



Final Project
Navigation Competencies
UnFrozen Challenge “Let It Go”



Lectures: 5-2 through 7-2

Reading: Chapter 4, *Mobile Robotics for Multidisciplinary Study*

Purpose: The purpose of the final project is for the student team to demonstrate the integration of several concepts learned this quarter. The primary goal is to develop several navigation competencies on the CEENBot. The first task is to use topological and metric navigation to move the mobile robot from a start point to a goal point in the world. The localization task involves using sensor feedback with an exploration routine to determine the location of a lost robot in the world. The final task is to create a topological or metric map of the robot’s world by using sensor feedback with a coverage algorithm. The students will demonstrate some of the navigation routines during the final project competition which involves using a search algorithm to determine the location of a heat or light source in the world, thus the name, “UnFrozen”.

Objectives: At the conclusion of this project, the student should be able to:

- Implement topological path planning on a mobile robot to move the robot from a start point to a goal location
- Implement metric path planning on a mobile robot by using wavefront propagation or grassfire expansion to move the robot from a start to a goal location
- Use sensor data and an exploration algorithm to localize a robot in a world given an a priori map
- Use sensor data and a coverage algorithm to create an occupancy grid or topological map of the robot’s world
- Create a world model that the robot can use for navigation
- Use a search or exploration algorithm to locate a heat and/or light beacon in a world

Equipment: Base Robot
Range and Contact sensors
Temperature Array
Photoresistors

Software: CEENBoT API static library
AVR Studio 6.0
WinAVR GCC toolchain



Theory:

Part Ia - Topological Path Planning and Execution

Topological path planning is based upon landmarks in the world. A distinctive place is a landmark where the robot can make a navigation decision. Typically the robot will use one behavior to move in the world and then change when it gets close or in the neighborhood of the distinctive place. For the purpose of this lab the distinctive places in the world will be hallways, corners, dead-ends and t-junctions. The student team will design behaviors and perceptual schema for identifying gateways in an artificial environment. The robot will be given a list of navigation commands based upon the world's topology and use a parsing routine and a sequencer to move the robot from a start point to a goal point. For example, if the robot is given "SLRT" (S = Start, L = go Left, R = go Right, T = Terminate) then it would start, follow hallway, turn left at the next gateway, turn right at the next gateway, then stop at the last gateway. The robot should continue to move forward until a gateway is encountered or until the stop command is encountered. In order to extend the robot's topological path planning capabilities, the navigation commands could be modified to include numerical information such as "3L" (make the third left") or "1L" (make the first available left). In addition, if the robot has a left command next but the first available gateway is on the right then it should continue to following the hallway until it can make a left. This would mean that the robot must keep track of the type of and number of gateways encountered. Figure 1 is an example of topological navigation using the command "SRLT".

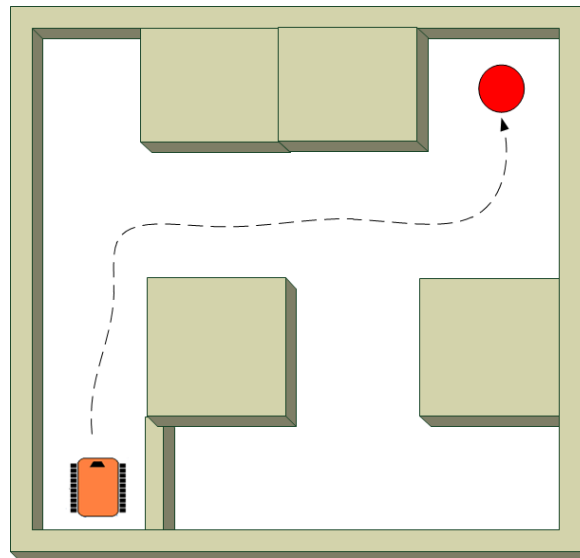
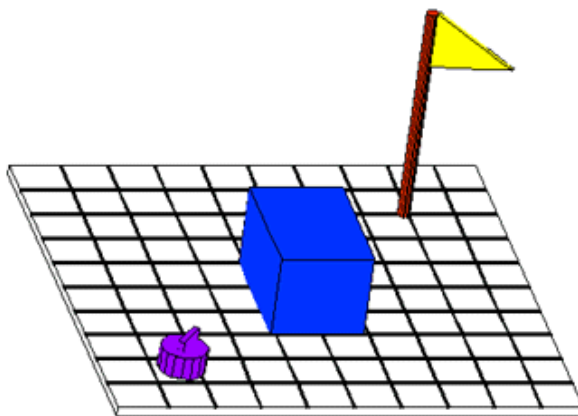


