CS 3050: Group Project

CS 3050

November 10, 2015

DUE: 10/12/2014

Topic: Motion planning of robots

Description

This is the final project of the course. The project can be done alone or in a group of at most people. The project contributes to 10% of your total grade. You should send an email to Eric and myself by November 17 with the subject "CS3050 project" indicating whether you are choosing to do the project alone or in a group. If you choose to do the project in a group, please include the name of your teammates in the project (one email per team is fine).

The purpose of this project is to utilize and build upon your knowledge in graph searching algorithms. In robotics, it is common to traverse through rooms or buildings and gather data. To do this the robot traverses through, using graph searching, until it finds the way out. In this project you will be implementing a motion planning algorithm which coordinates the motion of two robots in a room. The first robot starts at starting point S and has to travel to exit location E. The second robot starts at starting location F and has to travel to exit location L. In the room, a robot can only move one spot per movement, which could be up, down, left and right. This means that a robot can't teleport to other parts of the room. Furthermore, for coordination purposes:

- Only one robot moves in one time instant. The two robots may alternate their movements, but they never move simultaneously.
- In order to maintain their coordination, they are *not allowed* to come closer than an *interference parameter* r. In other words, the two robots should always maintain a horizontal and vertical distance $> r$.

You cannot assume dimensions of the room as each line of the room may be sized differently than other lines. An example input that you should expect will be similar to the sample input file is included below and in Blackboard.

Constraints

The most natural way to implement the project is to view the room as a graph with each square representing a vertex (what are the edges?) If you choose to use this representation, you will develop a graph traversal algorithm which finds a path for the first robot from (S) to (E) and a path for the second robot from (F) to (E) (use backtracking when going to previous spaces) such that the two robots are never closer than r.
The constraints of the program include:

1. The program can be written in C or C++. If you want to use a different language, please check with us.
2. Either use a netbeans project or use a Makefile to compile and run the program. Please do not use Visual Studio.
3. Include a README file to tell the TA's how to run your program and how to interpret your output.
4. You should include a project report in which you detail your efforts, the algorithms used (along with their complexity) and the contribution of your team members. We shall use this report primarily to give you partial credit if your program does not work.
5. You may not use any graph theory libraries to solve this program.

6. All graphing algorithms should be implemented.
7. A moderate amount of error checking and resource management is required. Your application should check that each line from the input file is properly formatted, that the file is successfully opened, the file is successfully closed upon reading of the file and that all allocated space is de-allocated at the exit. If the input is not properly formatted, you should report the error on stdout and exit the program.
8. You may use standard input/output libraries, arrays, vectors, stacks, heaps, lists and queues. If using other libraries, please please check with us.
9. A moderate amount of formatting and documentation is required. Comments should be descriptive and used to illustrate the purpose and inner workings of an algorithm or function; they should not be used to annotate each line or self-evident logic.

Input

Your program should take at least two arguments on the command line. One of these arguments is the input file to the program. The second argument is the interference parameter r. input file itself is formatted as follows. The input file is the floor plan of the room. In the floor plan, each character of the floorplan means:

1. #: Wall or obstacle
2. End of line character and start of line also indicates that you have reached a wall.
3. Space: Empty spot that a robot can move to.
4. S: Starting position of the first robot in the room.
5. E: Exit position of the first robot in the room.
6. F: Starting position of the second robot in the room.
7. L: Exit position of the second robot in the room.

Sample Input File

```
           ###################
           #       #        F###################################
           #   #  E#                                           #
           #   #####         ################################## #
           #                 #                                # #
############################  ################                 # #
#                                         #                 # #
#       ##################                #                 # #
#                      #####              #                 # #
#      ###############       #            #                 # #
#      #                     #            #                 # #
###################  ##########################              # #
               #L                      #                     # #
               #             #     #   ######################## #
               #             #     #                          #
               ############  #     #   ########################
               #             #     #   #
               #             #######   #
               #             #     #   #
               #             #     #   #
               #             #     #  S#
               ########################
```

Due Dates

- November 17 is the date that you must inform us about the composition of your team.
- During the week of November 30-December 4, please try to meet one of the TAs or myself during office hours and show the progress of the project. If the times do not work, please send us an email and we will try to find additional time to check the progress.
- The final submission is due December 10 at 11:59:59 pm.

Submission

You should place all your submission material into a folder named as follows: *pawprint1_pawprint2_pawprint3_project*. Where the *pawprint1_pawprint2_pawprint3* is the list of you and your group member pawprints.
You will submit one file, a compressed directory containing all the source code and appropriate Makefile(s). The naming convention should be: *pawprint1_pawprint2_pawprint3_project.tgz* or *pawprint1_pawprint2_pawprint3_project.zip*.

Grading

There are 100 points possible for this assignment. The grade breakdown is as follows:

- 15 points for README, error checking and resource management.
- 15 points for general programming style and adherence to the constraints.
- 20 points for the project report.
- 15 points for correctly inputting the floorplan.
- 35 points for finding and displaying the paths taken by the robots.