Project 2 Milestone 5

1. Introduction:

In actual warehouses and factories, Robots similiar to ours are nearly autonomous. They receive instructions about where to go and what to do from a computer with a human supervising the actions. In this Milestone, we are going to use Bluetooth to communicate between the NXT Brick and PC.

1.1. Name of the assignment, due date:

Grid Navigation – Project 2 Milestone 5 - October 24, 2013

1.2. Team 5 Members:

Phuoc Nguyen, Khoa Tran: Coders Corey Short, Trevor Davenport: Reporters

1.3. Approximate number of person hours spent on the project :

```
approximately 8 hours.
```

1.4. Location of your project code so I can read the JavaDocs:

```
Team 5/Project 2 – Milestone 5/
```

1. Which of the performance specifications you robot meets (basic and extra credit):

   ```
   Our robot meets all of the necessary performance specifications. Shown below:
   Same as Grid navigation Milestone 4, Shortest path.  Your robot will start at (0,0),
   heading 0. When the robot stops at its destination,  enter the new destination from
   the PC keyboard, or by a mouse click on the grid (after you program the PC side to do this).
   The robot should make a distinctive sound when it transmits a message and  another when a
   message is received.
   ```

2. Hardware design. Describe briefly the design.

   ```
   Our hardware, similiar to an older Milestone of project 2, uses downward facing light sensors
   to track locations on a coordinate plane. We modified our robot since the last milestone and made
   some necessary changes in order to complete this checkpoint. We removed the unnecessary bumper and
   respective sensors. However, we still need to use the ultra sonic sensor in order to detect the
   obstacle.
   ```

3. Experimental work (if any):

   ```
   Much of our experimental work came from testing our code thoroughly. Moreover, setting up Bluetooth
   was a bit of a hinder for our team, but Corey and Khoa managed to make it possible on their laptops.
   ```

```
We also had to determine the distance of objects for our ultrasonicsensor. Also, because we used the
code written by professor Glassy so we have to experience all of the methods and test them out
before we can use them correctly.
```

4.1. Experiment description and purpose:

```
The purpose of this milestone is to demonstrate the abilities of our rover to communicate over Bluetooth.
Also, we need to work on GUI/Frame and mostly the Mouse events.
```

4.2. Calculations and analysis:

```
Most of our calculations came in using a spin-off of Dijkstras Algorithm to determine the
nearest coodinates and shortest paths.
For the mouse click coordinates on the screen, we measure the length of each side of the square
and the distance between each edges of the grid to the possible edges of the JFrame. Based on
those information we are able to calculate the accurate position of the mouse click (pixel)
then covert them into our (x, y) coordinate.
```

4.3. How results were used in our code:

```
After mapping out how to solve the problems on paper, Khoa and Phuoc chose to implement the
code appropriately.
```

1. Software design – top level description of how your code works: what each class does, how classes collaborate, how data flows between classes. This information should be in class header comments.

```
1.    Manage user interaction with screen, mouse and keyboard
2.    Establish Bluetooth connection  to the robot.
3.    Transmit the destination coordinates to the robot.
4.    Receive data from the robot and draw robot path and obstacles
5.    Display these on the grid.
```

Classes:

```
1. GNC -- Is an interface that with four methods (setMessage, drawRobotPath,
        drawObstacle, and incomingMessage)

2. BTSend -- A class that is used to send the Bluetooth signal.

3. GridControlCommunicator -- Helps to establish the Bluetooth connection, reads in the data
input stream, drawsRobotPath and robotObstacle.

4. GridNavController -- This class is our GUI setup for the communication and navigation
instructions. GridNavController implements the interface GNC and provides a graphic
interface to connect our robot.

5. OffScreenGrid -- The OffScreenGrid manages the drawing of the grid and robot path on an Image.
This was provided by Glassey. We also modified this class in order to calculate the coordinate
based on the mouseClick event, and the function is called findNearestCoordinate(), then based
on the coordinate that we have, we update the Xfield and YField, or result in ERROR if the
coordinate is out of bound.

6. BTShortestPath -- Due to our Milestone 4 being correct, we were able to simply use the previous
code and extend it to work with the current milestone.
```

5.1. Sources of programming ideas and also classes that are used but not coded or modified for this lab (except NXJ classes).

```
Much of the skeleton code was provided by Professor Glassey himself, we simply modified and
corrected what needed to be changed in order to work with our previous milestone 4. We used
our previous ShortestPath code and simply extended it to work with this milestone.
```

5.2. For complex classes, diagram the flow of data and flow of control among methods within the class,explain how methods correspond to subtasks. The method – subtask link can be contained in 5.1 instead.

```
    http://parker.ieor.berkeley.edu/ieor140/labs/GridNav5TeleRobot_files/image001.jpg
```

5.3. Link to the Documentation generated from JavaDoc comments within the code as generated by Eclipse

```
    The javadocs is located in the same folder that we put our project in.
```

1. The most interesting/challenging/difficult parts of the project

   ```
   The most interesting/challenging part of this project is to learn how to use bluetooth in order to
   communicate between NXJ Brick and the PC. It took us an hour in order to figure out that we have to
   flush() the datastream to update/get the value. Otherwise, we just get the NULL value. Moreover,
   most of the group's coders are using UBUNTU so it took us another hour in order to part in the
   NXT Bluetooth library and the Bluecove library to make it work.
   ```

2. Link to the source code:

   ```
   Steam Folder / TEAM / Team5 / Project 2 – Milestone 5/
   PC: https://github.com/IEOR140-T5/Project2-PC/
   NXT: https://github.com/IEOR140-T5/Project2-NXT/
   ```