Figure 1: Topological Navigation ("SRLT")



Part Ib - Metric Map Path Planning and Execution

In this lab, you will use a wavefront algorithm on an a priori map to create a path from the robot’s start position to goal location. Use the obstacle avoidance and move to goal behaviors to move through the list of goal points until the robot arrives at the final destination. Assume that the algorithm uses an eight-neighborhood so that the robot can move diagonally. The configuration space will be an occupancy grid divided into 18” x 18” squares, where free space is represented by 0’s and occupied space by 99’s. You should devise a scheme to represent the robot’s start position and goal position. Your code should be flexible enough such that these values can be specified at run time. Figure 2 is an example of the world representation.



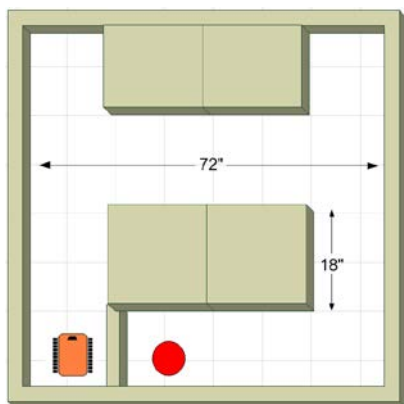
0	0	0	0	0	0	0	0
0	0	0	0	0	0	G	0
0	0	0	0	0	0	0	0
0	0	0	99	99	99	0	0
0	0	0	99	99	99	0	0
0	0	0	99	99	99	0	0
0	0	0	0	0	0	0	0
0	S	0	0	0	0	0	0

a. Real world

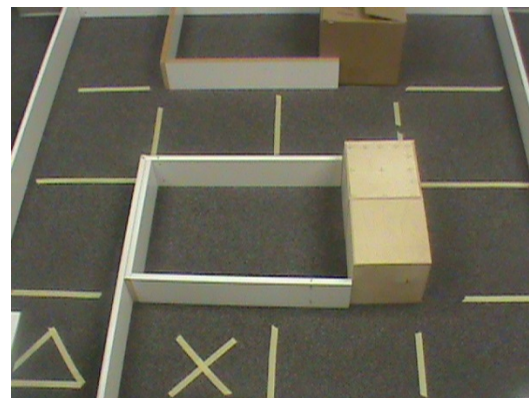
b. Configuration Space (8 x 8 matrix)

Figure 2: World Representation

The test arena for the lab demonstration will be 6 ft x 6 ft with 18” x 18” obstacles. This artificial world or configuration space will be a 4 x 4 grid where the robot’s start point is denoted by a ‘Δ’ and the goal point is marked by an “X”. Figure 3 shows a sample test arena but it may not necessarily be the world you will have.



a. Artificial world



b. Real world

Figure 3: Test Arena



The wavefront is created by starting at the destination and creating eight connected neighbors back to the start point. The numbers represent the number of steps to the goal point. The robot would then follow the numbers in the reverse order to arrive at the goal point (see Figure 4). The goal is for the robot to always move such that the steps to the goal position are reduced. This path plan has 9 steps from start to finish. Note that you may need to grow the obstacles by the robot’s width or radius to avoid clipping them.

7	6	5	4	3	2	1	1	1
7	6	5	4	3	2	1	0	1
7	6	5	4	3	2	1	1	1
7	6	5	99	99	99	2	2	2
7	6	6	99	99	99	3	3	3
7	7	7	99	99	99	4	4	4
8	8	8	7	6	5	5	5	5
9	9	8	7	6	6	6	6	6
10	9	8	7	7	7	7	7	7

Figure 4: Test Arena Wavefront

Instead of representing the world map as an occupancy grid, it can be based upon the topology of the space. The salient features of the space are walls, hallways, corners and junctions. Each square will be represented by an integer between 0 and 15, dependent upon where walls are present around the square. The north (0001), east (0010), south (0100) and west (1000) walls represent one bit of that integer (see Table 1).

