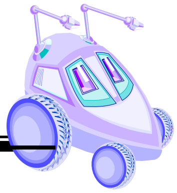


Final Project Navigation

(Demonstration 1 due in class on **Tuesday, 5/14/09**)

(Competition in Kahn room on **Thursday, 5/21/09**)

(Project Report and Code due in Angel drop box by midnight on **Friday, 5/22/09**)

Reference: http://roboticsprimer.sourceforge.net/workbook/Navigation

Purpose: The purpose of the final project is to demonstrate the integration of some of the concepts learned this quarter by implementing several navigation tasks on the Traxster II. You will use metric and topological path planning to move the robot to a goal location given an a priori map. You will then use the robot to explore an unknown environment and create a map. The robot will use this map to plan a path to a goal location in the environment. These navigation tasks will be tested and demonstrated in an artificial and natural environment.

Equipment: Base Robot
Various sensors and peripherals

Software: Microsoft Visual Studio.NET 2008 with C#
Serializer.NET library and firmware
Bluetooth transmitter

I. Part I – Metric Path Planning and Execution

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 test arena will be 6 ft x 6ft with 1 ft x 1 ft obstacles (see Figure 1). The configuration space will be an occupancy grid divided into 6" x 6" squares, where free space is represented by 0's and occupied space by 1's. You should devise a scheme to represent the robot's start position and goal position and your code should be flexible such that these values can be specified at run time (see Figure 2).

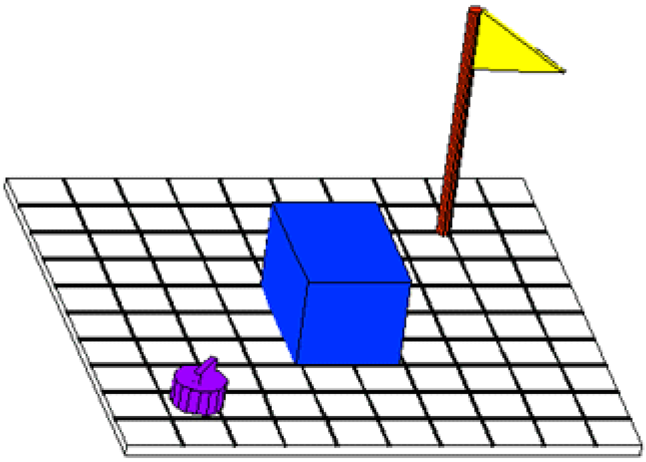


Figure 1: Test Arena

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	**2**	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	**S**	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 2: Test Arena Configuration Space

The wavefront is created by starting at the destination and creating eight connected neighbors back to the start point. The robot then follows the numbers in the reverse order to arrive at the goal point (see Figure 3).

11	10	9	8	7	6	5	4	3	3	3	4
11	10	9	8	7	6	5	4	3	**2**	3	4
11	10	9	8	7	6	5	4	3	3	3	4
11	10	9	**1**	**1**	**1**	**1**	4	4	4	4	4
11	10	10	**1**	**1**	**1**	**1**	5	5	5	5	5
11	11	11	**1**	**1**	**1**	**1**	6	6	6	6	6
12	12	12	**1**	**1**	**1**	**1**	7	7	7	7	7
13	13	13	**1**	**1**	**1**	**1**	8	8	8	8	8
14	14	14	**1**	**1**	**1**	**1**	9	9	9	9	9
15	15	14	13	12	11	10	10	10	10	10	10
16	**15**	14	13	12	11	11	11	11	11	11	11
16	15	14	13	12	12	12	12	12	12	12	12

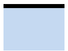
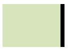
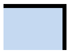
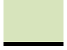
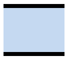
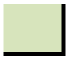
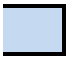
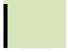
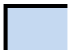
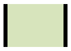
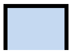
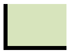
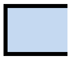
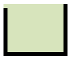
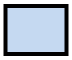
Figure 3: Test Arena Wavefront

During the demonstration, you will be given the map as a 12 x 12 array of 0's and 1's in .txt file that represents obstacles and a goal point. You will be given the robot's start position at the beginning of the demonstration. Your program should open the .txt file, read the world map as an array, run the wavefront algorithm and plan the path to move the robot from the start position to the goal. You should then place your robot at the start position and press start and it should move to the goal point. 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.

