jamiees2 / **astar.py**

Created 5 years ago

A* Algorithm implementation in python.

astar.py

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
class Node:
    def __init__(self,value,point):
        self.value = value
        self.point = point
        self.parent = None
        self.H = 0
        self.G = 0
    def move_cost(self,other):
        return 0 if self.value == '.' else 1

def children(point,grid):
    x,y = point.point
    links = [grid[d[0]][d[1]] for d in [(x-1, y),(x,y - 1),(x,y + 1),(x+1,y)]]
    return [link for link in links if link.value != '%']
def manhattan(point,point2):
    return abs(point.point[0] - point2.point[0]) + abs(point.point[1]-point2.point[0])
def aStar(start, goal, grid):
    #The open and closed sets
    openset = set()
    closedset = set()
    #Current point is the starting point
```

```
    current = start
    #Add the starting point to the open set
    openset.add(current)
    #While the open set is not empty
    while openset:
        #Find the item in the open set with the lowest G + H score
        current = min(openset, key=lambda o:o.G + o.H)
        #If it is the item we want, retrace the path and return it
        if current == goal:
            path = []
            while current.parent:
                path.append(current)
                current = current.parent
            path.append(current)
            return path[::-1]
        #Remove the item from the open set
        openset.remove(current)
        #Add it to the closed set
        closedset.add(current)
        #Loop through the node's children/siblings
        for node in children(current,grid):
            #If it is already in the closed set, skip it
            if node in closedset:
                continue
            #Otherwise if it is already in the open set
            if node in openset:
                #Check if we beat the G score
                new_g = current.G + current.move_cost(node)
                if node.G > new_g:
                    #If so, update the node to have a new parent
                    node.G = new_g
                    node.parent = current
            else:
                #If it isn't in the open set, calculate the G and H score for the node
                node.G = current.G + current.move_cost(node)
```

```
                node.H = manhattan(node, goal)
                #Set the parent to our current item
                node.parent = current
                #Add it to the set
                openset.add(node)
    #Throw an exception if there is no path
    raise ValueError('No Path Found')
def next_move(pacman,food,grid):
    #Convert all the points to instances of Node
    for x in xrange(len(grid)):
        for y in xrange(len(grid[x])):
            grid[x][y] = Node(grid[x][y],(x,y))
    #Get the path
    path = aStar(grid[pacman[0]][pacman[1]],grid[food[0]][food[1]],grid)
    #Output the path
    print len(path) - 1
    for node in path:
        x, y = node.point
        print x, y
pacman_x, pacman_y = [ int(i) for i in raw_input().strip().split() ]
food_x, food_y = [ int(i) for i in raw_input().strip().split() ]
x,y = [ int(i) for i in raw_input().strip().split() ]

grid = []
for i in xrange(0, x):
    grid.append(list(raw_input().strip()))

next_move((pacman_x, pacman_y),(food_x, food_y), grid)
```

rickhenderson commented on 6 May 2016

Mind if I play with this and see what I can do? I'm interested in trying it on OpenAI Gym. You should check out that project if you are interested.

rickhenderson commented on 6 May 2016

Dear God why would request input that way! 👎

Domiii commented on 11 Oct 2016

http://www.redblobgames.com/pathfinding/a-star/implementation.html <3 :)

OliverEdholm commented on 27 Nov 2016 • edited

Thanks, this really helped my studies.

Manhattan distance is currently:

```
def manhattan(point,point2): return abs(point.point[0] - point2.point[0]) + abs(point.point[1]-point2.point[0])
```

Shouldn't it be like this?:

```
def manhattan(point,point2): return abs(point.point[0] - point2.point[0]) + abs(point.point[1]-point2.point[1])
```

Saw that this was uploaded quite a while ago but I'm commenting this anyway.

Sincerely, Oliver.

raulvillora commented on 1 Dec 2016

Hi everyone. I am a bit lost. What do you mean by '# Enter your code here. Read input from STDIN. Print output to STDOUT' ?

We don´t know how to run the code. Can anybody help us?

Thanks so much.

ZinoKader commented on 3 Dec 2016 • edited

My understanding is that the grid that next_move takes is two nested lists, the higher level one being the blocks/nodes in X and each and every one of these X-blocks contains a list with Y blocks beneath it. Can someone confirm this or clarify?

rimonece commented on 16 Mar 2017

Hi everyone, can anyone please explain how to set up data in this algorithm code?

Isha8 commented on 13 Aug 2017

Hi, what does the astar method return? I don't see a return statement, so what exactly is stored in path and how?
Thanks.

hemantgupta2442 commented on 16 Jan

It says invalid syntax for line 73
print len(path) - 1
Don't know how to solve that.
Help please

jbarsce commented 13 days ago

@hemantgupta2442

Try "print(len(path) - 1)", because it sounds like you are using python 3.