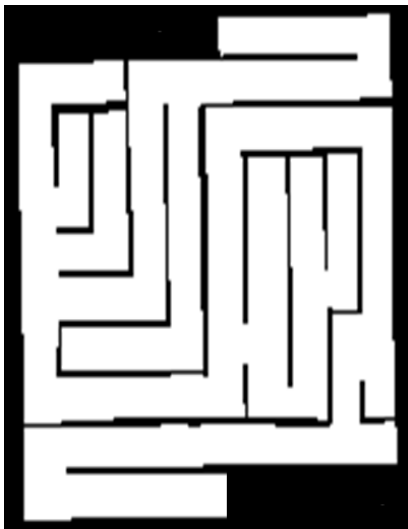
Table 1: Topological map coding

Integer	Binary	Hexadecimal	Direction	Wall Location
0	0000	0		
1	0001	1	North	
2	0010	2	East	
3	0011	3		
4	0100	4	South	
5	0101	5		
6	0110	6		
7	0111	7		



8	1000	8	West	
9	1001	9		
10	1010	a		
11	1011	b		
12	1100	c		
13	1101	d		
14	1110	e		
15	1111	f		

Using the coding in Table 1, the maze shown in Figure 5 is represented by an 11 x 10 matrix of integers.



15	15	15	15	15	13	5	5	5	3
9	5	7	9	1	5	5	5	5	6
10	11	11	10	10	9	5	5	5	3
10	10	10	10	10	10	11	11	11	10
8	6	10	10	10	10	10	10	10	10
8	5	6	10	10	10	10	10	10	10
8	5	5	6	10	10	10	10	6	10
10	13	5	5	6	8	2	10	9	2
12	5	5	5	5	6	12	6	10	14
9	5	5	5	5	5	5	5	4	7
12	5	5	5	5	15	15	15	15	15

Figure 5: Maze Topological Map

To use the topological map to plan a path from a robot start location to a goal point it is possible to use the wavefront algorithm again. However, instead of the robot moving to cells on the occupancy grid, the robot will use behaviors and rules such as move forward, turn left, follow wall, follow hallway, avoid obstacle, etc. The navigation involves taking the list of actions and executing them. During the demonstration, you will be given the map as a 4 x 4 array of integers or in a .txt file that represents the world's salient features.

Part II - Localization

1. The user will provide the robot with an a priori map of the world as a text file. The map will be a 4 x 4 array with topological map or occupancy grid encoding.



- The localization algorithm should process this map to identify key features such as the gateways or distinctive places [corners (C), hallways (H), dead-ends (D), t-junctions (T)]. Although it is not a gateway, a series of 10's or 5's or neighboring 1's and 4's or 8's and 2's indicate a hallway (H) in the world. The numbers (1, 2, 3) indicate a gateway where the robot can make a navigation decision. The distinctive features or gateways are the nodes in the world and the hallways can be used with a local control strategy such as follow center or follow wall to move between nodes. An example of coding a map is shown in Figure 5.

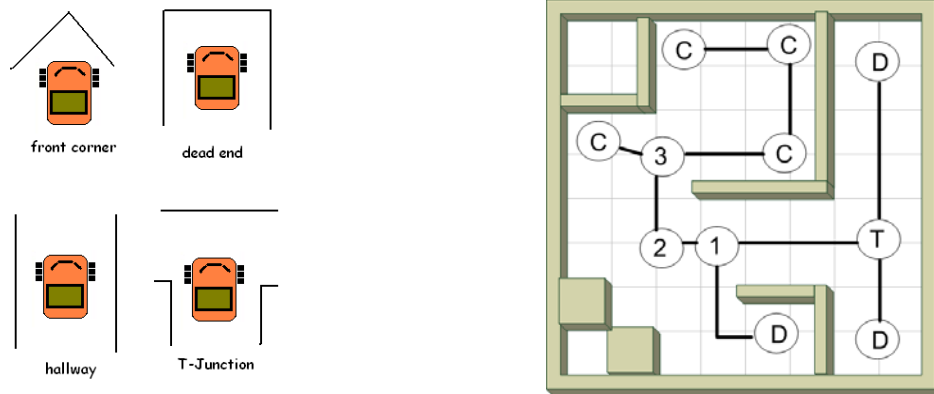


Figure 5: Feature Extraction (Map Coding)

