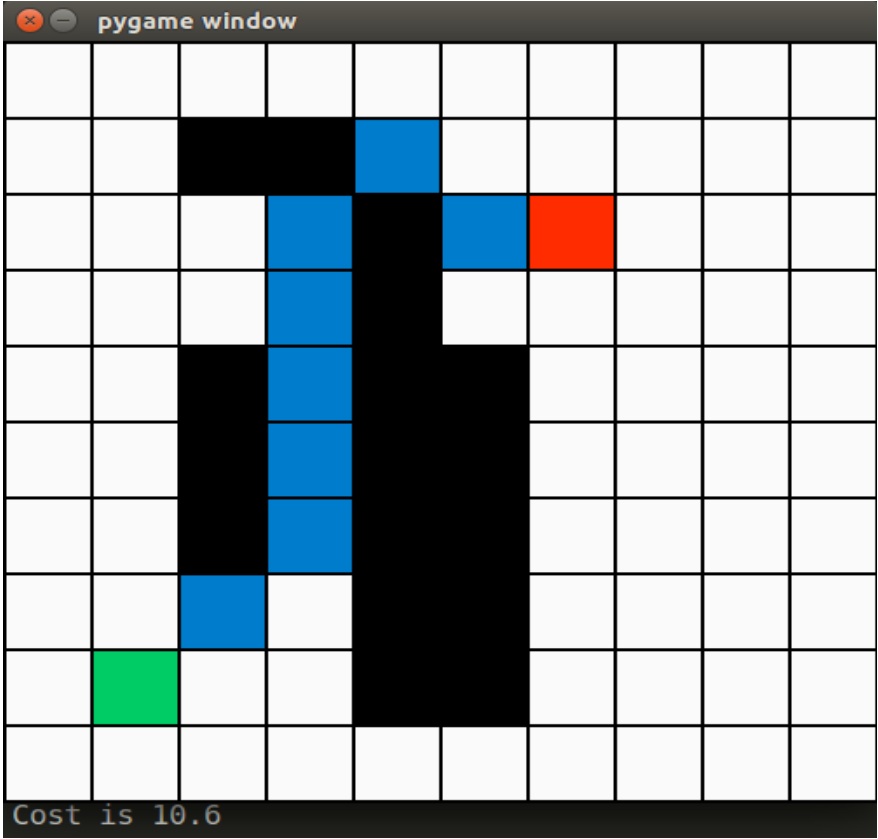
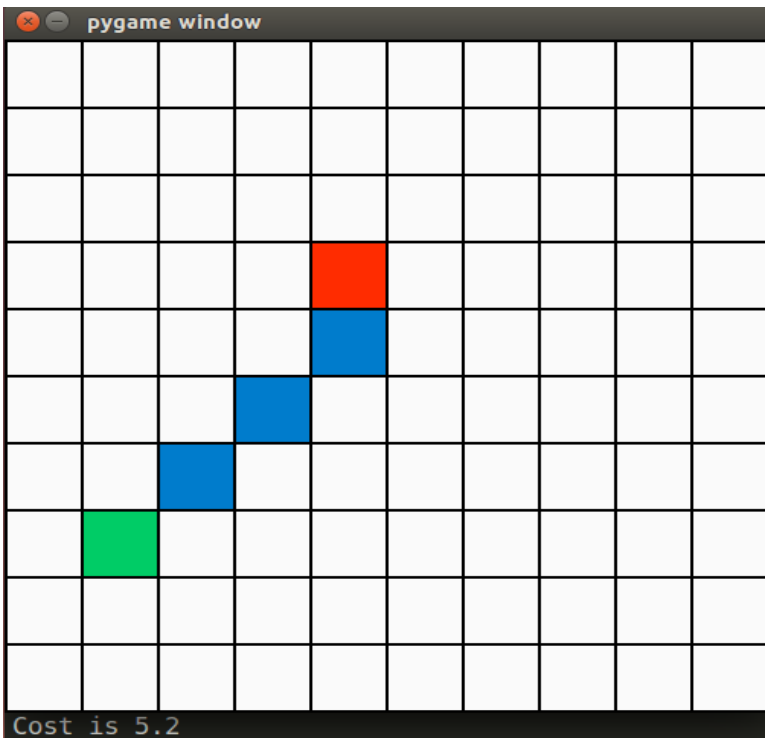


