A* Algorithm pseudocode

```
1       Create a node containing the goal state node_goal
2       Create a node containing the start state node_start
3       Put node_start on the open list
4       while the OPEN list is not empty
5       {
6       Get the node off the open list with the lowest f and call it node_current
7       if node_current is the same state as node_goal we have found the solution; break from the while loop
8           Generate each state node_successor that can come after node_current
9           for each node_successor of node_current
10          {
11              Set the cost of node_successor to be the cost of node_current plus
                    the cost to get to node_successor from node_current
12              find node_successor on the OPEN list
13              if node_successor is on the OPEN list but the existing one is as good or better then
                   discard this successor and continue
14              if node_successor is on the CLOSED list but the existing one is as good or better then
                   discard this successor and continue
15              Remove occurences of node_successor from OPEN and CLOSED
16              Set the parent of node_successor to node_current
17              Set h to be the estimated distance to node_goal (Using the heuristic function)
18               Add node_successor to the OPEN list
19          }
20          Add node_current to the CLOSED list
21      }
```