IEOR 140 Project 2 Milestone 4 - 10/11/2012

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 and project writing.

Hours Spent

Approximately 6 hours of work

Experimental Work

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.

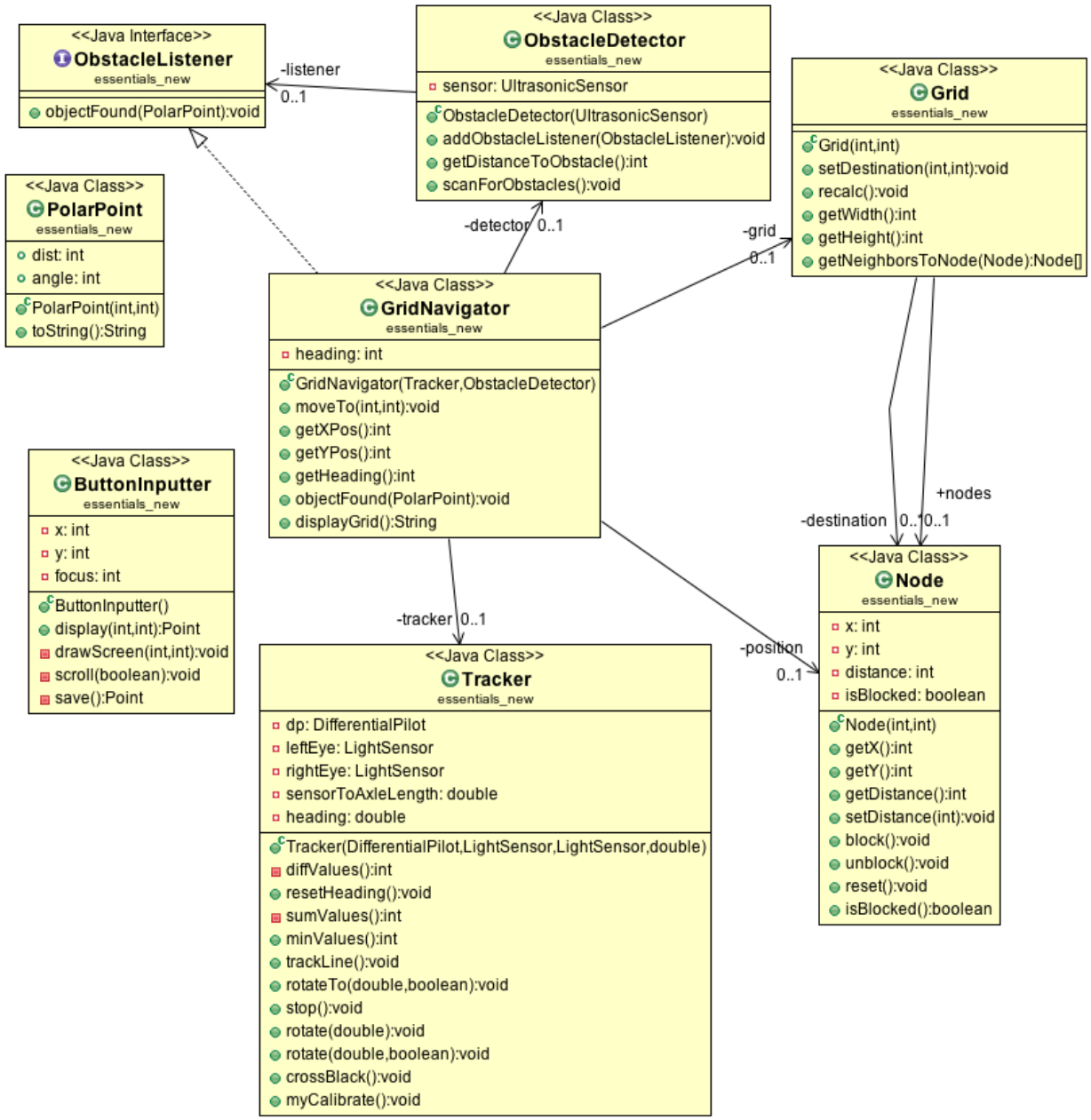
Task Analysis

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.

Class Responsibilities

- 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 t



Interesting/Challenging/Difficult

The most interesting part of this milestone was coding the shortest path algorithm so that it worked in the most efficient way. The most challenging and difficult part of the milestone was making sure that our checks for obstacles would update the shortest path in the right order in order to avoid multiple obstacles as we navigated the grid.

Appendix

Source Code | Java Docs