Assignment 2 A* D*

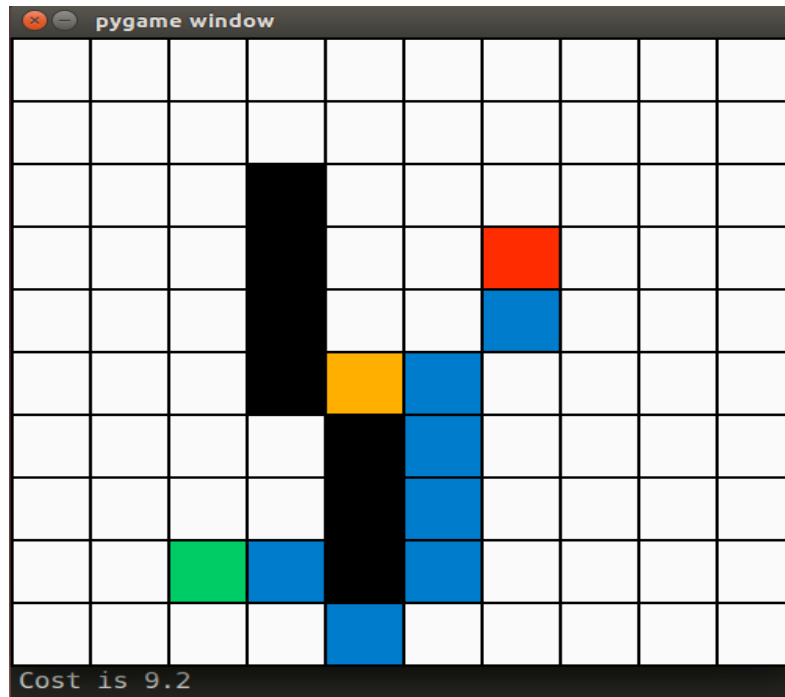
Black cell = obstacle
Green cell = start point
Red cell = End point
For A* (With obstacles)



Without Obstacles



For D* Orange cell = new dynamic obstacle



Explanation:

A*

A* uses the distance between the current location and the target and moves to the square that has the smallest distance. It evaluates squares (henceforth called a “node”) by combining $h(n)$, the distance(cost) to that node and $g(n)$, the distance(cost) to get from that node to the goal node. The total cost $f(n) = g(n) + h(n)$ is calculated for each successor node and the node with the smallest cost $f(n)$ is selected as a successor.

The agent can move in 8 possible directions, N,S,E,W,NE,NW,SE,SW

Cost for N,S,E,W = 1

Cost for NE,NW,SE,SW = $\sqrt{2}$, assumed to be 1.4

The distance between two nodes is simply determined by calculating the straight line of sight distance between the two nodes. Though this might not be the true distance, it never overestimates the actual distance.

In my code I have maintained two lists, an OPEN list and CLOSED list

The list OPEN stores all successive paths that are yet to be explored while list CLOSED stores all paths that have been explored.

I have maintained a dictionary for each cell that covers its state(telling if it is start,goal or wall), its f , g and h costs and also its parent node. This is used at the end to trace the path from the Goal to the Start position, thus generating the optimal route.

My path tracing is done when I reach the goal, I start using the parent key in my dictionary to trace back the path to the goal.

For A* a minheap of the f -scores is also maintained, depending on which we decide which path to follow.

D*[Code not working in every case (errors exist)]

In My D* code, I have run the A* algorithm ones, thus getting the OPEN and CLOSED list, my minheap and also my dictionaries already available. Ones doing this I create a new obstacle in my optimal path, and then make the h cost of all the nodes whos parent was the dynamic obstacle node to a very high value. And from there I remove that particular node from my minheap and my f-score dictionary. And apply A* again, so this time when it reaches any of the nodes with very high h-value it avoids it and takes a new path, which makes it reach goal point.