CS255B Project 3

For this project, assume that the robot starts out well-localized. To do this, you must be able to locate points within the map, place your robot there (internally), and center a particle cloud on it. Or, reliably globally-localize the robot.

1. Gradient planner
   Write a gradient planner, constructing a navigation function for an arbitrary 2D configuration space in a grid. Your solution should have the following properties.
   - interpolation to give Euclidean distance, rather than Manhattan or chessboard distance
   - fast updating using Dijkstra’s algorithm or similar method
   - ability to have multiple goals
   - safety margins around obstacles
   - path extraction from an arbitrary start position

2. Local controller
   Write a controller that will take the robot along the path produced by the gradient planner. The idea behind the local controller is to control the robot based on current information from the sensor, no matter what the state of localization. The controller should be able to move the robot around un-mapped obstacles.

   You have your choice of local controllers:
   - The Project 1 controller, modified to take into account the global path
   - A local gradient planner that takes as input the sensor readings, the map, and the global plan
   - Any other controller that you think would work.

3. Map update
   This is a chance to do some mapping! Well, at least to modify the map. Propose and implement a solution to map updating in the presence of new objects in the environment. Your solution should
   - add new, stable objects to the map
   - get rid of objects that are no longer present
   - decide when to replan

TESTING

4. Showing the gradient
   You should be able to show a graphical plot of the gradient field for a world of your choosing. Put a goal somewhere in the world, and show the gradient at each point by using a small vector (a line segment is fine). You must also draw the path from a clicked point to the goal, to make sure that gradient interpolation is working.

5. Moving to goals
   You will be given a start position and a set of three goals to go to. You will input this information, and the robot must achieve the three goals efficiently.

6. Avoiding traps
   You will be given a start and goal position. The environment will be such that a new obstacle will lead the robot down a garden path, from which it must recover.