Project Title: NAVIGATE THE MARS ROVER
Problem statement: To calculate the shortest path between the source and the destination through a stopping point. Consider a source node ‘S’, a stopping point ‘SP’ and a final destination node ‘D’. The sum of the length of the shortest path from ‘S’ to ‘SP’ and from ‘SP’ to’ D’ would be returned as shown in the following block diagram.

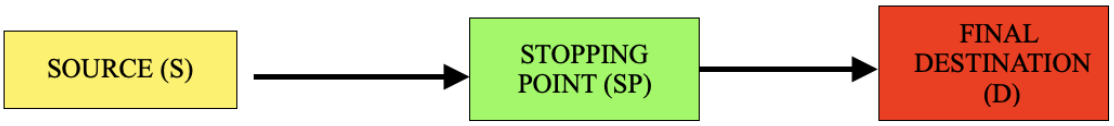

FIG: SHORTEST PATH FROM SOURCE (S) TO FINAL DESTINATION (D) THROUGH THE STOPPING POINT (SP)

Grid Made:
We have made a grid in which we'll be having three points in the starting first as the starting point, second as the stopping point and the third as the destination.
By using a mouse command, the obstacles will be built on the grid. These obstacles are of grey colour.
Once the search starts it recognises the grey boxes in the grid as not walkable and creates a shortest path considering the obstacles.

Algorithm to be used: A* Search Algorithm

A*Search algorithm, unlike other algorithms, is a smart algorithm which is used by many games and web-based graphs to find the shortest path. A*Search algorithm is a graph traversal and a path search algorithm.

Working of the Algorithm:

- Initially the coordinates of source node ‘S’ will be passed as an argument to the A* algorithm along with the stopping point ‘SP’.
- The heuristic that is the estimated cost movement between the source node and the destination node would be calculated and hence the length of the shortest path would be returned and stored in variable say ‘x’.
- After this, the coordinates of stopping point ‘SP’ along with the coordinates of final destination would be passed as arguments to the A* algorithm function.
- The heuristic would be calculated which would help to calculate the length of the shortest path from stopping point ‘SP’ to the final destination that would be returned and stored in the variable say ‘y’.
- The sum of lengths ‘x’ and ‘y’ would be returned and displayed on the website screen as the length of the shortest path from source to destination through the stopping point ‘SP’.
- Heuristic would be calculated using the following three ways:
    - ➢ Manhattan: It is the sum of absolute values of differences in the goal’s x and y coordinates and the current cell’s x and y coordinates respectively.
    - ➢ Euclidean: It is nothing but the distance between the current cell and the goal cell using the distance formula, **h** = sqrt ((current_cell.x – goal.x)2 + (current_cell.y – goal.y)2)
    - ➢ Chebyshev: It is the maximum of the absolute value of the difference between both the coordinates.
- Further the A* Algorithm also provides some option on the basis of which it calculates the shortest path that are as follows:
    - ➢ Allow Diagonally: In which the shortest distance includes the diagonal movement.

- ➢ Bi-Directionally: In which the algorithm searches for the shortest path from both the sides that is from source to destination and from destination to source as well.
- ➢ Don’t cross corners: In which the algorithm doesn’t cross the corners of the walls constructed by the user to find the shortest path.