

AI Pac-man(b) Team5 分組作業報告

400410048 資工四 黃韻文

● A* search

A* search 的公式為 $f(n)=g(n)+h(n)$ ， $g(n)$ 為起點到相鄰的 node 的路徑長， $h(n)$ 則表示相鄰 node 到終點的估算距離， $f(n)$ 則為該 node 的 cost。而 A* search 則是去尋找當前最低的 cost 去做下一次的起點，所以選用的 data structure 為 Priority queue，用以保證每次取出來的 node 所花費的 cost 都是最低的。

實作的方法為：

一開始將起始位置放入 Priority queue 中。接著從 Priority queue 取出 node，如果沒有 node，則表示起點跟終點間並沒有 path 路徑存在，如果有，則判斷是否為終點，是終點則把一直紀錄的 path 回傳。如果不是終點則取出他相鄰的 node，並判斷有無被搜尋過，有則跳過此 node 並繼續尋找未被搜尋過的 node，並計算該 node 的 cost 並與 path 一起存入 Priority queue 中，並重複此步驟直到找出終點。

● CornersProblem

實作: `class CornersProblem(search.SearchProblem)`

其中的一些函式：

```
def __init__(self, startingGameState):
    top, right = self.walls.height-2, self.walls.width-2
    top、right 放入 self 裡,使其知道地圖的大小,走最遠不會超過(right, top)
def getStartState(self):
    visitedCorners = (False, False, False, False)
    start = (self.startingPosition, visitedCorners)
    return start
```

先將是否遇到四個角落都設為 false，並將其與起始位置放回 start，並回傳

```
def isGoalState(self, state):
```

將現在的狀態判斷四個角落是否都有到達過，若都有的話就都是 (TRUE)1

將其全部 and 起來，回傳 `return corners[0] and corners[1] and corners[2] and corners[3]`

```
def getSuccessors(self, state):
```

```
cur_location = state[0]、toVisit = state[1]
```

如果下一個走到的地方不是牆、就判斷是不是走到

```

conner (1,1)(1, self. top)(self. right, 1)(self. right, self. top)
如果有走到這些 conner 就把那個 visitedCorners 裡的角落那設為 true 回傳
succ = ((next_x, next_y), visitedCorners), action
若不是走這些 conner，則回傳原本 "toVisit"
succ = ((next_x, next_y), toVisit), action
self._expanded +=1 #增加 expand node
def getCostOfActions(self, actions):題目本身已實作

```

● heuristics

我們所要做的 heuristics function 其估計的 cost 值必須要小於實際 cost 才有辦法保證找到最短路徑，否則小精靈會多走不必要的路而損失分數。在地圖上計算距離 cost 常見的方法有 manhattan 距離跟 euclidean 距離，但是實際的地圖其障礙物的分布並沒有規律，並不能保證每次都用這兩種其中某一種方式即是 \leq 實際 cost，有時甚至會遠小於實際 cost，expand 太多不必要的 node，於是我們想或許可以結合這兩種計算距離的方法，並根據障礙物的資訊（牆壁）去嘗試優化。

我們的想法是先記錄每次拿進來的路徑中碰到牆壁的次數(iswall)有多少，沒碰到牆壁的次數(notwall)又是多少，假設說這段路徑 iswall=0，則有極大可能可以用 euclidean 的方式達到逼近實際 cost 的估計值；iswall > notwall，則有可能這段路徑上需要轉彎的次數較多，用 manhattan 的方式去估計有可能會遠小於實際 cost，所以還得再加上可能會多花費的距離以求能逼近實際 cost，我們是先假設加上碰到牆壁的次數代表可能多花費的距離；除了上述這兩樣之外剩下的情況都用 manhattan 的方式去估算 cost。

評分隊友

學號	姓名	分數(0-10)	評論(50 字)
401410009	張育恩	8	實際見面不多話，但是做作業時有在關注進度，FB 也會看留言，不會未讀訊息，會幫忙轉 e-course 的最新消息，是個好隊友。
401410088	李惟信	9	交流最多的一個學弟，態度很積極，能幫忙的地方就會盡心去做，幫我連繫其他大三的學弟，GOOD！
401410045	蔡家禾	7	沒說到什麼話，網上也沒什麼交流，但是約出來討論實作方法的時候，感覺上滿有想法的。