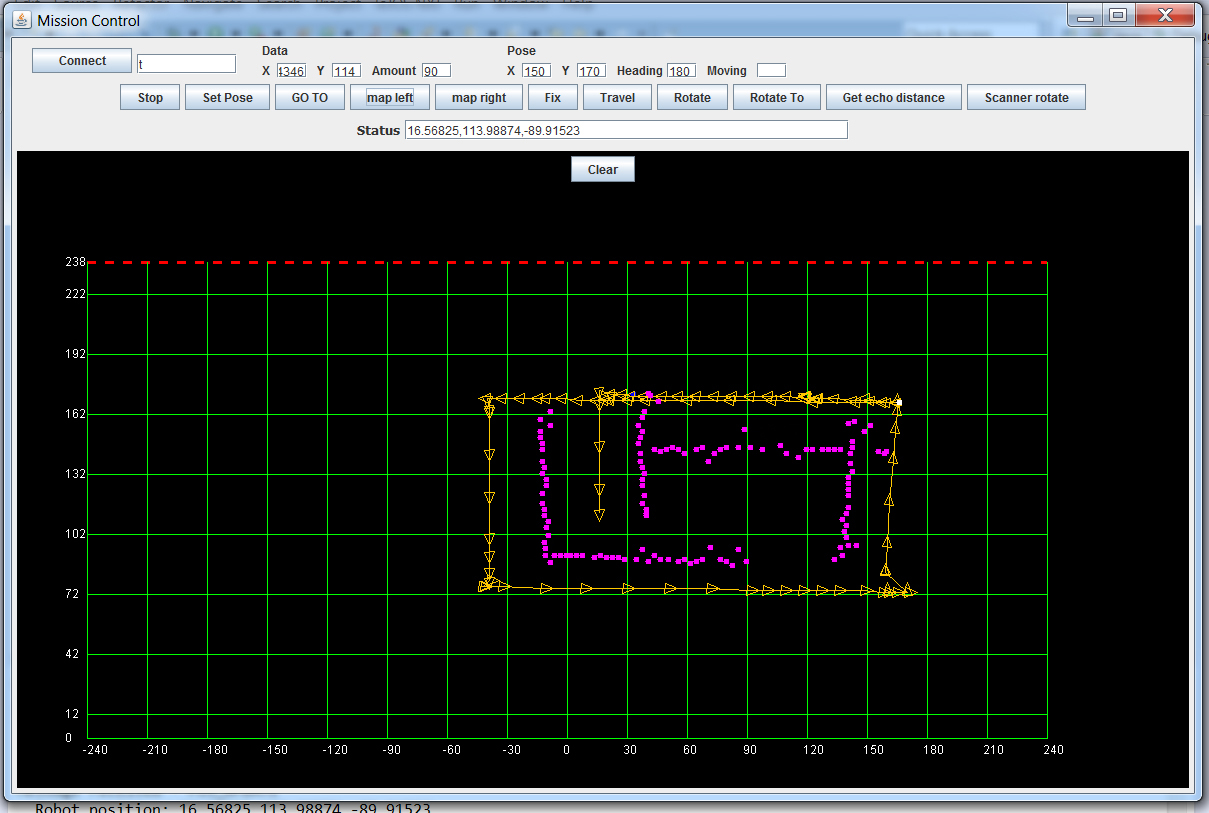
\* In the PC side: sendMapLeft(), sendMapRight(), drawWall()

* When send\_map() is executed, it will send out the x and y coordinate to the NXT brick. Also, if Map Left is called, the PC also send out 90 degree which is the angle that the NXT should rotate to in order to carry out the action. Otherwise, if Map Right is called, -90 degree will be sent instead. We have 2 functions sendMapLeft() and sendMapRight() to handle this task. Also, the enum SEND\_MAP is sent to the NXT for both sendMapLeft() and sendMapRight(). Two different enums are not needed because we send +90 to mapLeft and -90 to mapRight. The robot need only know that a map message has been sent to it such that it can handle the message accordingly.
* After the NXT processes the message, it sends back POS\_UPDATE and WALL to the PC. Those 2 messages are sent continuously when the NXT is running, that's how the PC knows where and when to update its location and the location of the wall. In the PC part, we use DrawObstacle method to draw the location of the wall and update the location of the robot by drawing line.

\* In the NXT side: sendData(), sendWall()

- When the SEND\_MAP command is received, the scanner is first being rotated to the desire angle 90 or -90. We change the method sendData() so it will take in 2 boolean values to determine whether the robot should send back the wall location or not. If the command is SET\_POSE then the wall location will not be sent. Otherwise, if SEND\_MAP is called then the sendData() will get the echoDistance from the scanner and send back the location of the wall to the PC by using the new function sendWall(). Inside the sendWall() method, based on the current Pose from the navigator and the angle from the current getTachoCount(), we set the current Pose by using PointAt(obstacleDistance , angle + pose.getHeading()), then we send back the x and y from the current pose that we just update to the PC side.

**Picture of the Map:**

****