

IEOR 140 Final Project Milestone 4 - 11/13/2012

Team 4: Nate Bailey and Raymond Ma

Responsibilities

In this project, Nate was in charge of program design and coding. Raymond was in charge of hardware design, experimental work, and project writing.

Hours Spent

Approximately 20 hours of work

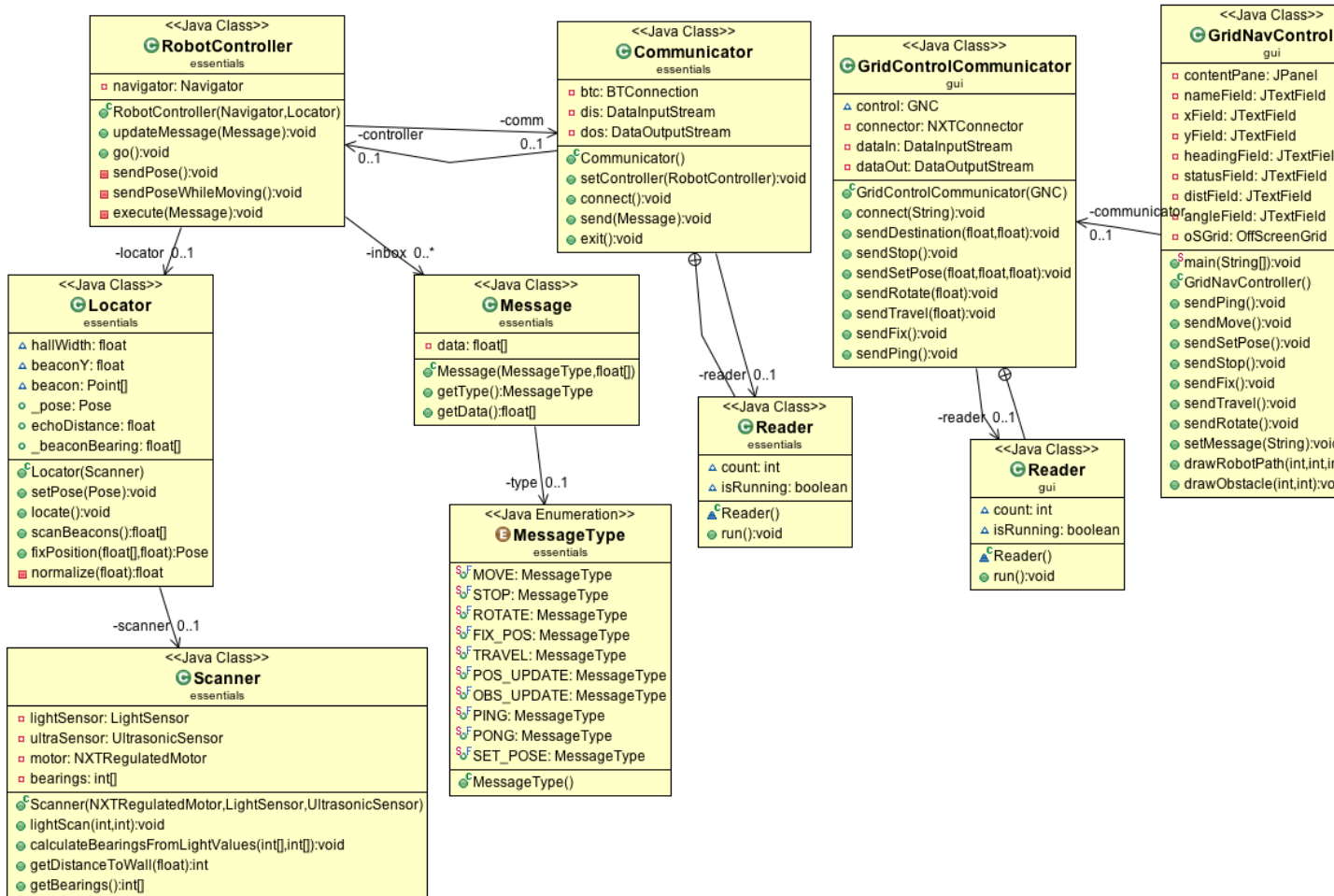
Project Code

<https://github.com/ieor140-team4/FinalProject>

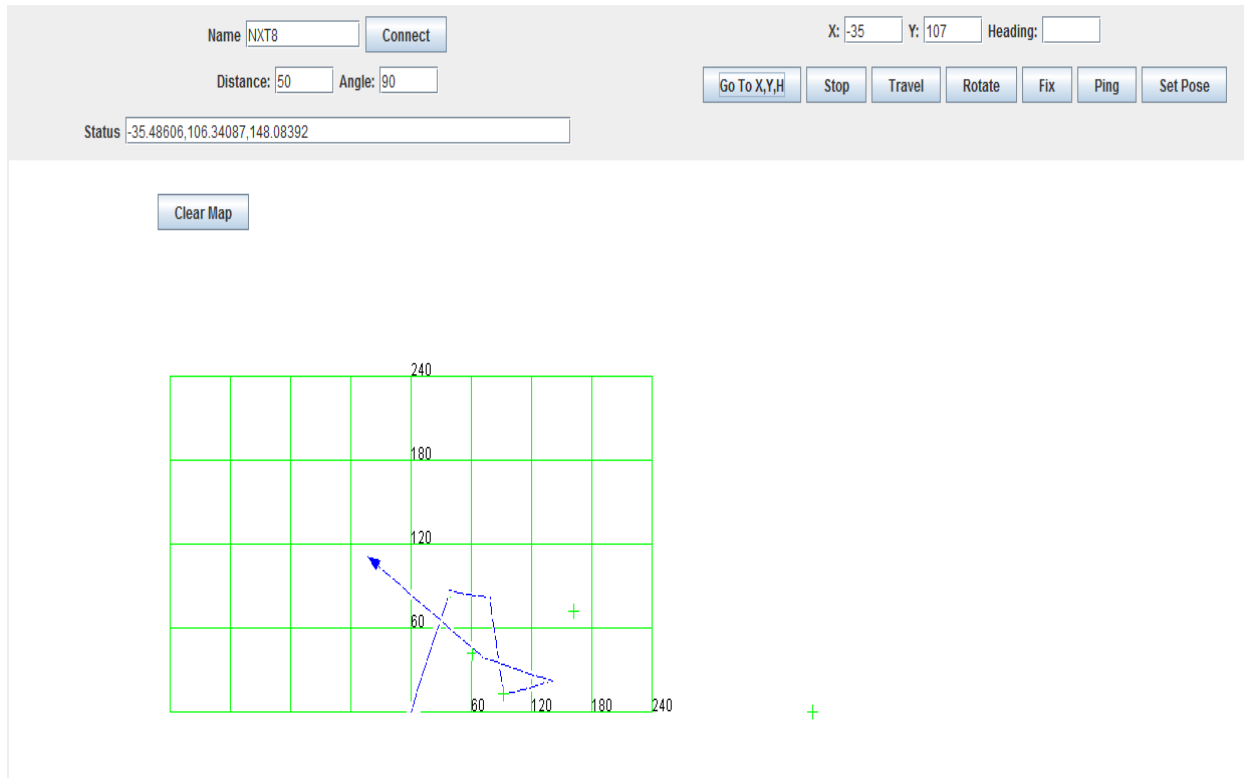
Performance Specifications

Our robot met all of the performance specifications (there were no bonus specifications to meet).

Object Collaboration Diagram



Screenshots of Program



Task Analysis

- Communicator receives a message from the computer over bluetooth
- Communicator then adds message to the RobotController's inbox
- RobotController will pull the oldest message from the inbox, run that command, and then deletes the message
- If a stop command is issued, the inbox is cleared and the stop command is run immediately with all waypoints cleared

Class Responsibilities

The Communicator class is in charge of receiving messages from the computer over bluetooth and putting them in the RobotController's inbox. The RobotController is tasked with reading the oldest inbox message and executing the message at which point it will clear the task from the inbox. Depending on the message type, the RobotController will send the command to a different class. In the case of a MOVE command, the RobotController will send x and y to the Navigator to move to the given coordinates. A TRAVEL command will give the move distance to the Differential Pilot in Navigator which will move the robot forward the given distance. A ROTATE command will send the desired number of degrees to turn to the Differential Pilot in Navigator once again. A FIX_POS command will activate the Locator which will use the Scanner to determine its current position from the two lights. The SET_POSE manually sets the pose in both Locator and Navigator. The special case of a stop command will clear the inbox of all messages and run the stop command (which also clears all waypoints).

Data Flow Between the Robot and PC

The data flow between the robot and PC is accomplished through a bluetooth connection. On the PC, using a Java GUI interface, commands are given from GridNavController to GridControlCommunicator. GridControlCommunicator then has an open connection to Communicator on the robot. Communicator receives the message, parses it, and passes it to RobotController. For messages send in reverse from the robot to the PC, it goes in the opposite direction from RobotController to Communicator, which then passes it to GridControlCommunicator and then to GridNavController.

Interesting/Challenging/Difficult

The most interesting part of this milestone was designing a super beautiful GUI. The most difficult and challenging parts of the milestone was finding bugs in our program.

Appendix

[Source Code](#) | [Java Docs](#)