

### CSCI 4360/5360 Homework 8 (Due midnight, Thursday, April 13<sup>th</sup>)

In projects, we will program a mobile robot to do motion planning with the wavefront planning algorithm. The objective is to navigate a known obstacle course (a “world” from a given starting point to a given goal point, while not coming in contact with any obstacles nor running outside the “world” space.

The world consists of a 8'x4' flat space divided into 6"x6" squares, which makes for a 16x8 square grid. The obstacles are 4 squares long and 1 square wide and there are 4 of them. On demo day, your robot will run the following two courses:

#### World 1:

```
0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
```

#### World 2:

```
0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
```

We will give the robot its initial coordinates (x,y starting from the upper-left corner, which is positioned at 0,0), and the goal coordinates. The robot is to plan an efficient path from its starting position to the goal, then execute it, without touching obstacles or going off the world map, and stop in the goal square.

#### For this homework:

Each student independently implements a C++ version of the wavefront algorithm. [This program counts 200 points towards the homework.](#)

- Make sure to test your program with the above two worlds, where the goal position can be any square in the world. You may copy the two data files from the course page.
- For each world configuration, your program should allow the user to enter the x- and y-coordinates of the goal square in the form of:

**x y <press enter>**

Your program outputs the best path in the form of :

**The number of steps from (x\_start, y\_start) to (x\_goal, y\_goal) is k:**

**The path is: (x1, y1) (x2, y2), ... (xk, yk)**

where xi, yi are the coordinates of the intermediate steps in the path derived.

- Turn in your program electronically in the D2L Dropbox marked “wavefront”

**Only turn in the source file: wavefront.cpp**