

AI 作業(二): 人的隨便AI

@arbuztw

人的AI (參考)

- 計算地圖上任兩點的實際距離
- 看看相鄰的格子(上、下、左、右)哪個比較「好」
- 「好」的定義
 - 離鬼的距離
 - 可以吃到的金幣

DFS

- 怎麼算距離?

DFS

- 怎麼算距離?

	4	3	2	3	4	
	5		1		5	
	6	7	0	7	6	

DFS

- 怎麼算距離?

	4	3	2	3	4	
	5		1		5	
	6	7	0	7	6	

- 不一定是最短距離QQ

IDDFS

- 限制每次可走的距離

IDDFS

- 限制每次可走的距離
- $limit = 1$

			1			
		1	0	1		

IDDFS

- 限制每次可走的距離
- $limit = 2$

			2			
			1			
	2	1	0	1	2	

IDDFS

- 限制每次可走的距離
- $limit = 3$

		3	2	3		
	3		1		3	
	2	1	0	1	2	

IDDFS

- 限制每次可走的距離
- $limit = 4$

	4	3	2	3	4	
	3		1		3	
	2	1	0	1	2	

IDDFS

- IDDFS (Iterative Deepening DFS)
- 虛擬碼

```
DFS(x, y, d, lim)
  if (d > lim)
    return
  dis[sx][sy][x][y] = d

  for (dx, dy) = {(0,-1),(0,1),(-1,0),(1,0)}
    if (dis[sx][sy][x+dx][y+dy] > d 且 (x+dx, y+dy) 非牆壁)
      DFS(sx, sy, x+dx, y+dy, d+1, lim)

IDDFS(sx, sy)
  for all (x, y)
    dis[sx][sy][x][y] = INFINITY

  for limit = 0 to MAX_DEPTH
    DFS(sx, sy, sx, sy, 0, limit)
```

與鬼的距離

- 到三個鬼中最近的

```
min_dist = 到ghost[0]的距離  
for (i = 1; i < 3; i++)  
    min_dist = min(min_dist, 到ghost[i]的距離)
```

- (Challenge)計算距離該格一定距離內的格子到鬼的距離的max
 - 多看幾步的概念
 - 太遠的降低權重(e.g. 1/距離)

可以吃到的金幣

- 附近的格子中有多少金幣

```
get_gold = 0;
for (lx ...) {
    for (ly ...) {
        令d為(lx, ly)跟(x,y)的距離
        if (d <= GOLD_DIST 且 是金幣) {
            get_gold += SCORE_GOLD / (d + 1);
        }
    }
}
```

- 每個格子的權重不同，愈遠愈小(e.g. 1/距離)
 - 愈遠的愈不重要，因為可能在到達那格前就會改變方向

合起來算

- 鬼的距離 * 100 + 可以吃到的金幣
- 愈大的點愈「好」

實作函式

- `PacmanInit()` (Optional)
 - 在地圖畫好後被呼叫
 - 可以預先算好地圖上兩點的實際距離
 - `int dis[WIDTH][HEIGHT][WIDTH][HEIGHT];` 是全域變數
- `PacmanDecide()`

```
/* 每個回合會被呼叫，決定 Pacman 的方向  
   如果回傳的不是合法的方向 (例如有障礙物)，則會被忽略掉  
   - 回傳值: Pacman 的下一個方向, DIR_UP, DIR_DOWN, DIR_LEFT,  
             DIR_RIGHT 或 DIR_NONE  
*/  
int PacmanDecide() {  
    ...  
}
```

- 遊戲資料及邏輯與上週投影片相同

評分標準

- 在時限內得到的遊戲分數(吃到金幣、存活)
- Challenge
 - 調整參數
 - 考慮死路
 - ...
- 最多可繳交三種不同的AI(請包在同一個壓縮檔中)，評分時會取最高分者
- 請附上README說明你的AI如何運作