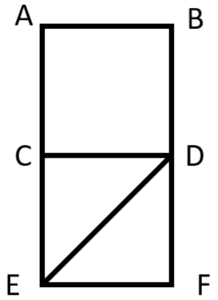
LAB 2: SEARCH

Due: Tuesday, February 23rd
3pm class demo, 5pm code/video submission



PatrolCorp has hired your team to write a new surveillance path planning algorithm for their robots. Specifically, PatrolCorp requires that every hallway is patrolled once, and only once, per patrol cycle. In the building diagram on the right, one possible trajectory that will satisfy PatrolCorp's requirements is CDBACEDFE. Your algorithm may choose any intersection within the building as a starting point.

PatrolCorp will not give you access to their top secret facilities for testing of your algorithm. Instead, they have provided several example building plans for you to test on, as well as one additional mystery building plan which will be revealed at the end of the project. You will have to demonstrate your system by using the Scribbler's drawing capabilities.

1. Implement a method to read in the problem specification files provided with this assignment.

   Each building plan is stored in a separate CSV (Comma Separated Values) file. The file contains two distinct components (1) the building junction locations (vertices) and (2) the building hallways (edges). In (1), each row of the file specifies one vertice where each row contains three values: vertice name, x position, and y position. In (2), each row of the file specifies one edge where each row contains two values: the first vertice name, and the second vertice name. The two sections are separate by a single empty row. An example of a two vertice map with one edge between is seen below:

   ```
   A,0,0
   B,1,1

   A,B
   ```

2. Write an algorithm that selects a starting node and prints out a path that the robot should follow (e.g., CDBACEDFE).

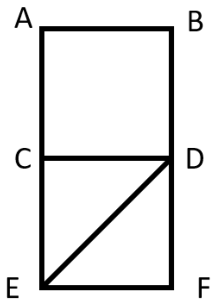
3. Using the coordinates provided in the problem definition file, calculate the motion commands your robot requires in order to complete the path above.

4. Demonstrate that the robot correctly follows the intended path. Place the robot on paper (you

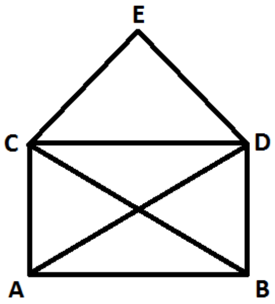
may need to use more than one sheet), and put a pen or marker into the hole at the center of the robot. Have the robot complete the full planned trajectory and verify that the resulting diagram correctly reproduces the intended diagram, vising each edge once. Note that while there are several API functions to control the robot, we recommend using TurnBy (e.g., *TurnBy(90, "deg")*) in this assignment. If you get an error when calling this method, see notes at the end of this document.

The two example buildings plans that PatrolCorp have provided are seen below:

Map #1



Map #2



The final map (Map #3) will be revealed during class on Feb 23rd, where your scribbler will be expected to draw the map during the class period.

Evaluation:

Part 1 (85% of grade): Demonstrate that your algorithm works using the provided problem definition files. Submit the following files on Canvas by 5pm on Feb 23rd.

a. A zip file containing your code. (*Lastname1Firstname1_Lastname2Firstname2_code.zip*)
b. A video of your robot executing Map #1 and Map #2
c. Each partner should complete the peer evaluation form (please complete this even if you a working without a partner).

Part 2 (15% of grade): Bring your laptop and robot to class on Feb 23rd and demonstrate the functionality of your algorithm on a brand new domain. Paper and markers will be provided.

Grading Rubric

Print out valid path for Map #1	20 points
Print out valid path for Map #2	20 points
Robot correctly executes Map #1	20 points
Robot correctly executes Map #2	20 points
Peer evaluation form	5 points
Robot prints out valid path for Map #3	7.5 points
Robot correctly executes Map #3	7.5 points

Notes:

- Using commands such as *TurnBy(90, “deg”)*, your robot should be able to reproduce the indicated patterns with reasonable precision. It will not be perfect, but the shape should be recognizable.
- If you get an error that *TurnBy* is not defined, check your version of Myro. When you run *init()*, the listed version should be 1.1.6. If not, run *upgrade(“myro”)*
- Python has a CSV library. You are welcome to use it to read in the building map.
- The building maps do not specify a scale. You are welcome to define any scale you want for your implementation. The dimensions of the paper that will be provided in class on Feb 23rd are 24” x 36”.
- Depending on the map, the robot’s end location may not be at the same node as its start location. This is fine.