IEOR 140 Project 2 Milestone 5 - 10/18/2012

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, coding, and project writing.

Hours Spent

Approximately 6 hours of work

Project Code

https://github.com/ieor140-team4/Project2-good (robot)
https://github.com/ieor140-team4/Project2-comp (computer)

Performance Specifications

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

Hardware Design

Our hardware design consisted of the default GridNavigation robot. We had two touch sensors pointed at the ground in order to track both sides of a line and an ultrasonic sensor pointing forward in order to detect blocked nodes. We didn't use touch sensors for this robot. The robot was a two wheel with a castor wheel design with the center axle directly below the middle of our robot.

Experimental Work

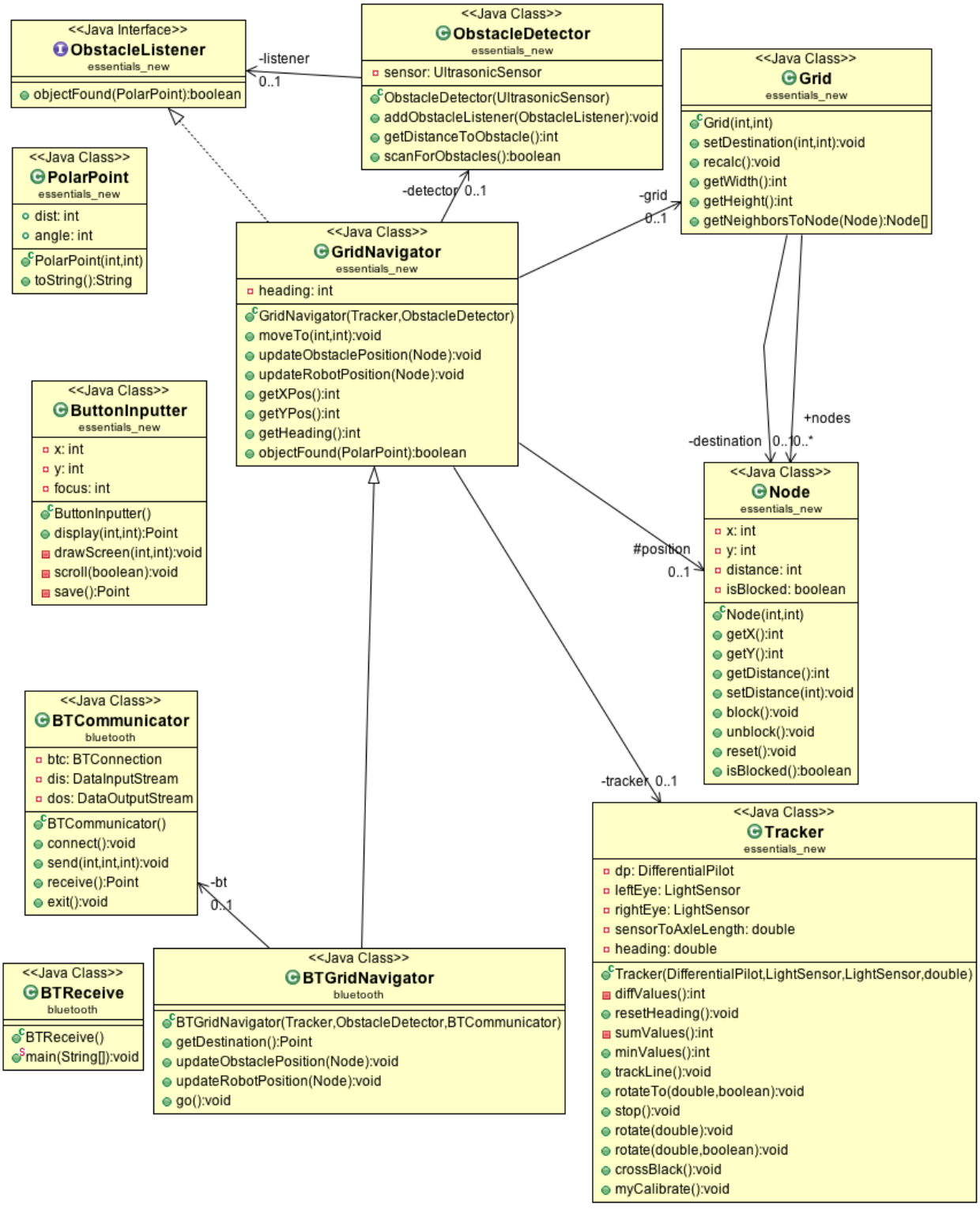
For milestone 4, we found consistent readings when one intersection away from the obstacles with distances of around 24-26, and semi-inconsistent readings when two intersections away. When we were two intersections away, if we saw the obstacle at all, we saw distances of 52-56. Using this information, we used a threshold of 30 and only cared if the obstacle was one unit away. After milestone 4, we also added implementation to detect obstacles further away. We used distances between 30 and 60 to see obstacles 2 intersections away, and distances between 60 and 100 to see obstacles 3 intersections away. For milestone 5, we had no experimental work conducted as all we did was implement bluetooth connection and a GUI.

Problem Analysis

1. Finding obstacles - ObstacleDetector finds obstacles then tells the GridNavigator. GridNavigator then calls the Grid to recalculate the shortest path.
2. Finding the shortest path - GridNavigator tells Grid and then Grid uses Dijkstra shortest path algorithm.
3. Moving - Grid Navigator calls Tracker will then initiates a trackline() in order to move along the grid.
4. Bluetooth Connection - connection between the computer and the robot to pass destination information to the robot and return obstacle information and current location to the computer. BTGridNavigator serves as the interface between the bluetooth connection and moving.
5. GUI - graphical display which allows for easy visual access to Bluetooth connection, destination entering, and a visual representation of the grid which displays information regarding blocked nodes and the robot's current location.

Software Design

- ObstacleDetector - detects objects and passes information on to the GridNavigator which can then call Grid to calculate the new shortest path.
- Grid - calculates the shortest path with obstacle information from ObstacleDetector passed through GridNavigator.
- GridNavigator - updated to use Grid and Node instead of (x,y) coordinate and is now also an object listener to the ObstacleDetector so it knows where obstacles are and how to avoid them. Also contains the method used to navigate around the grid to specified coordinates, by using Djikstra's Shortest Path Algorithm. (Which is contained in a Grid object that GridNavigator has access to.)
- BTGridNavigator - A subclass of GridNavigator that serves as the interface between the new BTCommunicator on the robot that receives the point from BTCommunicator and uses it to move the robot.
- BTCommunicator - controls the bluetooth connection to the computer and receives a point to pass to BTGridNavigator and also transmits blocked node information and the robot's location



Interesting/Challenging/Difficult

The most interesting, difficult, and challenging part of this milestone was coding the bluetooth connection to allow for the robot to receive and pass information between the computer and

itself. Also, creating a GUI that could register clicks and convert them into x,y coordinates on the displayed grid was a very interesting concept, although most of it was already done for us in the project starters.

Appendix
Source Code (robot) | Source Code (computer)
Java Docs (bluetooth) | Java Docs (robot) | Java Docs (computer)