- Once the robot is placed in the world, the robot should then use some motion algorithm such as random wander, follow center, or wall following to explore the world and identify gateways. It should keep track of the gateways passed and the order in which they were passed. Although there will be some odometry error, it would also aid the localization algorithm to keep track of odometry such as distance traveled and turns. Within three to four iterations of this process, the robot should be able to use a probabilistic method such as the Partially Observable Markov Decision Process (PMDP) to localize itself. Figure 6 provides an example of this localization process with the proposed robot locations after each step marked on the map.

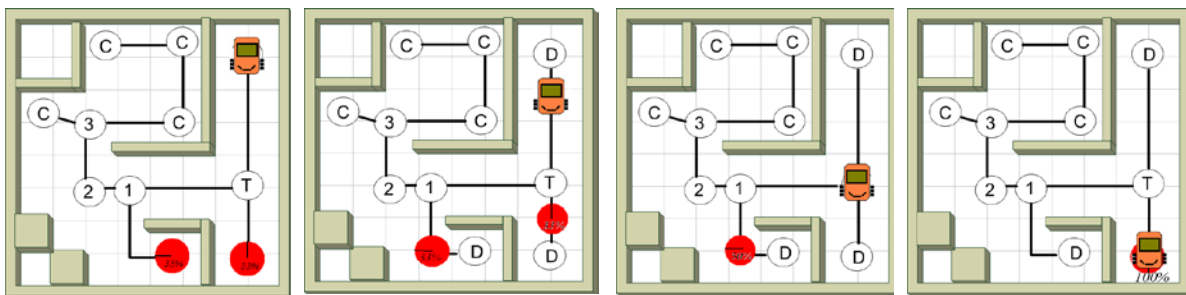


Figure 6: PMDP Localization

- Lastly, after determining its location using registration, the robot should use wavefront propagation to plan a path from its current location to the goal location (or home). The robot's home position and goal location will be specified at run time. After localization, you can use prior code to plan a path for the robot.



Part III - Mapping

It is possible to use Histogrammic in motion mapping (HIMM) to identify objects in the environment. Although your textbook states that this algorithm was created for a sonar sensor, it is reasonable to use it for the primary axis of the IR sensor. For this method when the cell value exceeds some threshold it is coded as occupied or used to create a topological map. You do not have to use this technique you have the flexibility of selecting your own technique to create the model. Alternately, you can create an occupancy grid with 0's and 99's based upon the free and occupied space. Figure 7 provides an example of using a Generalized Voronoi Graph (GVG) and HIMM to create a world map.

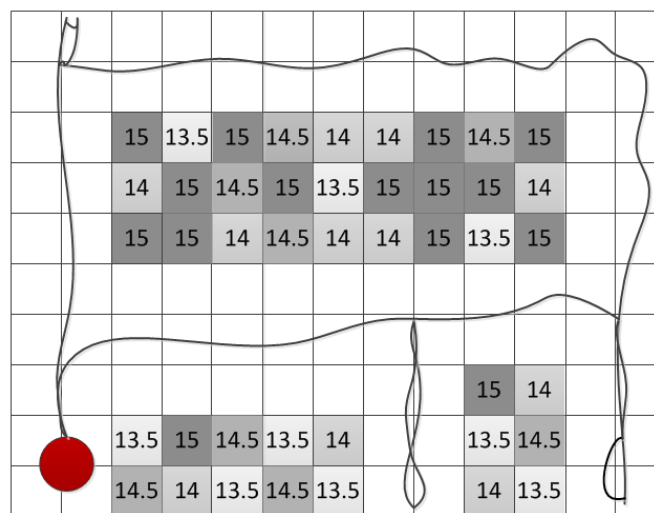


Figure 7: GVG with HIMM

PROCEDURE

PART I – Path Planning

a. Topological Path Planning

1. The student team should place the robot in several corners and intersections of hallways in the artificial environment and determine the perceptual schema to identify these gateways and distinctive places.
2. It would be advisable to use a combination of the IR and contact for sensor redundancy to identify these locations in the world. There will be some sensor error, this is a standard problem in mobile robot navigation and the program should be designed in such a way to minimize the effect of the error. Figure 8 provides descriptions of the possible world landmarks.