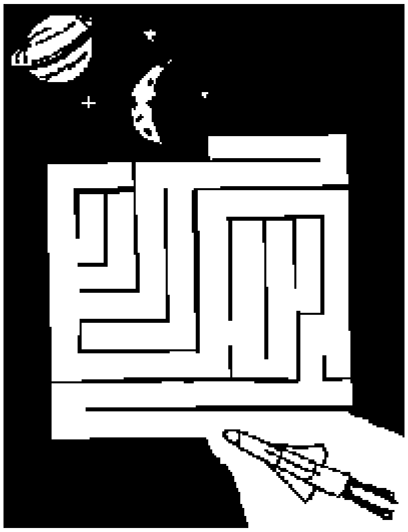
II. Part II – Topological Path Planning and Execution

In this exercise, you will use an a priori topological map to plan a path from a robot start location to a goal position. Instead of representing the world map as an occupancy grid, it will 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	Direction	Wall Location
0	0000		
1	0001	North	
2	0010	East	
3	0011		
4	0100	South	
5	0101		
6	0110		
7	0111		
8	1000	West	
9	1001		
10	1010		
11	1011		
12	1100		
13	1101		
14	1110		
15	1111		

Using the coding in Table 1, the maze shown in Figure 4 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 4: 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 12 x 12 array of integers in a .txt file that represents the world's salient features. You will be given the robot's start position at the beginning of the demonstration. Your program should open the .txt file, read the world map as an array, and use an algorithm to plan the path to move the robot from the start position to the goal. You should then place your robot at the start position and press start and it should move to the goal point. 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.

III. Part III – Map building, path planning and execution (artificial environment)

In this exercise, you will build a map of the 6' x 6' test arena. You have the option of using a metric or topological encoding for the map. You will use the map to plan a path from a start to goal location. Try to reuse as much code as possible from Parts I or II once the map has been created to simplify the path planning and execution. If you prefer, you may also implement any of the other mapping or path planning algorithms presented in class. 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 dimension of the environment and the robot start position for the mapping try to create the best map possible. Use a wander, cover, wall following or other motion algorithm to move the robot through the environment and code the space where the IR or sonar hits objects with a value.

During the demonstration, you will start the robot and let it explore the environment and create the map. Your program should output the matrix array as a text file which represents the world map. You will be given a start position and goal location for your robot in the newly created map. You will then use an algorithm to plan the robot's path and start the robot to execute it. You will be graded on how well the robot is able to use your world map to plan a path to a goal location while avoiding obstacles.

IV. Part IV – Map building, path planning and execution (natural environment)

Select a location to map on the second floor of Moench Hall and repeat Part III.

<u>Evaluation</u>

Because this project requires an integration of several concepts and is more difficult than prior labs, there will be two demonstrations required and a graduated grading scale. The following tables show the grade level and point distribution for the final project.

Table 2: Project Grade Level

Grade Level	Task
D	*Metric path planning and execution*
C	*Topological path planning and execution*
B	*Map building, path planning and execution (artificial environment)*

Table 3: Project Grade Point Distribution

Task	Percent
Demo 1 – Path Planning (metric, topological)	30
Demo 2 – Map Building, Path Planning (artificial, natural)	30
Code	20
Report	20
Competition – 3rd place	5 pt bonus
Competition – 2nd place	7 pt bonus
Competition – 1st place	10 pt bonus

Submission Requirements

You must demonstrate your localization and maze navigation routine by **Thursday, 5/14/09**.

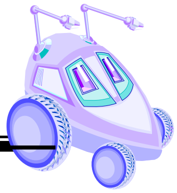
You must demonstrate your map making and path planning routine by **Monday, 5/18/09**.

Your robot will use its navigation routines in a competition on **Tuesday, 5/19/09**. You submit your properly commented code and final project report by midnight on **Friday, 5/22/09**. Please use the following checklist to insure that your memo meets the basic 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 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 implement the robot tasks analyze the experimental results
11. The *methods* section should summarize the procedure used to test the robot algorithms and confirm that they meet the project objectives
12. The *results* section should summarize the results of the tests and provide error analysis if necessary
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. Also provide any other information that you feel is significant and aids the reader in understanding the purpose, procedure, and results of your experiment/ project.