Juan Benitez, Sigi Lopez
CSE 180 Robotics
Professor Carpin
05/04/18

Final Project Report

The final project for robotics consisted of searching for treasures inside an enclosed area. This area had an unknown number of obstacles which tested how our autonomous robot would react. Our main strategy consisted of four nodes, a planer, explore, obstacle avoidance and camera. All of these nodes worked in parallel to create a self-automated procedure to complete the search for the entire space. The goal was to find the most treasures and report them with their name and as coordinates based on the global map. The main goal of this assignment was to test us on most of the material covered through the whole semester.

Our planer node was our core for finding out where the robot was going to search. The implementation that we created was based on the boundaries of the given obstacle course which was 20x20 meters. Since we didn't know how the treasures were going to be distributed we decided to split the map based on a grid. This is very important on how the robot planned its path, the smaller the squares, the higher the chance the robot would locate more treasures. The increase in squares to traverse substantially increases the chance of covering more space on the map. We suddenly noticed that this implementation also brought a downside which was that the more squares the slower the robot would take to cover the whole area on the map. The implementation of making the grid was very mathematically simple we just created a data structure that would store all the coordinates of each upper left corner. We had one subscriber for this node which subscribed to a topic called exploreStatus. This subscription was responsible for checking when the explore node needed another coordinate. The planer node also had a publisher which published the necessary coordinates for that time as pose geometry messages. Overall the planer node was the core for the whole implementation since it dealt with such an important task.

The explore node focused mainly on moving the robot with an action server called move base. This node had one publisher which posted a Boolean that essentially asked the planer node for the next coordinates. In addition, the node contains one subscriber for receiving the coordinates from planner. The node also subscribes to a topic published by the obstacle avoidance node. This topic is responsible for stopping a goal if it's necessary if the robot gets conflicted with an obstacle. We decided to use move base as our main navigation system since it uses different types of sensors to locate itself. This is an advantage over just posting twist messages to the robot because move base tends to have less miscalculations. On the other hand, we couldn't just depend on move base because the longer it runs the more inaccurate it is from its true position. This is why we also included an obstacle avoidance node that uses laser feedback and posts geometry twist messages to the robot. The most complicated part arose when the robot could no longer rely on move base for its movement controls. We did not expect at first, canceling a goal was going to get really complicated since we were still trying to make sure the robot reached its target. Unfortunately, we were not successful at implementing both this node and the obstacle avoidance node together. Instead of making the robot avoid troubling obstacles we had to block the node with waiting for move base for a few seconds. In consequence on following this method is that the robot would sometimes get stuck in places where it should not.

The “Obstacle_Avoidance” node detects objects, stops the “explore” node and changes his orientation to avoid objects. Object detection is done by subscribing to the laser and calculating which objects is the nearest to the husky. If the node calculates that an object is closer than 1 meter to the husky, the node will publish true to the topic “/stop” which the “explore” node is subscribed to and halts its exploration. The husky will then begin to rotate in place and using the laser it will calculate when objects are further than 1 meter. When that is the case and the node detects no objects is closer than 1 meter it will send a false to the topic “/stop” indicating to the explore node that it can continue its exploration. This node guarantees that the husky does not collide with any obstacles.

The “camera” node is in charge of detecting, calculating and storing the information of the treasures. In order for the node to identify a treasure it subscribes to the topic “objectsDetected” which receives a custom message type giving us the distance from the treasure to the husky. This information is temporarily stored in a array for calculation purposes. Furthermore we use a transformation from “odom” and “baselink” to calculate the huskys coordinates. With the huskys coordinates and the information stored in the array we can calculate where the treasure is with respect to the map. After the calculations we store the information in a struct called “Treasure”, similar to the custom message type used by the logica_camera_plugin. The “Treasure” is then stored into a vector of type “Treasure” that holds all of the previous treasures it has found. This vector list is constantly displayed as to constantly have feedback of all the treasures it has found.

Our final project result did not fully function as we had expected, we had a slight complication enabling our Object Detection node and as a result our husky got stuck a couple of times. The husky was able to get free itself and continue its operation but we would have hoped it would of never had to spend time trying to free itself. All of the other nodes worked as expected and by the end of the presentation the husky was able to find 3 treasures. This project allowed us to grasp a better understanding about the intricacy of running multiple nodes in parallel. It also gave us a different perspective on programing, to have multiple programs running to achieve a goal, rather than what were used to, having one long program.
After all we have learned a lot from these experiences which we would take into the future for other classes and possible careers.