CSCE452-Project-3

Report

Our project started with three main goals:

1. Be able to click on a part of the window to create an obstacle of a constant size

2. Be able to specify a start and end position for the paint bot and have the robot find a path between these two positions

3. Implement cell detection and obstacle avoidance

We completed the above by doing the following:

1. Be able to click on a part of the window to create an obstacle of a constant size

We completed this by taking the point which is clicked and drawing a box first of 200 x 200, then 150 x 150, and finally 100 x 100. The widths are simply added onto whatever point is clicked by the mouse and the obstacles are created. They are represented in code by two "point" objects which have a reference height and width that can be used for calculating the total area which they cover.

2. Be able to specify a start and end position for the paint bot and have the robot find a path between these two positions

This was fairly simple. Once all three obstacles have been placed and the user has chosen a start and endpoint for the robot it will perform a breadth first search checking for collisions with obstacles until it finds its way to the end position. You can see the implementation of breadth first search in graph.py. This was much easier than we thought it would be

3. Implement cell detection and obstacle avoidance

We completed cell detection by having the planner check for bisecting, intersecting, aligning, and edge sharing between each of the three obstacles. During our breadth first search for the destination the algorithm will perform a check on the window pane to see if the cell it wants to move to is coloured red or white. If the colour is white we know that there is no object there, if it is red we will be within the obstacle and therefore would have a collision.