

CISC 3415 Fall 2011, Project 6

1 Description

This project builds directly on the last one. Now that we can get the robot to follow a plan, we turn to building a plan.

2 Start up

1. Prof. Parsons will give you a folder called `project6` which contains a lot of familiar-looking files.
2. The control program is in the file `make-plan.cc`, so first build that:

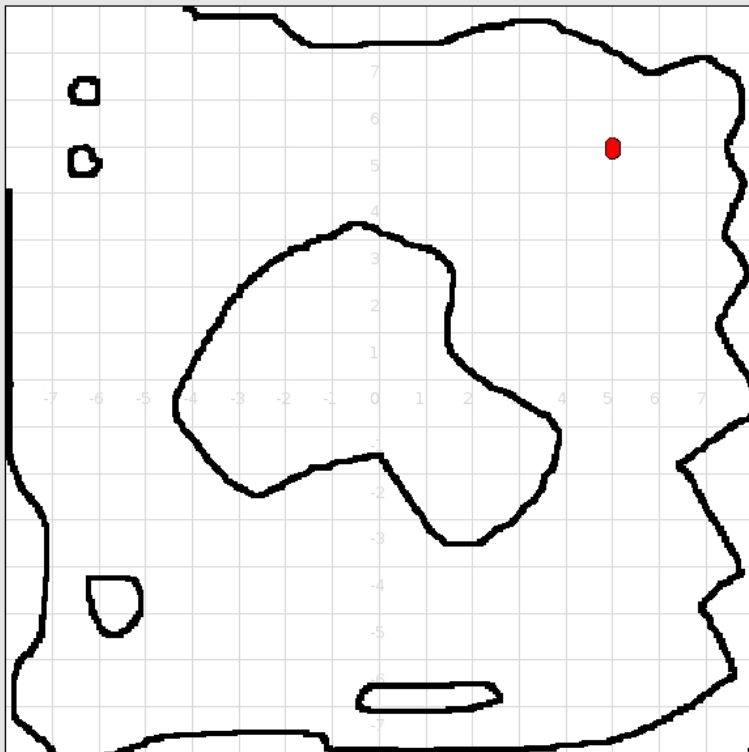
```
./build make-plan
```

then run `player` on the config file `world6.cfg`:

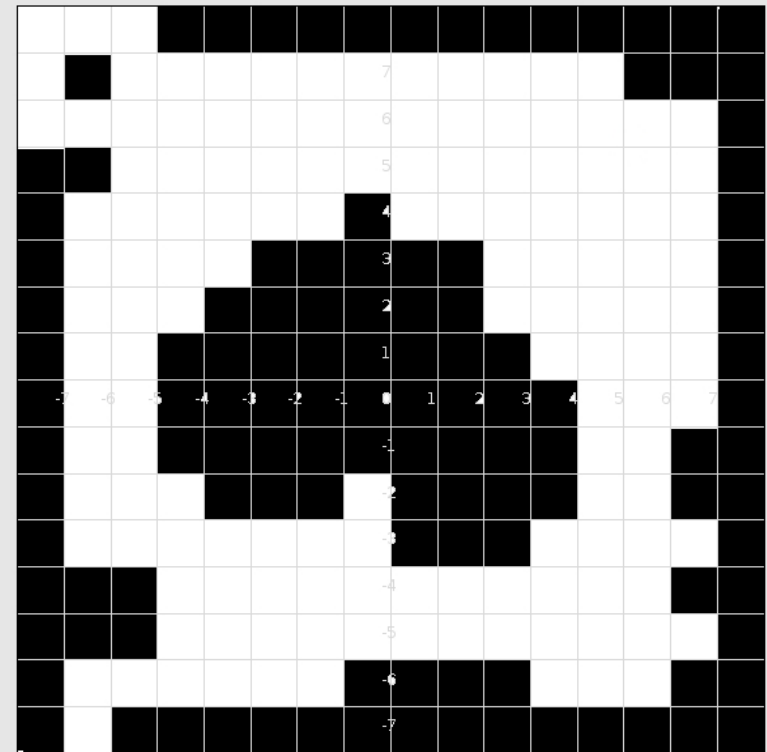
```
player world6.cfg
```
3. The robot, as it has at the start of previous projects, will trundle off to the south-east.
4. However, before it starts to move, it will display something new
5. This is an occupancy grid with a resolution of 0.5 meters (for the format of the grid, see below). For the purposes of the project, this is a *map*.
6. The program will also display a plan, just like last time.

3 Map format

1. In terms of this project, an occupancy grid map is an array of integers.
2. A 1 indicates that a square is occupied, a 0 indicates that it is not occupied.
3. The code in `make-plan.cc` assumes that the map is square, so there is only one parameter that controls the size of the map.
4. In the example in the folder `project6`, each grid square is half a meter on each side.
5. This example map is for a map that is 16 meters on each side.
6. An example map is given in Figure 1(b), while Figure 1(a) shows the corresponding simulated world from Stage.
7. Note that this is NOT the world you'll be using in the project (though it is probably familiar from previous projects).
8. In addition, the occupancy grid in Figure 1(b) is lower resolution than the map in `make-plan` — each square is one meter on each side.



(a) An example world in which the robot operates.



(b) An occupancy grid map.

Figure 1: A simulated world and the corresponding occupancy grid map

4 Use the map to plan

1. The big task for this week is to write a robot controller that can use an occupancy grid to plan a route through the world represented by the occupancy grid.
2. The task assumes that you start with an occupancy grid, a set of coordinates that the robot starts at, and a set of coordinates that the robot has to end up at.
3. The task is then to generate a plan that is a sequence of sets of coordinates. The first of these sets of coordinates is the initial location of the robot, the last set of coordinates is the final location of the robot, and each intermediate point is a point that the robot should pass through on its way from the initial location to its final location.
4. You need to choose the intermediate locations carefully so that the robot can move in a straight line from one to the next without encountering an obstacle.
5. A suitable way to create the plan is to use the wavefront planner we discussed in class.
6. You should use the map that came in `make-plan.cc` since that corresponds (well enough) to the map that `world6.cfg` will load.
7. Your final program should be able to create a plan from the map and then follow the plan.
The plan following can just be the code you wrote for the last project.
8. The start point for the robot is where it starts in the simulation, $(-6, -6)$. The end point is $(6, 6)$.
9. Once your controller can do this, save your working program as **<your-names>-proj6.cc** and make sure you put your name in the comments.
10. You'll need to submit this to Prof. Parsons after you are done with the project,

5 Working on a real robot

1. By the time you get to the lab on Tuesday, Prof Parsons should have finished setting up the config files and instructions for a version of this exercise that will run on a real robot.
2. This will be an extra credit exercise with separate lab sheet.