Artificial Intelligence

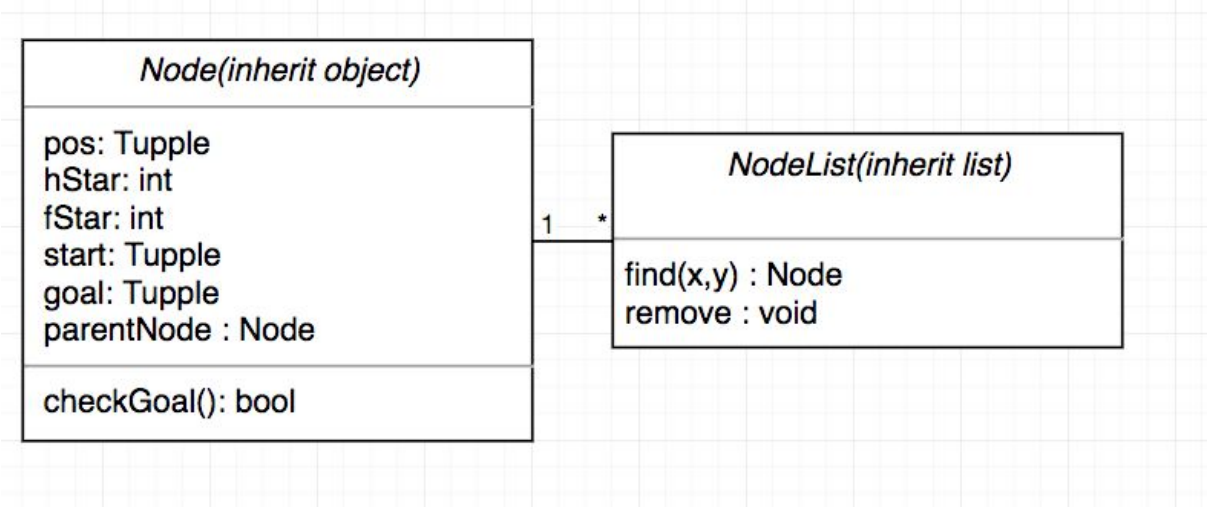
AI Assignment1

a1658945
HIDEKI YOSHINAGA

The program robotplanner is built for searching the best path between 2 coordinates in a given map. The algorithm for the search is using A* and its written in python.

there are two classes, one is “Node” and the other is “NodeList”

UML



- **Node**
  - pos: store the current position
  - start : store the start position(for all node)
  - goal : store the goal position(for all node)
  - hStar : min cost to goal
  - fStar : hStar + gStar
  - parentNode : store the last node
  - checkGoal() : return t or f depends on the current pos and goal pos
- **NodeList**
  - find(x,y) : return the node with pos (x,y)
  - remove(node) : remove node from list

Algorithm

1. Use two NodeList to create openList and closeList.
2. Add start node to openList. g(Start) = 0, f(Start) = h(Start)
3. If openList is empty, then no route available

4. In openList, find the node with the min fStar.
5. if the current node is goal, search is done. otherwise put to closeList
6. find next move from up down left right (with validation)
7. for each move, do the following things:
    a. calculate f'(next) = g(current) + cost(current, next) +h(next)
    b. depends on the next node, do the following
        i. if nextNode not in either nodelist, f(next)←f'(next) and add it to openlist, also change its parent.
        ii. if nextNode in open list, and f'(next)<f(next), remove nextNode from openlist, f(next)←f'(next) then add to openlist again. update parent info.
        iii. if nextNode in closedlist and f'(next)<f(next), f(next)←f'(next) , move it to openlist. update parent.
8. keep doing 3 to 7 until it reaches its base case
9. from the end node, trace back to the startNode(where parent == none).
10. calculate the coordinates and print out the moves.

TEST

input: python robotplanner.py testgrid_large.txt 0 1 4 14
output: no route available

input: python robotplanner.py testgrid_large.txt 0 8 19 14
output: UP RIGHT RIGHT RIGHT RIGHT RIGHT UP RIGHT RIGHT RIGHT RIGHT RIGHT DOWN RIGHT RIGHT RIGHT RIGHT RIGHT RIGHT RIGHT DOWN DOWN DOWN DOWN DOWN RIGHT DOWN

input: python robotplanner.py testgrid_large.txt 10 3 5 0
output: LEFT UP UP LEFT UP LEFT

input: python robotplanner.py testgrid_large.txt 5 14 2 0
output: UP UP UP UP RIGHT UP UP LEFT UP LEFT LEFT LEFT LEFT LEFT LEFT UP UP UP RIGHT RIGHT UP UP RIGHT UP UP

input: python robotplanner.py testgrid_large.txt 0 13 0 0
output: no route available

```
Hidekis-MacBook-Air:AI HIDEKI$ python robotplanner.py testgrid_large.txt 0 1 4 14
no route available
Hidekis-MacBook-Air:AI HIDEKI$ python robotplanner.py testgrid_large.txt 0 8 19 14
UP RIGHT RIGHT RIGHT RIGHT RIGHT UP RIGHT RIGHT RIGHT RIGHT RIGHT DOWN RIGHT RIGHT RIGHT RIGHT RIGHT RIGHT RIGHT DOWN DOWN DOWN DOWN
DOWN RIGHT DOWN
Hidekis-MacBook-Air:AI HIDEKI$ python robotplanner.py testgrid_large.txt 10 3 5 0
LEFT UP UP LEFT UP LEFT
Hidekis-MacBook-Air:AI HIDEKI$ python robotplanner.py testgrid_large.txt 5 14 2 0
UP UP UP UP RIGHT UP UP LEFT UP LEFT LEFT LEFT LEFT LEFT LEFT UP UP UP RIGHT RIGHT UP UP RIGHT UP UP
Hidekis-MacBook-Air:AI HIDEKI$ python robotplanner.py testgrid_large.txt 0 13 0 0
no route available
```