Code Documentation

Introduction

The software design for the robot was one of the primary tasks for this project. The sub-team responsible for the coding element of the project, called the C-team, consisted of Marty Thompson, the Senior member, and Steven Layton, the Junior member. As always, software design is an ongoing process throughout the entire development of the robot. As there are changes in the hardware aspect of the design, there will also have to be changes to the software.

As was learned from the last project, the Handyboard cannot effectively handle more than three processes, so we produced only two concurrently running processes (excluding the main method). These processes are as follows:

- Look for goal location
- A running process that continually seeks new targets.

These processes will be described in more detail later.

In addition to these processes, a path planning and execution algorithm was widely used by this robot. Using this, the robot was able to seek out goals, determine the nearest target, and plan a path. Once the plan is made, it is executed, and the target is brought back to the goal, assuming it is the closest goal to that particular target.

This software design attempts to work with a world model. The behaviors and algorithms chosen for the software element of this project were chosen to be used in this world model. These behaviors will be further described in the following sections.

Look for Goal Location Process

This process was continuously running as the robot traversed the arena. Its purpose is to immediately make the robot check to see if a target object is within the hopper whenever it runs over a goal location. If the hopper is full, and the process detects a goal location, it proceeds to perform the target drop off procedure. This procedure involves a full stop of the robot, and a reversing of the motors to ensure the target is no longer within the hopper. Once this is done, the robot emits three tones, and waits for the operator to push the start button, which then tells the robot to proceed searching out targets.

Running Process

This process is also continuously running as the robot traverses the arena. Its purpose is to continuously seek out new targets, and to call the path planner and path executor. This process is primarily made up of two consecutive while loops. The first loop is concerned with finding the known targets provided by the instructor, and the second loop tries to find unknown targets and attempts to strike the blue dynamic object within the arena.

Path Planning and Execution

The planner function takes four parameters, the *x* and *y* coordinates of the current location, and the *x* and *y* coordinates of the target location. It proceeds to check two different paths to the target location for obstacles. If one path is clear while the other has obstacles, the planner makes the path for the clear path. If both paths have obstacles in the way, then the path with the least obstacles is chosen. Once a path has been selected, a global array is updated to include the vertices of the path.

Once this array has been updated, the plan execution function is called to go along the chosen path. Using the current location, current direction, and helper functions to turn and go straight, the plan executor is able to traverse the path, and arrive at the target location. Once the target location has been reached, the running process determines the next location to go to, and then a new path is planned and executed.

Miscellaneous Functionality

This robot also had other functionality that made it more robust. For example, one operation that increased the robots effectiveness in traversing the arena is the align function. This function is similar to the alignment function used in Project One. When the robot came to a goal location, it would align itself so that it is following a line perpendicular and parallel to the arena sides.

An additional function that increased the robustness of the robot was the cube alignment function. This function used the CMUcam's ability to track color, in this case orange, to get the orange target cubes into the hopper of the robot. This is important, of course, as the point of the project is to get the target cubes to the goal locations.