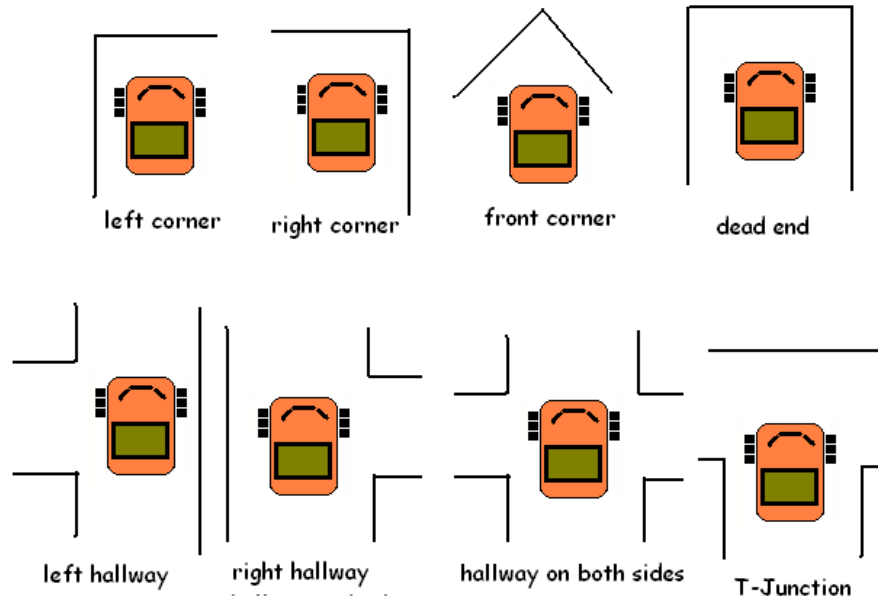


Figure 8: Distinctive Places

3. Include a table similar to Table 2 that includes the test data and perceptual schema for each of the landmarks.
4. Program the robot to identify the gateways and make navigation decisions such as follow hallway, turn left or turn right based upon a topological path plan.
5. Finally, use the robot push buttons to give the robot a command such as “SRLT” and place it in the world so that it can execute the path.

Table 2: Perceptual Schema Data Table

Landmark	Front IR	Left IR	Right IR	Right Contact	Left Contact
Corner in front					
Corner on left					
Corner on right					
Hallway on both sides					
Hallway on left					
Hallway on right					
T-junction					
Dead End					

b. Metric Path Planning with Occupancy Grip

1. Download the occupancy grid world map from the final project folder in Moodle. The world map is a text file that includes a 4 x 4 array of 0’s and 99’s that represents free space and obstacles. You must devise a method to code the a priori map into your program. You can also display the map on the LCD to confirm



that it matches the real world. Download the “PixelPrint.c” program from the Moodle folder to see examples of how to print pixels on the LCD screen.

2. At the beginning of the demonstration, you will be given the robot’s start position and goal point. Create a technique to use a pushbutton sequence to input the start position and goal position to the robot to the robot.
3. The robot should plan a path from the start point to goal position by using the wavefront algorithm. The LCD should display the path planned by the robot. One suggestion for doing this is to show a list of cells that the robot will traverse from the start point to goal point.
4. You should then place your robot at the start position and press start and it should move to the goal point. Your algorithm should include a combination of odometry and reactive behaviors to avoid hitting the walls while executing the planned path.
5. You will be graded on how well your algorithm works; the efficiency of the path chosen by the robot, the ability of the robot to reach the goal point while also avoiding obstacles. *(Note that the robot’s center of rotation is between its wheels not the center of the chassis so you should offset the robot in the starting cell so that the robot’s center of rotation is at the center of the cell. Another technique to prevent the robot from hitting walls and obstacles is to select the path that maximizes the distance between walls and obstacles.)*

