Group 10

Amit Maole
Brent Eskridge
Klo Utley
Tony Lopez

PROJECT 3: Software Final Report

For this project, we used a simple software design of a finite state machine to navigate the robot around the arena. The main states of the system involve finding a destination, navigating to it, and retrieving the cubes for the destination. Another state was to be pursuit of the blue robot, but this was not implemented. Both of these states used a simple data set comprising the world model. Each destination and target are tracked along with the robot's current position and orientation.

Upon startup, the robot enters its initialization state. The known locations for the targets and destinations are loaded into the world model. Each item is given a confidence level indicating that the validity of the data is in question. Once each target and destination location is loaded, the nearest destination to each target is calculated and saved. The robot then determines the location of the closest unvisited destination and navigates toward it. After arriving at the destination, it was planned that the robot verify its orientation with the tape marking the destination. However, there was not time to finish this functionality due to persistant navigation problems. The robot then retrieves the list of targets that were determined to be closest to this destination. The initial design called for the robot to verify the existence of each target with the CMUCam, but the code was not completed for this ability. Therefore, it was assumed that the information given was correct. The robot then sets a path for one of the targets and navigated towards it. When the robot senses that it is at the correct position, it turns around and returns to its current destination of interest. The target is then marked as visited in the world model. This process is repeated for each target associated with the current destination. Once all the targets have been retrieved, the destination is marked visited. The robot then repeats the process by finding and navigating to the closest unvisited destination. If there are no more destinations to visit, the robot was to enter the pursuit state. Unfortunately, the functionality for this state was not completed due to time constraints.

The following data structures were used in the software:

Structure: coordinate		
	Represents a location within the arena. The values of -1.0 indicate a non-existent location.	
float	x	The x value of a Cartesian coordinate
float	y	The y value of a Cartesian coordinate

Structure: path		
	Represents one element or leg of a calculated path.	
float	distance	The distance to travel along this path element
int	direction	The direction to travel along this path element

Structure: destInfo		
	Represents all of the information about a destination in the world model.	
coordinate	position	The position of the destination
int	confidence	The confidence in the destination's position
int	visitedFlag	Flag indicating whether or not this destination has been visited

Structure: targetInfo		
	Represents all of the information about a target in the world model.	
coordinate	position	The position of the target
int	confidence	The confidence in the target's position
int	closestDestIdx	The index of the closest destination
int	retrievedFlag	Flag indicating whether or not this target has been retrieved

Structure: destDistance		
	Used for sorting destinations based on their distance from a specified location.	
int	destIdx	The index of the associated destination
float	distance	The distance to the associated destination

Structure: targetDistance		
	Used for sorting targets based on their distance from a specified location.	
int	targetIdx	The index of the associated target
float	distance	The distance to the associated target