

AIML HW#2 Documentation

電機三 B06901019 楊軒

第五題 Evaluation Function 的程式碼：

```
currentPos = currentGameState.getPacmanPosition()

ghostStates = currentGameState.getGhostStates()
ghostPos = []
scaredPos = []

for ghost in ghostStates:
    if ghost.scaredTimer > 1:
        scaredPos.append(ghost.getPosition())
    else:
        ghostPos.append(ghost.getPosition())
```

將每一隻鬼根據他們的 `scaredTimer`，分為兩組：`scaredTimer` 大於 1（小精靈可以去吃）的鬼的位置存進 `scaredPos`，其他的存進 `ghostPos`。

```
nearestFood = lambda x, y, _: currentGameState.hasFood(x, y)
nearestGhost = lambda x, y, _: (x, y) in set(ghostPos)
nearestScare = lambda x, y, _: (x, y) in set(scaredPos)
nearestCaps = lambda x, y, _: (x, y) in currentGameState.getCapsules()

dist2Food = breadthFirstSearch(currentGameState, nearestFood)

dist2Ghost = 0
if len(ghostPos) > 0:
    dist2Ghost = breadthFirstSearch(currentGameState, nearestGhost)
dist2Scared = 0
if len(scaredPos) > 0:
    dist2Scared = breadthFirstSearch(currentGameState, nearestScare)
```

利用 BFS，分別求出離小精靈最近的食物、可以攻擊的鬼和一般的鬼距離小精靈多遠。

```

dist2Caps = 10 * len(currentGameState.getCapsules())
if len(currentGameState.getCapsules()) > 0:
    for x, y in currentGameState.getCapsules():
        if manhattanDistance((x, y), currentPos) < 5:
            continue
        break
    else: dist2Caps = breadthFirstSearch(currentGameState, nearestCaps)

```

我想要讓小精靈在比較靠近 capsule 的時候才會主動去吃，因此我設定只有在其中一個 capsule 和小精靈的曼哈頓距離小於 5 時，才會主動去做 BFS 搜尋與 capsule 的距離。但是，為了保持在離 capsule 較遠時 evaluation function 的值較低，必須將此項參數（和 evaluation function 的值呈負相關）的初始值設定得較高；同時，若剩下 capsule 的數量較多，此項參數也要比較高。

```

ghostScoreTotal = 0.791 * dist2Ghost - 5.992 * dist2Scared -
                25.07 * len(scaredPos)
capsScoreTotal  = 19.81 * dist2Caps +
                150.43 * len(currentGameState.getCapsules())
foodScoreTotal  = 0.4947 * dist2Food ** 1.2 +
                7.01 * currentGameState.getNumFood()

```

這一個部分將各種參數分類成與鬼、膠囊和食物相關的三組，並依照重要順序做加權（以下是以重要順序做排序）：

- 與鬼相關，而且分數越低越好的：scared 數量 > 與最近的 scared 距離
- 與鬼相關，而且分數越高越好的：與最近的鬼距離
- 與膠囊相關，分數越低越好：剩餘的膠囊數量 > 與最近的膠囊距離
- 與食物相關，分數越低越好：剩餘的食物數量 > 與最近的食物距離

為了不要讓小精靈在非必要時暴衝去吃太遠的食物，對於 dist2Food 我再做了 1.2 次方，讓食物離小精靈越遠時，分數會扣得越快。

另外，這些參數在加權時都乘以了看起來不太漂亮的小數，是因應要考慮的參數較多，為了減少小精靈在連續好幾格遇到同樣分數而原地徘徊的情況。

```

result = ghostScoreTotal - 1.2 * capsScoreTotal - 2 * foodScoreTotal +
        currentGameState.getScore()

return result

```

最後，將以上的參數加總，其中分數越低越好的參數都是乘上一個負的係數，並再加上遊戲狀態本身的分數，即是整個 evaluation function 所回傳的分數。