c. Metric Path Planning with Topological Map

1. Download the topological map from the final project folder in Moodle. Review the theory section if you don’t remember how the map is coded to represent walls in the world. Use the map to plan a path from a robot start location to a goal point. It is possible to use the wavefront algorithm again however, instead of the robot moving to cells on the occupancy grid, the robot will use behaviors and rules such as move forward, turn left, follow wall, follow hallway, avoid obstacle, etc. The navigation involves taking the list of actions and executing them.
2. You will be graded on how well your algorithm works; the efficiency of the path chosen by the robot, the ability of the robot to reach the goal point while also avoiding obstacles.

Part II – Localization and Search

a. Localization and path planning

1. Download the map from the final project folder on the Moodle course website. You should input this map into your localization program.
2. During the demonstration, you will be given the robot’s start position so that the team knows where to place the robot in the world. The robot can have any orientation.



3. The robot's goal location should be entered using the pushbuttons. The robot should then use the localization algorithm to determine its location in the world and then use topological or metric path planning to move to the goal.
4. You will be graded on how well your algorithm works; the efficiency of the path chosen by the robot, the ability of the robot to reach the goal point while also avoiding obstacles. *(Note that the robot's center of rotation is between its wheels not the center of the chassis so you should offset the robot in the starting cell so that the robot's center of rotation is at the center of the cell. Another technique to prevent the robot from hitting walls and obstacles is to select the path that maximizes the distance between walls and obstacles or the center line.)*

b. Search and Home

1. The robot will be placed at a random start location in the world. This location will be denoted as the robot's 'home' position. The robot should then be able to use a movement algorithm to localize itself with respect to the world map. It should then make an audible or visible signal that it is localized.
2. There will also be a heat/light source placed randomly in the world. The robot's task is to use a search/cover algorithm to find the location of the heat/light source. The robot should make an audible or visible signal to denote that it has located the source such as a beep, light or LCD display.
3. The robot should then plan a path back to the home location and show the location of the source on the LCD. See Figure 9 for an example of the search and home task.

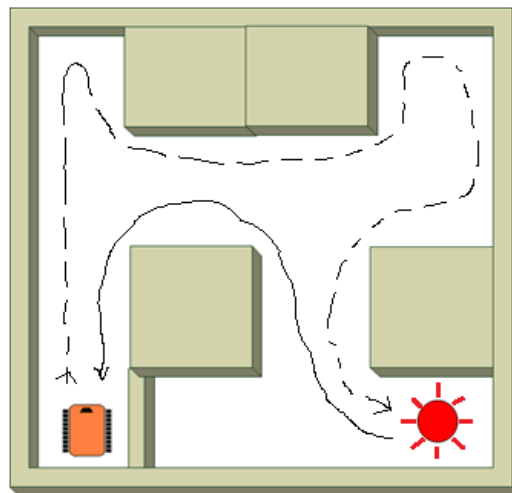


Figure 9: Heat or Light Search Bot

PART III – Map Making

1. In this exercise, you will build a map of the 6' x 6' test arena by using a cover algorithm and sensor feedback. The robot should be able to use the created map to plan a path from a start position to a goal location. Try to



reuse as much code as possible from prior labs or parts once the map has been created to simplify the path planning and execution.

2. Note that mapping will be tricky because of odometry error, sensor error, and other sources of error. A completely accurate map would require localization to reduce the uncertainty of position. However, this would require SLAM which is difficult to implement. Therefore, using only the robot's sensor data and odometry try to create the best world model possible.
3. You can use wander, cover, wall following or any other motion algorithms to move the robot through the environment. One possible technique is to create a generalized Voronoi diagram or follow center behavior to efficiently explore the environment. Use the robot's pushbuttons to start the motion algorithm.
4. You should show the robot's state on the LCD as it creates the world map. In addition, upon completion of the motion algorithm, the robot should display the world map on the LCD. Download the "pixel.c" sample code to help with ideas for displaying the map.
5. Once the robot has created the map, it should be able to use this map to create a path from a start position to a goal location. You will be given a start position and goal location and the robot then plans a path to the goal.
6. You will be graded on the ability to generate the map, the accuracy and detail in the map and how well the robot is able to use your world map to plan a path to a goal location while avoiding obstacles.

