A\* algorithm has 3 parameters:

* **g :** the cost of moving from the initial cell to the current cell. Basically, it is the sum of all the cells that have been visited since leaving the first cell.
* **h :** also known as the heuristic value, it is the **estimated** cost of moving from the current cell to the final cell. The actual cost cannot be calculated until the final cell is reached. Hence, h is the estimated cost. We **must** make sure that there is **never** an over estimation of the cost.
* **f :** it is the sum of g and h. So, **f = g + h**

At every step, the f-value is being re-calculated by adding together the g and h values. The minimum f-value node is selected to reach the goal state. This process continues until the algorithm reaches its goal cell.

Steps involved in A\* algorithm

1. Initialize the open list

2. Initialize the closed list put the starting node on the open list (you can leave its **f** at zero)

3. while the open list is not empty

a) find the node with the least **f** on the open list, call it "q"

b) pop q off the open list

c) generate q's 8 successors and set their parents to q

d) for each successor

i) if successor is the goal, stop search

ii) else, compute both **g** and **h** for successor successor.**g** = q.**g** + distance between successor and q

successor.**h** = distance from goal to successor (This can be done using many ways, we will discuss three heuristics- Manhattan, Diagonal and Euclidean Heuristics)

successor.**f** = successor.**g** + successor.**h**

iii) if a node with the same position as successor is in the OPEN list which has a lower **f** than successor, skip this successor

iV) if a node with the same position as successor is in the CLOSED list which hasa lower **f** than successor, skip this successor otherwise, add the node to the open list end (for loop)

e) push q on the closed list end (while loop)

Future Improvement:

1)It can only take square or rectangle as obstacle , not made for any type of polygon .So future aim is to make it to except any kind of polygon obstacle .

Some Thoughts to implement future improvement :

1)we can draw a line between 2 points and with the help of location of point with respect to line we can see which point is inside the polygon and which is outside .