Project #3

Robot Search and Rescue

Introduction

- NASA hires your robot explorer to travel to a distant 2-dimensional planet
- Unfortunately, an astronaut also crash lands on the planet
- Your job: To design a robot search and rescuer that utilizes its available information to navigate from its current position to the astronaut's crash site.

Your Task

- Design a search algorithm based on the following (input.dat):

MAP width height
- Defines the width and height of the world where (0,0) represents the top, left corner of the map

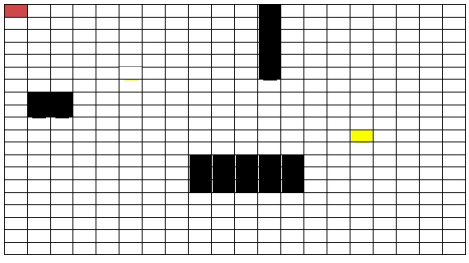
ORIGIN x_position y_position theta
- Defines the starting (x,y, θ) position of the robot where orientation is N, S, E, or W

GOAL x_position y_position
- Defines the (x,y) location of the goal position of the astronaut crash site

OBSTACLE x_position y_position width height
- Defines the width and height of an obstacle, starting at (x,y). These are grid locations that the robot cannot navigate through.

Example input.dat

MAP 20 20
ORIGIN 0 0 E
GOAL 15 10
OBSTACLE 8 12 5 3
OBSTACLE 1 7 2 2
OBSTACLE 11 0 1 6



Output.dat

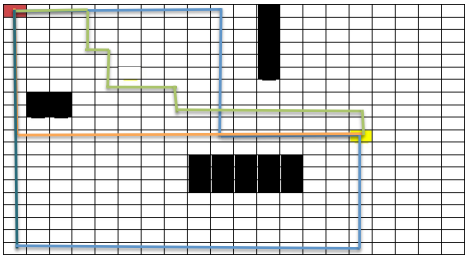
- Given the map environment, your program must output to "output.dat"
  - sequence of *x, y* grid points from the start to the goal position
  - total number of turns (i.e. direction changes) from start to goal
  - total number of grid cells traversed from start to goal
- Note
  - The robot can only navigate in the North, South, West, and East direction
  - The robot cannot navigate beyond the boundaries of the world

Grading Criteria

- Optimal credit (20 points for each case): Program outputs an optimal solution (i.e. minimum cells, then minimum turns) within 1.5 minutes
- Suboptimal credit (17 points for each case): Program outputs a sub-optimal solution within 1.5 minutes
- Partial credit (14 points for each case): The robot outputs a sequence of sequential grid locations that may not provide a solution and/or does not terminate within 1.5 minutes
- Deductions
  - Program crosses through an obstacle or a wall (3 pts for each occurrence)
  - Program will not compile (60 pts)
  - Program crashes or does not terminate (20 pts)
  - Program does not contain at least one class file (20 pts)
  - Program does not comply with good design constraints or requirements (e.g. non-working makefile, no zip file, etc.) (2 - 10 pts)

Example Scenario

MAP 20 20
ORIGIN 0 0 E
GOAL 15 10
OBSTACLE 8 12 5 3
OBSTACLE 1 7 2 2
OBSTACLE 11 0 1 6



25 cells, 2 turns

25 cells, 2 turns

25 cells, 7 turns

44 cells, 3 turns

Submission

- A zip file containing your code is to be submitted via T-Square (ECE3090-Assignments-Project3) by the DUE date of October 19th
- We will test your code on the Jinx cluster so make sure your program correctly compiles and runs on that system (development can be done on your machine - e.g by using GNU ARM). Information on the cluster is located at: http://support.cc.gatech.edu/facilities/instructional-labs/jinx-cluster
- Do your own work and abide by Georgia Tech's Honor Code!!