Demonstration:

Due to the complexity of this project and the fact that it requires an integration of several concepts, there will be **several** demonstrations required over the next couple of weeks and a graduated grading scale.

PART I – Path Planning

There will be three phases for this part of the demonstration. The student will put down the robot and enter the goal location in topological terms using the pushbuttons, the robot should then move from the start location to the goal position. In phase 2, the student will use the pushbuttons to enter the goal location and move from the start location to the goal position given an occupancy grid. In phase 2, the student will use the pushbuttons to enter the goal location and move from the start location to the goal position given a topological map. The LCD should show robot intent, behaviors, state, map, and planned path.

PART II – Localization and Search

As part of the demonstration, the student will use the pushbutton to enter a goal location. The robot should then use an exploration algorithm to localize itself and provide a visible or audible signal that it has. The robot should then drive to the goal location. In the second phase, the robot should be placed in the world and localize itself. It should provide a visible or audible signal that it has localized. It should then cover the world to find the heat or light source. Once located, it should drive home and display the beacon's location.



PART III - Mapping

The robot should be placed in the world and use a coverage algorithm to create a map of the world. The robot should display the created map of the world on the LCD. The student will then use the world model to plan a path from a start position to a goal location. The student will enter the goal location by using the pushbuttons and then move from the start position to the goal location.

Evaluation:

Table 3 shows the point distribution for the final project.

Table 3: Project Grade Point Distribution

Task	Percent	Due by
Demo 1 –Path Planning (Metric and Topological)	20	Thursday of Week 7
Demo 2 – Localization and Search	20	Thursday of Week 8
Demo 3 –Map Making	20	Thursday of Week 9
Code	20	Sunday of Finals Week
Report	20	Sunday of Finals Week
Competition – 3 rd place	5 pt bonus	
Competition – 2 nd place	7 pt bonus	
Competition – 1 st place	10 pt bonus	

Submission Requirements:

Your team will use the localization, search, and path planning algorithm during the final competition on **Thursday of Week 10**. Details of the final competition will be provided later.

You must submit your properly commented code and final project report by midnight on **Sunday of Finals week**. Recall that properly commented code has a header with the solution name, team members' names, description of the functionality and key functions, revision dates. In addition, all of the key variables and functions in the code are commented. Please use the following checklist to insure that your final project report meets the minimum guidelines.

Project Report Guidelines

1. The document should have default Word settings with respect to font and margins
2. All pages should be numbered
3. All headings must be numbered, left-justified, bolded, and capitalized at the beginning of the section.
4. All figures must have a number and caption underneath (i.e. GUI screenshots)
5. All tables must have a number and title above it (i.e. results error analysis)
6. The cover page should have title, partner names, course number and title, date
7. The report should order should be:
 - Cover page
 - Abstract
 - Table of Contents



- I. Objective
 - II. Theory
 - III. Methods
 - IV. Results
 - V. Conclusions and Recommendations
- Appendix/Supplementary Materials
References (if any)

8. The **abstract** should be a brief statement of the experiment purpose, verification and relevant results
9. The **objective** should state the purpose of the project and associated tasks in your own words
10. The **theory** should state relevant theory that will be used to achieve the purpose
11. The **methods** section should summarize the algorithms implemented to achieve the purpose and the procedures used to test the robot algorithms
12. The **results** section should summarize the results of the tests and verify that the robot was able to achieve the purpose and meet the project objectives
13. The **conclusions and recommendations** should address whether the purpose was achieved, possible sources of error, recommendations to improve the robot algorithm and answer any relevant questions related to the project.
14. Remember this is only a guide for the minimum requirements of your report. You are required to answer any questions or provide any details that you feel aid the reader in understanding the objective, theory, procedure, implementation and results of your project.